Foutcorrectie, precodering en bitallocatie-algoritmes in hogesnelheidstoegangsnetwerken

Error Correction, Precoding and Bitloading Algorithms in High-Speed Access Networks

Julie Neckebroek

Promotor: prof. dr. ir. M. Moeneclaey
Proefschrift ingediend tot het behalen van de graad van
Doctor in de ingenieurswetenschappen: computerwetenschappen

UNIVERSITEIT
GENT

Vakgroep Telecommunicatie en Informatieverwerking
Voorzitter: prof. dr. ir. H. Bruneel
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2016 - 2017

ISBN 978-90-8578-949-9
NUR 958
Wettelijk depot: D/2016/10.500/81

Members of the Jury

- Prof. Marc Moeneclaey
  Ghent University
- Prof. Gert De Cooman
  Ghent University
- Prof. Wout Joseph
  Ghent University
- Prof. Mamoun Guenach
  Nokia, Ghent University
- Prof. Jérôme Louveaux
  Université catholique de Louvain, Louvain-la-Neuve
- Prof. Marc Moonen
  KU Leuven

Dankwoord

9 jaar is niet de typische termijn van een doctoraat en hoewel ik het me initieel anders had vooropgesteld, vlogen de jaren voorbij. Gedurende die periode heb ik vele kansen gekregen en ik maak graag gebruik van deze paragraaf om de personen die ik dankbaar ben even in de aandacht te brengen.

De dag dat mijn promotor professor Marc Moeneclaey naar me toe kwam om me te vragen als onderzoeker te starten bij DIGCOM, onder de vorm van een doctoraat, zal ik niet snel vergeten. Ik had alle keuzevakken gevolgd die gegeven werden bij DIGCOM (ondanks hun moeilijkheidsgraad) en was er dat jaar ook met mijn thesis gestart. Kortom, ik vond het fantastisch steeds meer bij te leren over wat er gebeurde in het gebied van de digitale communicatie. Ook al waren er interessante alternatieven, ik moest er niet lang over nadenken om daar te starten. Ik wil hierbij Marc dan in eerste instantie ook bedanken, voor de kans die hij mij gegeven heeft, al die tijd bezig te zijn met dingen die ik echt graag deed.

Daarnaast ben ik Marc ook dankbaar voor zijn geweldige rol als promotor. Zijn deur stond al die jaren steeds open en het is indrukwekkend hoe hij elk ingewikkeld probleem op enkele bladen papier herleidt tot iets veel eenvoudigers. Ik heb heel veel van hem geleerd en ben dankbaar dat ik met hem heb mogen samenwerken.

Ook Frederik Vanhaverbeke wil ik bedanken voor de begeleiding en de wegwijs gedurende mijn thesis en de eerste jaren van mijn doctoraat.

Voor de technische ondersteuning gaat mijn oprechte dank naar Davy en Philippe. Het dagelijkse werk dat zij uitvoeren is zeer waardevol. Het volledige systeem veilig, snel en up running houden en daarnaast ook steeds bereikbaar zijn voor hulp. Ook dankbaar ben ik de mensen van het secretariaat: Patrick, Sylvia en Annette. Zij maken ons het dagelijkse leven iets eenvoudiger door alle administratieve beslommeringen op zich te nemen. Zij zijn steeds bereidwillig en bezorgen de pakjes tot op onze bureau.

Dankzij de samenwerking met Alcatel-Lucent (dat ondertussen Nokia heet) heb ik kunnen ervaren hoe onderzoek in de industrie zich afspeelt. Ik bedank dan ook in het bijzonder Mamoun Guenach, Jochen Maes, Michael Timmers, Werner Coomans, Paschalis Tsiaflakis, Rodrigo Moraes voor deze kans om mee te werken aan de ontwikkeling van G.fast. Ook Danny De Vleeschauwer, Koenraad Laevens en Natalie Degrande ben ik dankbaar voor de interessante samenwerking.

Voor de goede sfeer, ontspannende middaglunches in ons 'favoriet' restaurant De Brug, spelletjesavonden, badmintonwedstrijden, kubbtornooien, wedstrijden van de Rode Duivels, analyses van Game of Thrones afleveringen, ijsjes bij mooi weer, alle loop en triatlon events, de sportieve fietstocht,... bedank ik mijn collega's en lunchgenoten uitvoerig! Ook de fijne momenten op conferenties in Barcelona, Taiwan, Londen, Budapest,... zal ik niet snel vergeten. Vele collega's zijn ondertussen ook echte vrienden geworden.

Mijn ouders wil ik graag bedanken voor al de kansen en ervaringen die ze me gegeven hebben. Het is echt ongelooflijk hoeveel ze op verschillende vlakken in ons geïnvesteerd hebben en mee gebouwd hebben aan de mensen die we vandaag geworden zijn. Zonder hen was het zeker onmogelijk geweest. Ik wil ook andere dichte familieleden bedanken voor hun oprechte interesse en steun.

Voor ik dit dankwoord afsluit mag ik één iemand niet vergeten. Hij maakt mij het leven niet altijd gemakkelijk, maar desondanks is hij na al die jaren een deel van mij geworden en weet ik ondertussen dat ik niet meer zonder hem kan. Met hem is het leven nooit saai en heb ik vele nieuwe wegen betreden. Bedankt daarvoor!

Julie Neckebroek - November 2016

Contents

Abbreviations

ADC	Analog to Digital Converter
ADSL	Asymmetric Digital Subscriber Line
ARQ	Automatic Repeat Request
ATP	Aggregate Transmission Power
BER	Bit Error Rate
CND	Check Node Decoder
CNS	Column Norm Scaling
CO	Central Office
CP	Customer Premises
CRC	Cyclic Redundancy Check
DMT	Discrete Multitone
DP	Distribution Point
DSL	Digital Subscriber Line
DSLAM	DSL Access Multiplexer
DTU	Data Transfer Unit
EZF	Extended Zanatta-Filho
FDD	Frequency Division Duplexing
FEXT	Far-End Crosstalk

FTTdp	Fiber-to-the-Distribution-Point
FTTH	Fiber-to-the-Home
G.fast	Fast Access to Subscriber Terminals
GP	Goodput
HDTV	High-Definition Television
HG	Home Gateway
IN	Impulsive Noise
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISI	Intersymbol Interference
LAN	Local Area Network
LDPC	Low-Density Parity-Check Code
LLR	Log Likelihood Ratio
LP	Linear Precoding
LT	Luby Transform
MAC	Media Access Control
MC	Monte-Carlo
MDS	Maximum Distance Separable
MIMO	Multiple-Input Multiple-Output
MSA	Min-Sum Algorithm
NEXT	Near-End Crosstalk
NLP	Nonlinear Precoding
OSTBC	Orthogonal Space-Time Block-Code
P/S	Parallel-to-Serial Convertor
PHY	Physical
PSD	Power Spectral Density
PSK	Phase Shift Keying
PSTN	Public Switched Telephone Network
QAM	Quadrature Amplitude Modulation
QoE	Quality of Experience
RS	Reed-Solomon
RTT	Round-Trip Delay
S/P	Serial-to-Parallel Convertor
SDTV	Standard-Definition Television
SHDSL	Single-Pair High-Speed DSL
SISO	Single-Input Single-Output
SMSA	Scaled Min-Sum Algorithm
SPA	Sum-Product Algorithm
SR-ARQ	Selective-Repeat Automatic-Repeat-Request
STB	Set-Top Box
TCM	Trellis-Coded Modulation
TDD	Time Division Duplexing
TS	Transport Stream

VDSL	Very-High Bit Rate DSL
VND	Variable Node Decoder
WER	Word Error Rate
wLT	Weakend LT
XOR	Exclusive-Or Operation

Notations

$\oplus$	Modulo-2 sum
$\otimes$	Indicates that part II of the bitloading algorithm had to be executed
$\#$	Number of
$\neg$	Inverse bit operator
$\lfloor x \rfloor$	Largest integer less than or equal to x
$\mathcal{A}$	Symbol constellation
$\mathcal{A}_0^{(k),n}$	Subset of the symbol constellation $\mathcal{A}^{(k)}$ on tone k with n^{th} bit equal to 0
$\mathcal{A}_1^{(k),n}$	Subset of the symbol constellation $\mathcal{A}^{(k)}$ on tone k with n^{th} bit equal to 1
$\mathcal{A}_{\text{ext}}$	Periodic extension of the constellation $\mathcal{A}$
b	Number of bits per symbol point in the constellation
$\text{BER}^{(N_{\text{retr}})}$	BER after N_{retr} retransmissions
ByteER	Byte error rate
$\text{ByteER}_{\text{RS}}$	Byte error rate after RS decoding
C_M	M-QAM symbol constellation
C'_M	Set of transmit symbols for M-QAM in case of NLP
$E_{\text{PSD}}^{(k)}$	Vector of dimension $1 \times N_{\text{t}}$ with maximum spectral density constraints per tone
E_{ATP}	Maximum aggregate transmission energy constraint per line
F	Frequency spacing of the tones in multi-tone transmission

G_{dB}	Coding gain in dB
$\mathbf{H}^{(k)}$	Channel matrix of dimension $N_{\mathrm{u}} \times N_{\mathrm{u}}$
$\Im[x]$	Imaginary part of x
K	Number of information symbols in a codeword
K_{RS}	Number of information symbols per RS codeword
N	Number of code symbols in a codeword
$N_{0,\mathrm{imp},i}$	Power spectral density of the impulsive noise on the line of the ith user
$\mathbf{n}^{(k)}$	Noise vector of dimension $N_{\mathrm{u}} \times 1$ at frequency kF
N_{e}	Number of byte errors in N_{int} codewords
N_{F}	Number of sets of $N_{\mathrm{t}}/N_{\mathrm{F}}$ consecutive tones, with all tones within a same set experiencing the same fading gain, and the fading gains being correlated from one set to the next
N_{int}	Interleaver depth in codewords
n_{off}	Packet duration of an off-interval of impulsive noise
n_{on}	Packet duration of an on-interval of impulsive noise
N_{r}	Number of receive antennas
N_{retr}	Maximum number of retransmission of a DTU in ARQ
$N_{\mathrm{retr,max}}$	Maximum allowed N_{retr} due to latency constraints
N_{RS}	Number of code symbols per RS codeword
N_{t}	Number of tones in multi-carrier transmission
N_{tr}	Number of transmit antennas
N_{u}	Number of users at different locations connected to the same distribution point
$P_{\mathrm{e,DTU}}$	Probability of an erased DTU
$P_{\mathrm{e,RS}}$	Decoding error probability of a RS codeword
$P_{\mathrm{e,p}}$	Probability of an erased packet
P_{s}	Symbol error probability
$P_{\mathrm{unrec,DTU}}^{(N_{retr}+1)}$	Probability of an erroneous DTU after N_{retr} retransmisions
$P_{\mathrm{unrec,RS}}$	Probability of an unrecoverable codeword of a RS code
$\Re[x]$	Real part of x
R_{b}	Information bit rate
R_{c}	Code rate
R_{s}	Symbol rate
T_{ARQ}	Timer expiration duration for ARQ
T_{CP}	Time duration of the cyclic prefix
T_{DMT}	Time duration of a DMT symbol
T_{DTU}	Transmission time of the DTU
T_l	Inter-arrival time between the lth and $(l+1)^{\mathrm{th}}$ noise impulse
T_{lat}	Time duration of the latency introduced
T_{retr}	Time interval between retransmission instants of the same packet in a system with ARQ
t_{RS}	Error correcting capability of RS code
T_{RTT}	Round-trip delay time
x^*	Complex conjugate of x: $x^* = \Re[x] - j \cdot \Im[x]$

$\mathbf{x}^{(k)}$	Transmitted vector of dimension $N_{\mathrm{u}} \times 1$ at frequency kF
$\mathbf{y}^{(k)}$	Received vector of dimension $N_{\mathrm{u}} \times 1$ at frequency kF
τ_l	Duration of the lth noise impulse

Samenvatting

"Het is onmogelijk niet te communiceren."

– Paul Watzlawick

De dag van vandaag is digitale communicatie alomtegenwoordig en maakt ze een groot deel uit van eenieders leven. *Digitale* communicatie houdt in dat de over te brengen informatie eerst wordt omgezet naar digitale data, zoals bits. Steeds meer toestellen per persoon worden aangesloten op het internet, bijvoorbeeld digitale televisie decoders, smartphones, smart TVs, tablets, sporthorloges, digitale weegschalen... Kortom het internet is overal. Daardoor is er een vraag naar steeds meer bandbreedte wat overeenkomt met het verzenden en ontvangen van meer data per tijdseenheid. Ook het stijgend aantal toepassingen die communiceren over het internet, zoals radio and video streaming, en de stijgende video en audio kwaliteit zorgen ervoor dat het dataverkeer toeneemt.

In dit doctoraatsonderzoek hebben we ons hoofdzakelijk gefocust op het toegangsnetwerk tot het internet. Dit toegangsnetwerk treedt op als bottleneck voor de bereikbare toegangssnelheid. Al vele jaren spreekt men over de fiber-to-the-home (FTTH) connectie die elke huis rechtstreeks via onzettend snelle glazvezel met het glasvezel kernnetwerk verbindt. Maar in België en vele andere landen in Europa (Portugal is de enige uitzondering) bestaat enkel het

kernnetwerk uit glasvezel. De eindgebruiker is ermee verbonden via een toegangsnetwerk van koperen kabels, zoals coaxkabel of telefoonkabel bestaande uit getwiste paren, soms gevolgd door een draadloos toegangsnetwerk, zoals Wi-Fi of 4G. Ons onderzoek concentreert zich op het verzenden over telefoonkabel, namelijk de DSL lijn, en Wi-Fi. De reden waarom de installering van het FTTH netwerk uitblijft, is de uitzonderlijk hoge kost die gepaard gaat met het uitrollen van de glasvezelkabels, meer specifiek voor de nodige wegenwerken, het openleggen van voetpaden en opritten,... Momenteel wordt de kost gespreid over meerdere jaren en is er een graduele uitrolling van glasvezel bezig. Daardoor wordt het stuk koper of het toegangsnetwerk dat de eindverbruiker verbindt met het glasvezel kernnetwerk wel steeds korter. Het kortere toegangsnetwerk brengt verschillende opportuniteiten met zich mee. Op een kortere koperdraad is de verzwakking van de hogere frequenties lager en doordoor kan er een grotere bandbreedte worden gebruikt dan voorheen.

Doorgaans liggen de kabels die buren in eenzelfde straat bedienen naast elkaar in een *binder*. Doordat de kabels dicht bij elkaar liggen treedt er vaak overspraak op, dit wil zeggen dat het signaal afkomstig uit de ene kabel interfereert met het signaal in de andere kabels. Voor lage frequenties is deze overspraak meestal klein, maar voor de hogere frequenties kan het voorkomen dat de overspraak, afkomstig van de andere kabels in de *binder*, sterker is dan het rechtstreeks of oorspronkelijk signaal. Dit kan worden opgelost met een techniek die *precoding* heet. We bekijken twee soorten *precoding*, nl. lineaire en niet-lineaire. Ons doel is om een zo hoog mogelijke bitsnelheid te halen binnen bepaalde vooropgelegde limieten in vermogen. Om dit te bereiken hebben we twee algoritmes voorgesteld, namelijk het *column-norm scaling* (CNS) algoritme en het *extended Zanatta-Filho* (EZF) algoritme, die de bits verdelen over de beschikbare dragers op zo een wijze dat er met het beschikbaar vermogen zoveel mogelijk bits worden verstuurd. Dit is niet voor de hand liggend omdat bits alloceren op een drager voor een bepaalde gebruiker gevolgen heeft op de andere tonen en voor de andere gebruikers (zijn buren) omwille van de overspraak.

Natuurlijk willen we dat de verzonden bits correct aankomen bij de ontvanger en dat de ontvangen informatie overeenkomt met wat verstuurd was. Het optreden van bitfouten wordt veroorzaakt door witte ruis en impuls ruis op de koperkabel. Op het draadloos netwerk kan het signaal verzwakt worden door het optreden van fading. Dit alles bemoeilijkt het detecteren van de bits. Om binnen zulke omstandigheden toch op een betrouwbare manier informatie te verzenden, beschouwen we verschillende foutcorrigerende codes en retransmissieprotocols om de verzonden data te beschermen en analyseren hun prestatie.

Op de DSL lijn kunnen we concluderen dat het gebruik van de niet-lineaire precoder resulteert in hogere bitsnelheden dan de lineaire precoder. Daarnaast prefereren we het CNS bitallocatie algoritme boven het EZF algoritme, ook al

halen we er iets lagere bitsnelheden mee, de bitverdeling gebeurt aan een veel lagere rekencomplexiteit. Bovendien halen we voordeel uit het toevoegen van error controle aan ons systeem in vergelijking met ongecodeerde transmissie. *Low-density parity-check* (LDPC) codes halen het van de andere vooropgestelde beschermingsstrategieën (*trellis-coded modulation* (TCM), de opeenvolging van TCM met byte interleaver en RS code). We bekomen de hoogste bitsnelheden met de LDPC codes van hoge coderate, i.e. $R = 5/6$, $16/18$ en $20/21$, waarbij het afhangt van het niveau van impulsruis welke coderate verkozen wordt.

Ook de toepassing van retansmissieprotocols bevordert de connectiesnelheid. Het vernietigende effect van het occasioneel optreden van impulsruis wordt verholpen door gebruik van *selective-repeat automatic-repeat-request* (SR-ARQ); pakketten die verloren zijn worden opnieuw verzonden. In de afwezigheid van impulsruis halen we een klein bitsnelheidsvoordeel bij de mogelijkheid tot één retransmissie. Indien er wel impulsruis optreedt, halen we een serieus voordeel in bitsnelheid indien een maximum van 4 retransmissies is toegestaan.

In het geval van draadloze communicatie wordt de betrouwbaarheid verhoogd door een hogere diversiteit. Dit wil zeggen dat verschillende versies van het zelfde bericht de ontvanger bereiken via verschillende paden. Diversiteit kan verkregen worden door gebruik te maken van multiple-input multiple-output (MIMO) systemen, wat overeenkomt met meerdere zend- en ontvangstantennes, in combinatie met orthogonale spatio-temporele blokcodes (OSTBCs) en door het gebruik van SR-ARQ. Beide systemen brengen kosten met zich mee, namelijk respectievelijk een hardware kost bij een toename van het aantal antennes, en de kost van een groter buffergeheugen. Een afweging kan gemaakt worden tussen beide technieken om de vereiste diversiteit te bereiken.

We beëindigen dit proefschrift met een samenvatting van de belangrijkste resultaten en stellen enkele onderwerpen voor om dit onderzoek verder uit te breiden, zoals Hybrid ARQ en Raptor codes.

Summary

"One cannot not communicate."

– Paul Watzlawick

Nowadays digital communication is ubiquitous and largely present in everyone's lifes. *Digital* communications implies that prior to transmission, the information to be transferred is converted to digital data, like bits. More and more devices per user are connected to the internet, e.g., digital television decoders, smart phones, smart TVs, tablets, sport watches, digital scales,... To put it briefly, the internet is everywhere. This causes an increase for ever more bandwidth, corresponding to the transmission and reception of more data per time unit. Also the growing number of applications that communicate over the internet, like radio and video streaming, and the increasing quality of audio and video make that the data traffic increases.

In this dissertation, we are mainly focused on the access network to the internet. This access network serves as a bottleneck to the achievable access speed. Already for many years, fiber-to-the-home (FTTH) is mentioned as the link that connects every house directly via immense fast fiber to the fiber core network. But in Belgium and many other countries in Europe (Portugal is the only exception), only the core network consists of fiber links. The end user is

connected to it through an access network consisting of copper cables, like coax and twisted-pair telephone-cable, sometimes followed by a wireless access network like Wi-Fi or 4G. The reason for the delay of the FTTH network installation is the exceptional high costs associated with deployment of the fiber cables, i.e. the required roadworks, breaking up pavements and driveways,... Currently, this cost is spread over several years and there is a gradual deployment of fiber. This makes that the copper link that connects the end user with the fiber core network, is getting shorter. The shorter access network has several opportunities. The higher frequencies are less diminished and this lets us use a larger bandwidth than before.

Usually cables that serve neighbours in a single street are co-located in a binder. The presence of nearby cables in the binder causes the occurrence of crosstalk. This means that the signal from one cable interferes with the signal in the other cables. Crosstalk is usually small at the lower frequencies. At the higher frequencies, it is possible that the crosstalk originating from the other cables in the binder, is stronger than the original signal. This crosstalk can be canceled by applying a technique called precoding. We consider both linear and non-linear precoding. We target a maximization of the bit rate, subject to power limitations. To achieve this, we propose two bitloading algorithms, i.e. the column-norm scaling (CNS) algorithm and the extended Zanatta-Filho (EZF) algorithm, that allocate the bits on the available tones such that as many bits as possible are transmitted with the available power. This is not at all obvious, as allocating bits on one tone has consequences for the other tones and users (his neighbours) due to the crosstalk.

Of course, it is desired that the transmitted bits reach the receiver without errors and that the received information matches with what was transmitted. The occurrence of bit errors is caused by white noise and impulsive noise on the copper cable. On the wireless channel, the signal might be distorted due to the occurrence of fading. All this complicates the detection of the bits. To be able to reliably transmit information in these circumstances, we consider different error correcting codes and retransmission protocols to protect the transmitted data and we analyze their performance.

On the DSL line, we conclude that the use of the non-linear precoder results in higher bit rates than the linear precoder. Furthermore, we prefer the CNS algorithm above the EZF algorithm. Although, it achieves a minor lower bit rate, the algorithm runs at a major computation complexity reduction. Moreover, we obtain advantage from the addition of error control to our system as compared to uncoded transmission. Low-density parity-check (LDPC) codes beat the other proposed protection strategies (trellis-coded modulation (TCM), the concatenation of TCM with byte-interleaver and RS code). We achieve the highest bit rates with the LDPC codes of higher rate, i.e. $R = 5/6$, $16/18$ en $20/21$, where it depends on the level of impulsive noise which code rate is preferred.

Also, the application of retransmission protocols improves the connection speed. The destroying impact of the occasional occurrence of impulsive noise is tackled by the use of selective-repeat automatic-repeat-request (SR-ARQ); lost packets are retransmitted. In the absence of impulsive noise, we obtain a small increase in bit rate by the allowance of one retransmission. If impulsive noise is present, a huge advantage is possible in bit rate if a maximum of 4 retransmissions is allowed.

In the case of wireless communication, the reliability is increased by a higher diversity. This implies that different replicas of the same message reach the destination through different paths. Diversity can be obtained by the use of multiple-input multiple-output (MIMO), which corresponds to several send and receive antennas, in combination with orthogonal space-time block codes (OS-TBCs) and through the use of SR-ARQ. Both systems come with a cost, i.e. respectively a hardware cost associated with an increase of antennas and the cost of a larger retransmission buffer. A trade-off exists between both techniques, in order to achieve the required diversity order.

We finish this dissertation with a concluding summary of the most important results and propose some subjects for future research, such as Hybrid ARQ and Raptor codes.

1 Introduction

In this dissertation, we study the effect of error protection strategies and precoding techniques on the error performance and quality of service of the end application. We present some bitloading algorithms to maximize the error free throughput. This doctoral thesis consists of two separate although related parts.

In the first part, we consider transmission over DSL channels and examine which precoding and protection strategy is preferred to obtain the highest possible throughput. In the second part, the topic is shifted to the transmission of video over an indoor wireless channel subject to Rayleigh fading.

In Section 1.1, we give some background on the notion 'digital communication system'. Section 1.2 gives the motivation for this work and the outline of this thesis is presented in Section 1.3.

1.1 Background

In a digital communication system, a transmitter wants to transmit information bits over a transmission medium to a receiver as visualized in the block diagram in Figure 1.1. The information bits may originate from an analog or digital

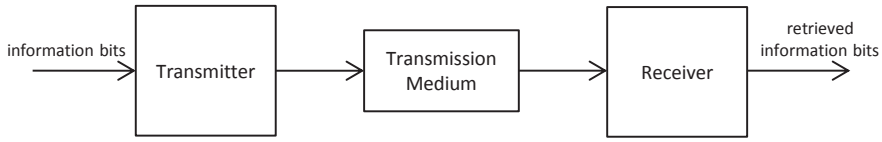

Figure 1.1: Block diagram of a general digital communication system.

source, and from any kind of application such as a website, a music station, a video game, telephone or e-mail. In the case of a telephone call, the voice comes typically from an analog source (a human) and is converted to information bits by an analog to digital converter (ADC). Also the transmission medium or channel can be of all kinds; e.g., the twisted-pair telephone line, coax cable, fiber, satellite connection, Wi-Fi network,... Our work is focused on two of these channels; i.e. the twisted-pair telephone line in Part I and the Wi-Fi network in Part II of this dissertation.

The transmitter transforms the information bits into a physical signal (e.g., optical signal, electromagnetic signal,...) that needs to be transmitted over the channel. The receiver observes the physical signal at the output of the channel, tries to reconstruct the information bits and outputs them to the application. On the channel, several sources of distortion may disturb the transmitted signal, like interference from other signals and noise. Therefore, the signal at the receiver side is likely to differ from the original transmitted signal. In this way, it is possible that the recovered information bits at the receiver are not equal to the original information bits at the transmitter.

In a good communication system, the system is designed so that with high probability the recovered information bits equal the original transmitted bits. Various techniques can be applied. For example, redundancy can be added to the transmitted information bits, so that, in spite of some erroneous bits at the receiver, the information bits can still be recovered (forward error correction (FEC)). Also retransmissions may be scheduled if the received bits are detected to be erroneous (automatic repeat request (ARQ)); the transmitter sends after notification a new copy of the original transmitted information bits.

Of course, transmitting information requires energy. An increase of power improves the received signal quality. But increasing the transmission energy is not always desirable, as it causes the transmitted signal to interfere more with other signals, and using less power is considered beneficial in a "green" world. Also, the transmit power of wireless communications is subject to regulations designed to limit the exposure to protect the public health. We want to use the transmission medium as efficient as possible; i.e., to achieve the required QoS with the least possible transmit energy.

1.2 Motivation

Prediction states that IP traffic will grow with 23 % from 2014 to 2019 [1]. Clearly, the bit rate demand is still increasing fast. The average number of devices per user grows, and expected is a growth in especially IP traffic from other devices than PCs; i.e. smartphones, TVs, tablets,...

Already many years ago, fiber-to-the-home (FTTH) was mentioned as the next very high-speed (speed of several Gbps) internet connection as fiber can carry data at very high speed over large distances. But due to the excessive costs that are needed to install fiber into all homes (breaking open pavements and driveways, road works,...), the fiber installation is spread over a longer time period and the deployment is gradually. The current broadband network infrastructure consists mainly of a fiber aggregation network with a copper access network that connects the end user with the fiber termination. The access network is the bottleneck to the achievable bit rate that the end user experiences. Thanks to the gradual deployment of fiber in the network, the access network becomes shorter and shorter and new challenges and opportunities to employ the channel become available. The shorter copper loop allows to use a higher bandwidth, so that new techniques can be applied to efficiently exploit the increased bandwidth.

In the second part of this thesis, we deal with a different challenge but use similar techniques as in the first part. Transmission over a wireless channel is subject to fading which may cause bursts of erroneous bits if a deep fade arises. The transmission of video, more specifically high definition television (HDTV) must meet severe restrictions on latency and only a limited number of visual distortions per time interval is allowed, in order to satisfy the required quality of service (QoS).

1.3 Outline

This dissertation is organized as follows.
Chapter 2 presents some general and basic concepts of forward error correction schemes and automatic repeat request protocols.

Part I

Chapter 3 gives an introduction on the transmission over the DSL channel. It gives an overview of the history of DSL and discusses two sources of noise from which transmission over DSL suffers.
Chapter 4 describes the system model as it is used throughout Part I of this thesis. A mathematical description of the system with crosstalk is provided, and the different precoding techniques to deal with this crosstalk are presented. Furthermore, the reader will find in this chapter an overview of the constellation

types used and a statistical model for the impulsive noise.
Chapter 5 investigates the error performance with linear and nonlinear precoding for uncoded transmission and various types of coded transmission.
Chapter 6 presents two bitloading algorithms for linear and nonlinear precoding under simultaneous aggregate transmit power and power spectral density constraints.
Chapter 7 provides the resulting error free throughputs for the system without impulsive noise and application of the different precoders, bitloading algorithms and coding schemes and concludes how the system should be designed in the absence of impulsive noise.
Chapter 8 extends the results from Chapter 7 by including impulsive noise, and concludes which error protection scheme is preferred in presence of impulsive noise.

Part II

Chapter 9 analyzes application layer ARQ for the protection of video packets over an indoor MIMO-OFDM link with correlated block fading. We examine a suitable form of error control to protect video packets against losses and to maintain a sufficient quality of experience for the end user watching the video.

Chapter 10 summarizes the conclusions of the obtained results. Furthermore, it proposes some topics for future research, that fell outside the scope of this dissertation. This chapter wraps up with a complete list of our publications.

2 Error Control

In this chapter, we introduce some general concepts of forward error correction (FEC) schemes and automatic request (ARQ) protocols. In Section 2.1, we start with uncoded transmission, the performance of which we will use as a benchmark for coded transmission. Sections 2.2 and 2.3 provide some background information about respectively trellis-coded modulation (TCM) and low-density parity-check (LDPC) codes. Reed-Solomon (RS) coding is introduced in Section 2.5. Sections 2.6 and 2.7 discuss respectively interleaving and ARQ. Section 2.8 presents the performance indicators which will be used in this dissertation.

2.1 Uncoded Transmission

In uncoded transmission, there is no channel encoder present and no redundancy is added to the information bit stream to be transmitted. The information bit stream is mapped to constellation symbols, modulated on waveforms and transmitted over the channel. On the channel, the signals are subject to noise and interference. Besides the Euclidean distance between the constellation points, that for a given constellation depends on the symbol energy, there is no protection that might increase the reliability of the transmitted data. At the receiver,

the received signals are demodulated, yielding noisy complex symbols. Hard decision on the noisy symbols is performed: the receiver assumes that the constellation point closest (in terms of Euclidean distance) to the received noisy symbol value was transmitted.

The addition of reduncancy by a channel encoder serves to increase the reliability of the received data but, unlike uncoded transmission, decreases the information bitrate for a given signal constellation.

2.2 Basic Principles of TCM

Trellis-coded modulation (TCM) is a commonly applied technique for combined coding and modulation in bandwidth-constrained channels. A coding gain can be achieved without increasing the signal bandwidth, compared to uncoded transmission. The redundancy introduced by the code is achieved by expanding the constellations. The modulation and coding are jointly optimized to maximize the Euclidean distance between coded symbol sequences by using set partitioning as described in Section 2.2.1.

TCM was first proposed in 1976 by Ungerboeck [2], but became well adopted and researched in 1982 only, when a more detailed publication of Ungerboeck [3] appeared.

Figure 2.1: Gottfried Ungerboeck (born March 15, 1940, Vienna, Austria).

Since then, TCM has been employed in many digital transmission systems with higher-order constellations, i.e., modulation schemes that transmit more than one bit per channel use, such as PSK and QAM. As far as digital subscriber line (DSL) systems are concerned, TCM has been adopted first in the single-pair high-speed DSL (SHDSL) standard [4], and remains an essential ingredient in the transceivers up to the most recently deployed very high speed DSL 2 (VDSL2) standard [5]. For more information about the DSL standards, the reader is referred to Section 3.1.

2.2.1 Set Partitioning

To maximize the minimum Euclidean distance between the coded symbol sequences in TCM, an appropriate mapping of the coded bits to constellation symbols has to be determined. Such a mapping method was developed by Ungerboeck [3], called set partitioning.

Consider a trellis encoder, which encodes m incoming information bits to $m+1$ encoded bits. These coded bits are mapped on constellation symbols from a symbol constellation $\mathcal{A}$, containing 2^{m+1} symbol points. This symbol constellation is partitioned into subsets in a way that the minimum Euclidean distance between the points in a subset is increased with each partitioning step. The subsets resulting from a given partitioning step have equal size. This technique is illustrated in Figure 2.2 for 16-QAM.

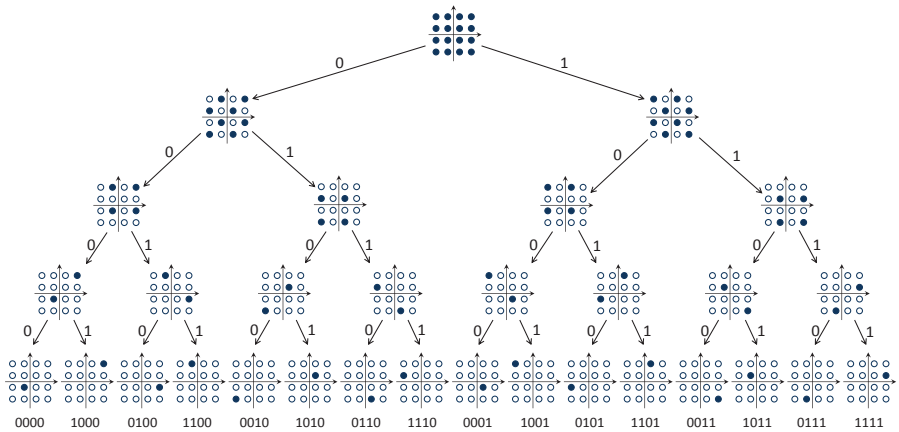

Figure 2.2: Set partitioning of 16-QAM.

In a first step, the 16-QAM constellation is partitioned into two subsets of 8 points each, by assigning two neighbouring points at minimum distance to different subsets. This way, the minimum squared Euclidean distance is increased from d^2 (for the original constellation) to $2d^2$. Each partitioning step in a squared constellation doubles the minimum squared Euclidean distance in the resulting subsets. Further partitioning of each subset divides again the points alternating in two separate subsets and the minimum squared Euclidean distance is increased to $4d^2$. After the fourth and last partitioning, each of the 16 subsets contains one remaining constellation point. The minimum squared Euclidean distance in these subsets is defined to be $+\infty$. In this example, the partitioning consists of four steps, until no further partitioning is possible and each subset contains one symbol point. In general the partitioning can stop after less than he maximum possible number of steps, as explained in Section 2.2.2.

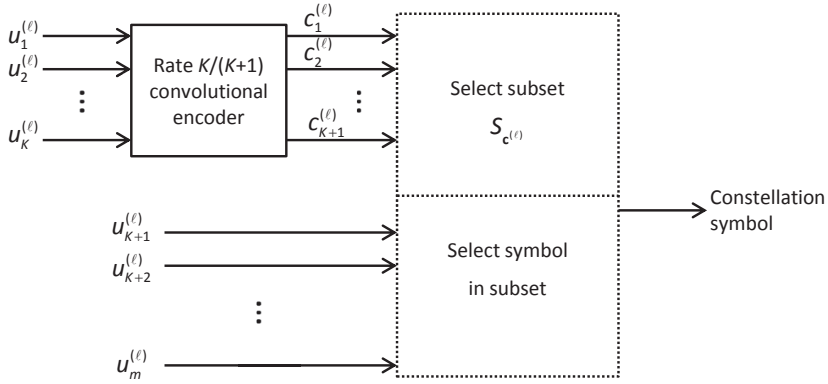

Figure 2.3: General structure of TCM encoder.

2.2.2 General Structure of TCM Encoder

Figure 2.3 shows the general structure of the TCM encoder, which transforms m information bits into a symbol from a constellation $\mathcal{A}$ that contains 2^{m+1} points. From the m information bits $\mathbf{u}^{(\ell)} = (u_m^{(\ell)}, u_{m-1}^{(\ell)}, \ldots, u_1^{(\ell)})$ that enter the encoder at time ℓ, the K least significant bits $(u_K^{(\ell)}, u_{K-1}^{(\ell)}, \ldots, u_1^{(\ell)})$ are encoded using a rate $K/(K+1)$ binary convolutional encoder, with $K \leq m$. The encoder produces $K+1$ coded bits $\mathbf{c}^{(\ell)} = (c_{K+1}^{(\ell)}, c_K^{(\ell)}, \ldots, c_1^{(\ell)})$. A number of $K+1$ partitioning steps are applied to the constellation $\mathcal{A}$, yielding 2^{K+1} subsets, each containing 2^{m-K} constellation points. When $K < m$, the $m-K$ most significant bits $\mathbf{u}'^{(\ell)} = (u_m^{(\ell)}, u_{m-1}^{(\ell)}, \ldots, u_{K+1}^{(\ell)})$ remain uncoded and are sent together with the coded bit vector $\mathbf{c}^{(\ell)}$ to the symbol mapper. The symbol mapper converts $(\mathbf{u}'^{(\ell)}, \mathbf{c}^{(\ell)})$ into a constellation symbol that belongs to $\mathcal{A}$. The subset $S_{\mathbf{c}^{(\ell)}}$ of the constellation $\mathcal{A}$ is selected by the coded bit vector $\mathbf{c}^{(\ell)}$, while $\mathbf{u}'^{(\ell)}$ picks the resulting constellation point among the 2^{m-K} symbols in the subset $S_{\mathbf{c}^{(\ell)}}$.

2.2.3 Convolutional Encoding

Convolutional codes were first proposed by Elias in 1955 [6]. A convolutional encoder of rate $R_\mathrm{c} = K/N$ is shown in Figure 2.4. The encoder contains K shift registers, each containing $L-1$ delay blocks; L is called the constraint length of the code. The N coded bits are obtained as modulo-2 sums of bits from the K shift registers; these sums are defined by the N generator sequences

$$\mathbf{g}_i^{(j)} = (g_{i,0}^{(j)}, g_{i,1}^{(j)}, \ldots, g_{i,L-1}^{(j)})$$

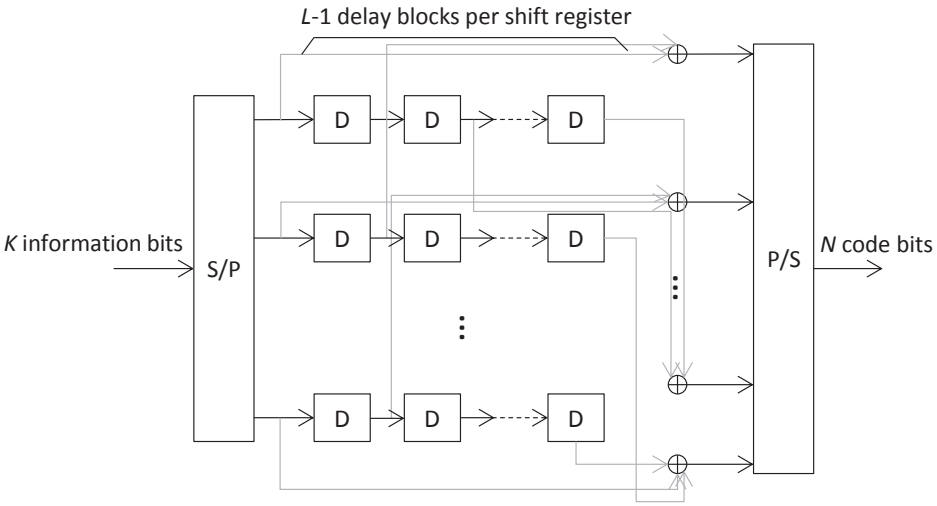

Figure 2.4: General rate K/N convolutional encoder.

Figure 2.5: Peter Elias (November 23, 1923, New Jersey – December, 7 2001).

with $i = 1, \ldots, K$, $j = 1, \ldots, N$ and $g_{i,l}^{(j)} \in \{0, 1\}$. These generator sequences determine which of the delay-block outputs are connected to which modulo-2 adders, as visualized in Figure 2.4. For $g_{i,l}^{(j)} = 1$, the lth delay block output from the ith register is connected to the jth modulo-2 adder. For $g_{i,l}^{(j)} = 0$, there is no connection. Taking $l = 0$ refers to the current input without delay.

The register is initialized to all-zero. The information bit sequence is demultiplexed by a serial-to-parallel convertor (S/R) into K lower-rate sequences which are fed to the K registers. At discrete instant ℓ, the ith register contains the L information bits $(u_i^{(\ell)}, \ldots, u_i^{(\ell-L+1)})$, and a N bit output vector is produced by applying a parallel-to-serial convertor (P/S) to the N modulo-2 sums

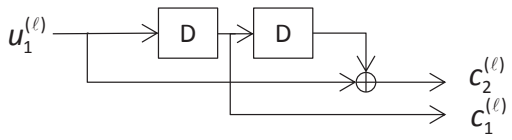

Figure 2.6: Example of a rate 1/2 convolutional encoder.

of the current register bits, computed according to

$$c_j^{(\ell)} = \sum_{i=1}^{K} \sum_{l=0}^{L-1} g_{i,l}^{(j)} u_i^{(\ell-l)} \tag{2.1}$$

for $j = 1, \ldots, N$.

For TCM in particular, a convolutional encoder is used with $N = K + 1$ coded bits. Adding one redundant bit by means of coded modulation is a solid choice as Ungerboeck states that the largest gain in SNR at a mutual information of m bits is made by doubling the signal constellation from size 2^m to 2^{m+1}, while the additional gain resulting from further constellation expansion is rather small [3].

2.2.4 Trellis Decoding

Convolutional code An alternative way to specify a convolutional code is by its trellis diagram. The trellis contains a number of nodes at each time instant equal to the number of possible states (at most $2^{K(L-1)}$ states), determined by the 'past' information bits in the shift registers. An edge connects two states at successive time instants if there exists an input bit sequence that causes the transition from the first state to the next. Each state has 2^K incoming edges and 2^K outgoing edges. Each outgoing edge from one state corresponds to a different input bit sequence. The total of all nodes at two successive instants and the connecting edges is called a trellis section. The structure of the trellis diagram is repetitive, it is a repetition of consecutive trellis sections. A simple example is shown in Figures 2.6 and 2.7. In Figure 2.6, a rate 1/2 convolutional decoder with 2 delay blocks is depicted. Figure 2.7 gives the corresponding trellis section with 4 states. To the left of the trellis, in front of each state, we mention the decimal input values $u_1^{(\ell)}$ corresponding to the outgoing branches from top to bottom. To the right of the trellis, behind each state we mention the decimal values of $(c_2^{(\ell)}, c_1^{(\ell)})$ corresponding to the incoming branches in the state from top to bottom.

The Viterbi algorithm [7] performs efficient maximum-likelihood sequence decoding of convolutional codes. It exploits the structure of the trellis diagram to determine the codeword with minimum distance to the received word.

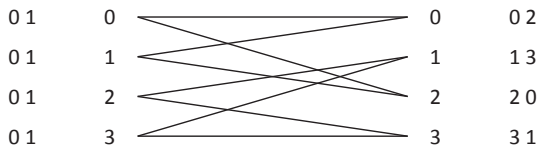

Figure 2.7: Trellis section corresponding to the convolutional encoder in the example from Figure 2.6.

Figure 2.8: Andrew Viterbi (born March 9, 1935, Bergamo, Italy).

A search of possible paths through the trellis has to be completed. On each edge, the metric for this search is the corresponding squared Euclidean distance between the noisy received symbols and the output symbols for the current transition. If two or more paths enter the same state, only the path with minimum cumulated squared Euclidean distance is kept.

TCM In TCM, the presence of uncoded bits gives rise to parallel transitions in the trellis diagram. The trellis diagram of TCM is given by the trellis from its convolutional code where each state transitions turns into 2^{m-K} parallel transitions, as the state transition is solely determined by the current state and the current input bits of the convolutional encoder. If two input vectors $\mathbf{u_1}^{(\ell)}$ and $\mathbf{u_2}^{(\ell)}$ differ only in the uncoded bits $(u_m^{(\ell)}, u_{m-1}^{(\ell)}, \ldots, u_{K+1}^{(\ell)})$, they will result in the same state transition. Thus, all symbol points from one subset $S_{\mathbf{c}^{(\ell)}}$ correspond to parallel transitions in the trellis diagram. Figure 2.9 shows an example for TCM with the convolutional encoder from Figure 2.6 with 4 bits at the output. The corresponding set partitioning of 16-QAM is depicted in Figure 2.10. The constellation points are labeled by the decimal value of the two bits $(c_2^{(\ell)}, c_1^{(\ell)})$ according to the subset they belong to.

The Viterbi algorithm is analogues for TCM as for a convolutional code. The presence of parallel transitions is solved by only retaining the transition with minimum Euclidean distance from all parallel transitions between two states.

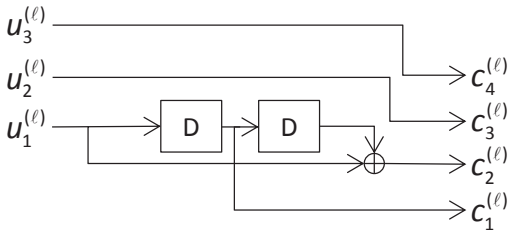

Figure 2.9: Example of TCM with 2 encoded bits and 2 uncoded bits, with convolutional encoder from Figure 2.6.

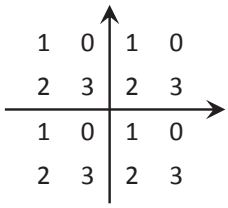

Figure 2.10: Mapping of subsets in 16-QAM.

2.3 LDPC Codes

A low-density parity-check (LDPC) code is a linear error-correcting block code characterized by a sparse parity-check matrix **H**. LDPC codes are capacity-approaching. Shannon proved [8] that block codes exist that achieve arbitrarily low error rate at any information rate up to capacity. Practical LDPC codes exist that operate close to the Shannon limit. LDPC codes are popular because they are equipped with fast encoding and decoding algorithms.

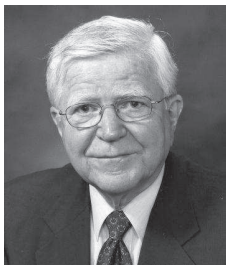

Figure 2.11: Robert G. Gallager (born 29 May 1931, Philadelphia).

LDPC codes were first invented by Gallager [9] in 1963 but were not considered useful because too complex to be implemented with the available hardware at that time. It took until 1996 before the LDPC codes were rediscovered by

MacKay and Neal [10, 11].

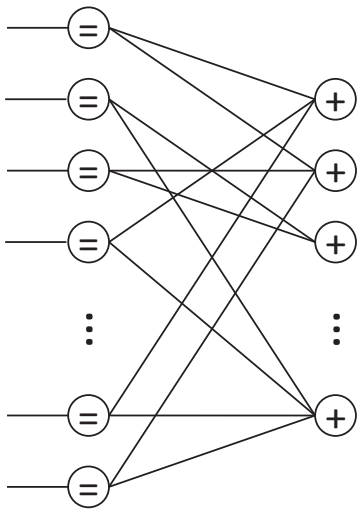

Figure 2.12: Bipartite graph of an LDPC code.

The LDPC code for rate $R_c = K/N$ can be visualized by a bipartite graph, as shown in Figure 2.12, which is determined by the $(N-K) \times N$ check matrix $\mathbf{H}$. A binary vector $\mathbf{c} = (c_1, ..., c_N)$ is a valid codeword if and only if $\mathbf{c}$ satisfies the constraint $\mathbf{c}\mathbf{H}^T = 0$, which represents $N-K$ parity-check equations. The graph consists of N nodes (the variable nodes) on the left and $M = N-K$ nodes (the check nodes) on the right, with an edge interleaver in between. The nth variable node is connected by means of an edge to the mth check node if $H_{m,n} = 1$. The nth variable node represents the nth bit in the codeword. This bit participates to $d_v^{(n)}$ parity checks (i.e., the nth column of $\mathbf{H}$ contains a 1 at $d_v^{(n)}$ positions), so that this variable node is connected to $d_v^{(n)}$ check nodes. We will refer to $d_v^{(n)}$ as the degree of the nth variable node. Each check node represents a parity-check equation. The mth check node has $d_c^{(m)}$ incident edges, indicating that $d_c^{(m)}$ variable nodes are involved in its parity-check equation (i.e., the mth row of $\mathbf{H}$ contains a 1 at $d_c^{(m)}$ positions). We will refer to $d_c^{(m)}$ as the degree of the mth check node.

2.3.1 Iterative Decoding

Iterative decoding or belief propagation is performed by passing messages between the variable nodes and the check nodes along the edges. We consider three decoders, which operate on soft information.

2.3.1.1 Sum-Product Algorithm

The sum-product algorithm (SPA) was developed by MacKay and Neal [11] and gives excellent decoding results. The algorithm operates on log-likelihood ratios (LLRs), which represent soft information of the coded bits, derived from observing the channel output. At initialization, all messages on the edges of the

Figure 2.13: David MacKay (April 22, 1967, Stoke-on-Trent, U.K. – April 14, 2016).

Tanner graph are set to 0. Next, the LLRs of the N coded bits are computed from the channel observations; this operation is referred to as *soft demapping*. Let us consider an AWGN channel characterized by $y = x + w$, where x is a symbol drawn from a constellation $\mathcal{A}$ representing b bits (among which the coded bit c_n), and w is complex-valued additive white Gaussian noise with variance $2\sigma^2$. The LLR $L_{\text{ch}\to n}$ related to the nth coded bit c_n is defined as

$$L_{\text{ch}\to n} = \ln \frac{p(y|c_n = 0)}{p(y|c_n = 1)} \qquad n = 1, \ldots, N \tag{2.2}$$

where $p(y|c_n)$ is the conditional probability density of the channel output y given the value of the coded bit c_n. For the considered channel model, (2.2) reduces to

$$L_{\text{ch}\to n} = \ln \frac{\sum\limits_{x' \in \mathcal{A}_{c_n=0}} \exp\left(-\frac{1}{2\sigma^2}\,|y - x'|^2\right)}{\sum\limits_{x' \in \mathcal{A}_{c_n=1}} \exp\left(-\frac{1}{2\sigma^2}\,|y - x'|^2\right)} \tag{2.3}$$

where $\mathcal{A}_{c_n=0}$ and $\mathcal{A}_{c_n=1}$ represent the subsets of the symbol constellation $\mathcal{A}$, corresponding to $c_n = 0$ and $c_n = 1$, respectively.

We denote by $N(m) = \{n : H_{m,n} = 1\}$ the set of variable nodes that are connected by an edge to check node m, and by $M(n) = \{m : H_{m,n} = 1\}$ the set of check nodes that are connected by an edge to variable node n.

We iterate from variable nodes to check nodes and back until some stopping condition is fulfilled. A decoding iteration consists of following two steps:

- Update messages $L_{n\to m}$ from variable node n to check node m:

$$L_{n\to m} = L_{\text{ch}\to n} + \sum_{m'\in M(n)\setminus m} L'_{m'\to n} \tag{2.4}$$

- Update messages $L'_{m\to n}$ from check node m to variable node n:

$$L'_{m\to n} = 2\tanh^{-1}\left(\prod_{n'\in N(m)\setminus n} \tanh\left(\frac{L_{n'\to m}}{2}\right)\right) \tag{2.5}$$

The output messages of the decoding process are computed as follows after every iteration:

- Update messages $L_{n\to\text{out}}$ from variable node n to the output:

$$L_{n\to\text{out}} = L_{\text{ch}\to n} + \sum_{m\in M(n)} L'_{m\to n} \tag{2.6}$$

  which corresponds to the logarithm of the a posteriori probability ratio of the nth coded bit c_n.

The decoder makes a hard decision for the nth coded bit; this decision is equal to 0 if $L_{n\to\text{out}} > 0$ and 1 otherwise. The iterations are stopped when the vector of N decisions is a valid codeword (i.e., when this vector satisfies the parity-check equations), or when the maximum number of iterations is achieved.

Note that during the iterations, the values of the LLR messages $L_{\text{ch}\to n}$ do not change: the soft demapping is not iterative. In principle, the decoding performance can be improved by performing *iterative* soft demapping. In this case, the messages $L_{\text{ch}\to n}$ are updated with each iteration, based on the messages $L'_{m\to n'}$ for all $m \in M(n')$ and all n' for which the corresponding coded bits $c_{n'}$ belong to the same constellation symbol as the considered bit c_n. However, applying iterative soft demapping considerably increases the computational complexity of the decoding process (especially when using large constellations), whereas it can be shown that in the case of Gray mapping the performance gain is very small. In this dissertation we consider Gray mapping (or a close approximation thereof), and restrict our attention to non-iterative soft demapping in order to limit the computational complexity.

Two major drawbacks of the SPA are: (i) the noise variance must be known; and (ii) its computational complexity is high (even with non-iterative soft demapping), especially for large constellations, because of the tanh(.) and $\tanh^{-1}(.)$ operations and the computation (2.3) of the LLRs.

2.3.1.2 Min-Sum Algorithm

The min-sum algorithm (MSA) [12] is similar to the SPA, with a few adjustments to reduce the complexity and to make the computations independent of the noise variance.

- First of all, the LLRs are approximated as

$$L_{\mathrm{ch}\to n} = \min_{x' \in \mathcal{A}_{c_n=1}} \left|y - x'\right|^2 - \min_{x' \in \mathcal{A}_{c_n=0}} \left|y - x'\right|^2 \tag{2.7}$$

  Note that the above LLR approximation does not depend on the noise variance.

- Secondly, the computation at the check nodes is replaced by a simple minimum operation: Update messages $L'_{m\to n}$ from check node m to variable node n:

$$L'_{m\to n} = \left(\prod_{n' \in N(m)\setminus n} \mathrm{sgn}(L_{n'\to m}) \right) \cdot \min_{n' \in N(m)\setminus n} |L_{n'\to m}| \tag{2.8}$$

The computation of the messages $L_{n\to m}$ and $L_{n\to \mathrm{out}}$ is the same as for the SPA. This MSA has the advantage that (i) it is low in complexity and is thus very fast as compared to SPA; and (ii) no knowledge about the noise variance is required. However, the resulting error performance is worse, compared to the SPA.

2.3.1.3 Scaled Min-Sum Algorithm

The scaled min-sum algorithm (SMSA) that we consider here is the simplified variable-scaled min sum algorithm proposed in [13]. To improve the performance of the basic MSA, a scaling factor $\alpha < 1$ is introduced which increases with every decoding iteration and eventually reaches its final value 1. This scaling factor α is computed as

$$\alpha = 1 - 2^{-\lceil i/S \rceil} \tag{2.9}$$

where i is the iteration index and the scaling factor S is the only design parameter of the algorithm.

- The update of the messages $L'_{m\to n}$ from check node m to variable node n is adjusted as follows:

$$L'_{m\to n} = \alpha \cdot \left(\prod_{n' \in N(m)\setminus n} \mathrm{sgn}(L_{n'\to m}) \right) \cdot \min_{n' \in N(m)\setminus n} |L_{n'\to m}| \tag{2.10}$$

SMSA can be considered as a fine-tuned version of MSA: the decoding performance is improved, but the decoding complexity remains low compared to SPA.

2.3.2 EXIT charts

Extrinsic information transfer (EXIT) charts [14, 15] are used to analyze the iterative decoding performance and convergence behavior of LDPC codes, and to design new LDPC codes. This method is not as accurate as density evolution [16] but its accuracy is reasonably good and the complexity is much lower.

An EXIT chart consists of two EXIT functions, which are related to how the messages during the decoding iterations are processed by the variable nodes and the check nodes, respectively. The variable node EXIT function and the check node EXIT function give $I_{E,V}$ as a function of $I_{A,V}$, and $I_{E,C}$ as a function of $I_{A,C}$, respectively, where

- **Variable node EXIT function**: $I_{E,V}$ is the average (over the N coded bits) of the mutual information $I(c_n; L_{n\to m})$, between the coded bit c_n (corresponding to the considered variable node n) and the "extrinsic" message $L_{n\to m}$ leaving the variable node, when the mutual informations $I(c_n; L'_{m'\to n})$ for all $m' \in M(n)\backslash m$, between the coded bit c_n and the "a priori" message $L'_{m'\to n}$ entering the variable node, are equal to $I_{A,V}$;
- **Check node EXIT function**: $I_{E,C}$ is the average (over the N coded bits) of the mutual information $I(c_n; L'_{m\to n})$, between the coded bit c_n (correponding to a variable node n connected to the considered check node m) and the "extrinsic" message $L'_{m\to n}$ leaving the check node, when the mutual informations $I(c_{n'}; L_{n'\to m})$ for all $n' \in N(m)\backslash n$, between the coded bit $c_{n'}$ and the "a priori" message $L_{n'\to m}$ entering the check node, are equal to $I_{A,C}$.

The reader is referred to [14] and [15] for the detailed information on how the EXIT curves are obtained. The EXIT functions depend on the degree distributions of the variable nodes and the check nodes of the considered LDPC code. An example of an EXIT chart is shown in Figure 2.14 for an LDPC code of rate 2/3, that we will use in Chapter 5. The EXIT chart shows $I_{E,V}$ (ordinate) as a function of $I_{A,V}$ (abscissa) and $I_{E,C}$ (abscissa!) as a function of $I_{A,C}$ (ordinate!). The check node EXIT function typically starts in the origin (0,0) and ends in the point (1,1). The variable node EXIT function has a starting point $\left(0, I_{E,V}\big|_{I_{A,V}=0}\right)$ and ends in the point (1,1). It can be shown from (2.4) that the mutual information $I_{E,V}\big|_{I_{A,V}=0}$ equals the average I_{avg} (over the N coded bits) of the mutual information $I(c_n; L_{\text{ch}\to n})$ between the coded bit c_n and the corresponding LLR $L_{\text{ch}\to n}$; I_{avg} depends on the SNR and the considered constellation. Considering that the extrinsic message at the output of a variable node (check node) is the a priori message at the input of a check node (variable node), the exchange of extrinsic information between the variable nodes and the check nodes can be visualized as a decoding trajectory in the EXIT chart. The decoding trajectory starts at (0,I_{avg}), and ends at the crossing point of the two EXIT curves. When the two EXIT curves cross only

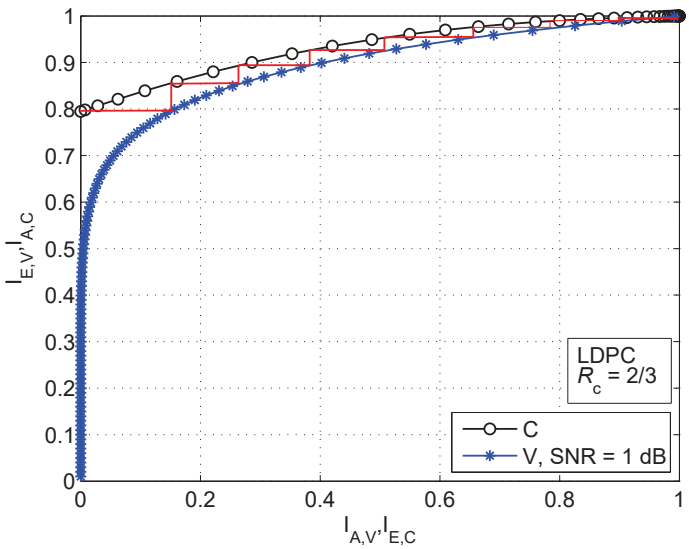

Figure 2.14: EXIT chart for the LDPC code of rate 2/3.

at the point (1,1), the "tunnel" between the two curves is said to be open; in this case the decoding trajectory converges to the point (1,1), indicating that the decoding error probability converges to zero. When the EXIT curves have an additional crossing point different from (1,1), this point corresponds to a mutual information less than 1; in this case the tunnel is said to be closed, and the decoding trajectory converges to this unwanted crossing point, indicating that the decoding error probability converges to a nonzero probability. It should be noted that although the EXIT chart analysis involves several approximations and assumes infinitely long codewords, its results describe fairly accurately the decoding behavior of finite-length LDPC codes.

2.4 Space-Time Codes

An alternative method to improve the reliability of data transmission is by application of space-time coding [17–19]. The transmitter and receiver are equipped with multiple antennas, which gives rise to a multiple-input multiple-output (MIMO) channel. A system with N_{tr} transmit and N_{r} receive antennas allows the introduction of space-time coding. Whereas an uncoded single-input single-output system, that is, $N_{\mathrm{tr}} = N_{\mathrm{r}} = 1$, provides only one link between the transmitter and receiver, the number of links provided by an orthogonal space-time block-coded (OSTBC) MIMO system equals $N_{\mathrm{r}}N_{\mathrm{tr}}$, which is referred to as the physical layer diversity of the MIMO system. As compared to a SISO

system, the larger number of links resulting from OSTBC MIMO gives rise to a considerably higher robustness against fading (because of the smaller probability that all links are simultaneously in a deep fade), and a much better error performance [18, 20].

Unlike most channel codes without spatial redundancy, the OSTBC MIMO system does not require additional bandwidth as compared to the uncoded SISO system, but comes at a substantial hardware cost that increases with the number of antennas. Optimum decoding of OSTBC MIMO reduces to linear processing and simple symbol-by-symbol detection at the receiver. In this dissertation, we will consider the Alamouti space-time code [17], which requires 2 transmit antennas (and an arbitrary number N_r of receive antennas). Alamouti space-time coding involves the transmission of two symbols during two consecutive intervals on two antennas, according to the scheme of Table 2.1, where s_1 and s_2 represent two symbols, and $(.)^*$ denotes complex conjugate. Hence, the symbol s_i reaches the receiver via $2N_\mathrm{r}$ links (i.e. N_r links for s_i and N_r links for $\pm s_i^*$).

Table 2.1: Transmitter configuration for Alamouti space-time code.

	antenna 1	antenna 2
interval 1	s_1	s_2
interval 2	$-s_2^*$	s_1^*

2.5 Reed-Solomon Codes

Reed-Solomon (RS) codes are linear error-correcting block codes presented first in a paper in 1960 by Irving Reed and Gus Salomon [21]. RS codes are frequently used in several fields of application: data storage (compact discs, blu-ray discs), QR codes, deep space communication (Voyager, Galileo), DSL,...

The RS code is defined over the Galois Field GF(2^q), which implies that a RS information symbol and code symbol represents q bits; typically, $q = 8$, in which case a symbol corresponds to a byte. The RS codewords have $N = 2^q - 1$ coded symbols.

Here we restrict our attention to systematic RS codes, so that the information part of the codeword can always be interpreted by the receiver, even in absence of a RS decoder. Per group of K information symbols, the RS(N, K) code computes $N - K$ parity symbols resulting in a systematic codeword of N coded symbols.

The RS(N, K) code is maximum distance separable (MDS), i.e., the minimum Hamming distance between codewords equals $N - K + 1$, which cannot be outperformed by any other code with the same number $N - K$ of parity

Figure 2.15: Irving Reed (12 November 1923, Washington – 11 September 2012) and Gus Salomon (27 October 1930, Brooklyn — 31 January 1996).

symbols [22, 23]. Hence, the code can correct a combination of e_1 symbol errors and e_2 symbol erasures, provided that $2e_1 + e_2 \leq N - K$. In the absence of erasures, up to $t = \left\lfloor \frac{N-K}{2} \right\rfloor$ errors can be corrected, and in the absence of errors, $\nu = N - K$ erasures can be resolved. Algebraic algorithms for decoding RS codes can be found in [23–27]. These decoding algorithms look for a valid RS codeword which differs from the received codeword in at most $\left\lfloor \frac{N-K-e_2}{2} \right\rfloor$ non-erased positions, with e_2 denoting the number of erased symbols; when no such codeword is found (because of a too large number of errors/erasures), a decoding failure is declared.

In practice, shortened systematic RS codes are often used. A shortened systematic $\mathrm{RS}(n, k)$ codeword is equivalent to a systematic $\mathrm{RS}(N, K)$ codeword with $K - k$ information symbols set to zero; these $K - k$ zero information symbols are not transmitted, yielding a $\mathrm{RS}(n, k)$ codeword with the same number of parity symbols as the original code, so that $n = k + (N - K)$. The minimum Hamming distance between codewords of the shortened RS code is the same as for the original RS code, and equals $N - K + 1 = n - k + 1$, which indicates that also the shortened RS codes are MDS; hence, $t = \left\lfloor \frac{n-k}{2} \right\rfloor$ errors can be corrected in the absence of erasures, and $\nu = n - k$ erasures can be resolved in the absence of errors. The decoding algorithm for the $\mathrm{RS}(N, K)$ code can also be used for decoding the $\mathrm{RS}(n, k)$ code, by simply adding $K - k$ zero information symbols to the received $\mathrm{RS}(n, k)$ codeword, and performing decoding of the resulting $\mathrm{RS}(N, K)$ codeword.

In the absence of erasures, a decoding error occurs when the number of symbol errors exceeds $t = \left\lfloor \frac{n-k}{2} \right\rfloor$. Assuming that symbol errors occur independently

with probability P_{s}, the decoding error probability $P_{\mathrm{e,RS}}$ is given by [22, 23]

$$P_{\mathrm{e,RS}} = \sum_{j=t+1}^{n} \frac{n!}{(n-j)!\,j!}\, P_{\mathrm{s}}^{j}\,(1-P_{\mathrm{s}})^{n-j} \tag{2.11}$$

For small P_{s}, we have $P_{\mathrm{e,RS}} \propto P_{\mathrm{s}}^{t+1}$. When the RS decoder declares a decoding failure, in some applications the k information bytes from the received codeword are forwarded unaltered to the destination; in this case the resulting byte error rate (ByteER) is obtained as [22, 23]

$$\mathrm{ByteER}_{\mathrm{RS}} = \sum_{j=t+1}^{n} \frac{(n-1)!}{(n-j)!\,(j-1)!}\, P_{\mathrm{s}}^{j}\,(1-P_{\mathrm{s}})^{n-j} \tag{2.12}$$

because the probability that a specific byte is erroneous equals j/n when the codeword has j byte errors.

When only erasures occur, the RS codeword cannot be recovered when the number of erasures exceeds $\nu = n-k$. Assuming that symbol erasures occur independently with probability $P_{\mathrm{s,e}}$, the probability $P_{\mathrm{unrec,RS}}$ of an unrecoverable codeword is given by

$$P_{\mathrm{unrec,RS}} = \sum_{j=\nu+1}^{n} \frac{n!}{(n-j)!\,j!}\, P_{\mathrm{s,e}}^{j}\,(1-P_{\mathrm{s,e}})^{n-j} \tag{2.13}$$

For small $P_{\mathrm{s,e}}$, we have $P_{\mathrm{unrec,RS}} \propto P_{\mathrm{s,e}}^{\nu+1}$.

2.6 Interleaving

Interleaving (combined with error correction) is a means to cope with burst errors that exceed the error correcting capability of the code. The information bit stream is split up into information words, which are sent to a FEC encoder. Typically, a high number (say, N_{int}) of resulting codewords is then broken up into smaller parts (bits/bytes/symbols) that are shuffled by an interleaving process such that the parts originating from any one of the original codewords are widely separated in time. At the receiver, the received parts are reshuffled ('de-interleaved') so that their original order is restored. The quantity N_{int} is referred to as the interleaving depth.

Interleaving aims at spreading each burst of errors over a large number of codewords such that the resulting number of errors in each codeword is small, and can hopefully be corrected by the code. Burst of errors can be caused by impulsive noise as discussed in Section 3.3 or by the inner decoder in a concatenated coding arrangement, as can happen with trellis decoding using a Viterbi decoder. Figure 2.16 illustrates the principle.

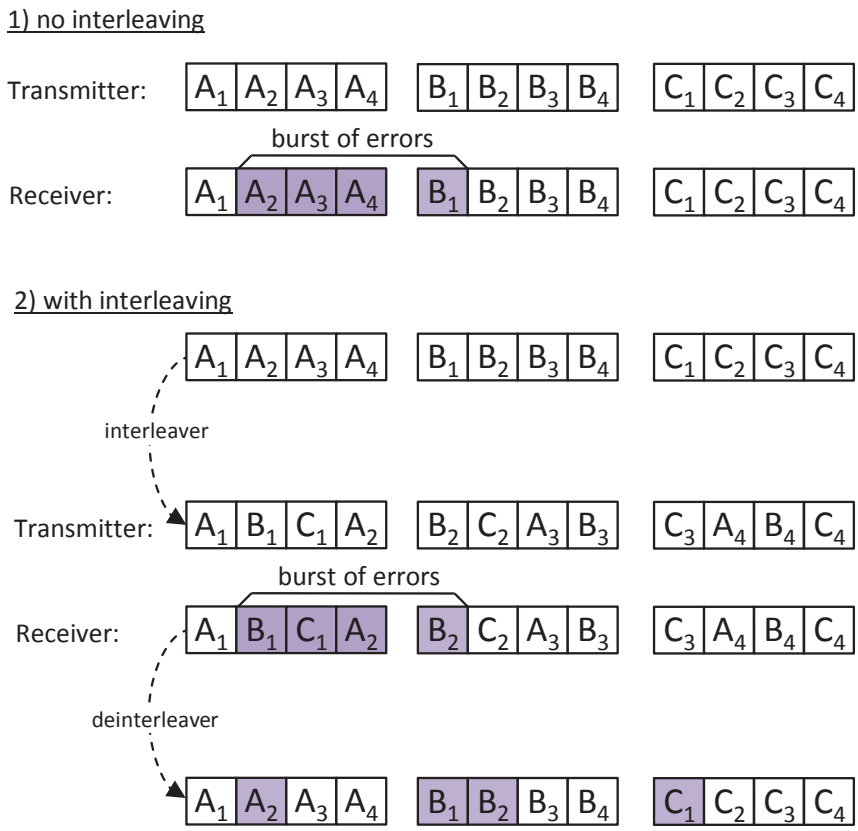

Figure 2.16: Illustration of an interleaver in the occurrence of a burst of errors.

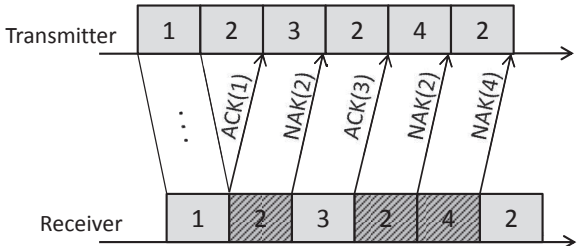

Figure 2.17: Transmitting with SR-ARQ.

The larger the interleaving depth, the more effective the interleaver. However, the interleaver introduces a latency of N_{int} codewords, so that the interleaving depth is limited by the maximum allowed latency of the particular application.

2.7 Automatic Repeat Request Protocols

The automatic repeat request (ARQ) protocol is an adaptive error-control protocol with retransmissions. In this dissertation, we will restrict our attention to selective-repeat ARQ (SR-ARQ). SR-ARQ enables the receiver to request the retransmission of a packet or data transfer unit (DTU) that has been erroneously received. Each DTU contains (apart from the data) a header including a sequence number and a cyclic redundancy check (CRC). The CRC enables the receiver to detect errors in the received DTU. If a transmitter sends a DTU to a receiver, the following scenarios may occur as illustrated in Figure 2.17:

- The DTU is delivered without errors to the receiver. A positive acknowledgment (ACK) is sent to the transmitter.
- The DTU is delivered at the receiver but the CRC fails (shaded DTUs in Figure 2.17). The receiver removes the erroneous DTU and sends a negative acknowledgment (NAK) to the transmitter with the sequence number of the erased DTU. Typically a maximum number of retransmissions per DTU is allowed, given by N_{retr}. Upon reception of this retransmission request, and if the maximum number of retransmissions is not yet reached, the transmitter issues the retransmission of a copy of the DTU.

The addition of ARQ as protection strategy leads to an increase of latency. The largest latency occurs when the maximum number of retransmissions N_{retr} is required (i.e., the first transmission and the subsequent $N_{\text{retr}} - 1$ retransmissions of the DTU are erroneous) and is given by

$$T_{\text{lat}} = T_{\text{DTU}} + N_{\text{retr}} \left(T_{\text{DTU}} + T_{\text{RTT}} \right) \tag{2.14}$$

where T_{DTU} is the transmission time of the DTU and T_{RTT} is equal to the round-trip delay time given by

$$T_{\mathrm{RTT}} = 2T_{\mathrm{prop}} + T_{\mathrm{proc}} + T_{\mathrm{ack}} \tag{2.15}$$

with T_{prop}, T_{proc} and T_{ack} respectively equal to the propagation time, the total processing time at transmitter and receiver and the transmission time of an acknowledgment message. A constraint on the maximum latency can be imposed by the application and will be enforced by limiting the maximum number of retransmissions N_{retr}.

The use of SR-ARQ requires a buffer at both the transmitter and the receiver side. The transmitter keeps a copy of each DTU in its buffer and only removes a DTU either if it receives a positive acknowledgment for the DTU or if the maximum number of retransmissions is reached. At the receiver, the buffer is needed to rearrange the DTUs such that they are passed in correct order to the higher layer.

Assuming that erroneous DTUs occur independently with probability $P_{\mathrm{e,DTU}}$, the probability $P_{\mathrm{unrec,DTU}}^{(N_{\mathrm{retr}})}$ that a DTU is erroneous after N_{retr} retransmissions is given by

$$P_{\mathrm{unrec,DTU}}^{(N_{\mathrm{retr}})} = P_{\mathrm{e,DTU}}^{N_{\mathrm{retr}}+1} \tag{2.16}$$

which is the probability that the DTU is erroneous after the first transmission and the N_{retr} subsequent retransmissions. Similarly, the BER after N_{retr} retransmissions is given by

$$\mathrm{BER}^{(N_{\mathrm{retr}})} = P_{\mathrm{e,DTU}}^{N_{\mathrm{retr}}} \mathrm{BER}^{(0)} \tag{2.17}$$

where $\mathrm{BER}^{(0)}$ is the BER corresponding to the single transmission of a DTU and $P_{\mathrm{e,DTU}}^{N_{\mathrm{retr}}}$ is the probability that the DTU is erroneous after the first transmission and the $N_{\mathrm{retr}} - 1$ subsequent retransmissions.

The average overhead (expressed in packets) introduced by ARQ is given by

$$\mathrm{ovh} = P_{\mathrm{e,DTU}} \frac{1 - P_{\mathrm{e,DTU}}^{N_{\mathrm{retr}}}}{1 - P_{\mathrm{e,DTU}}} \tag{2.18}$$

Note that $\mathrm{ovh} \approx P_{\mathrm{e,DTU}}$ for $P_{\mathrm{e,DTU}} \ll 1$.

In some situations (e.g., in the presence of impulsive noise or fading), the errors of a DTU and its copies are not necessarily independent. In these cases, a more refined analysis is necessary to determine the error performance and overhead.

ARQ is, in particular, suitable to combat occasionally large noise levels such as impulsive noise. The additional overhead and latency caused by retransmissions occur only when the noise level occasionally exceeds the error correcting capability of the code, causing the DTU to be erased by the receiver. In the absence of ARQ, using error correcting codes to cope with impulsive noise in a

static configuration of code settings would require the coding to be dimensioned to handle large noise levels that occur rather rarely, thereby introducing a large parity overhead and/or a large interleaving delay.

2.8 Performance Indicators

In this dissertation, various performance indicators of error control schemes will be considered, such as the bit error rate (BER), the word error rate (WER) and the goodput (GP).

The BER denotes the ratio of the average number of information bits which have been erroneously detected, to the total number of information bits transmitted. The WER is the ratio of the average number of codewords which have been erroneously detected, to the total number of codewords transmitted. Considering a codeword containing k information bits, and assuming that the decoder outputs valid codewords, we have $\mathrm{WER}/k \leq \mathrm{BER} \leq \mathrm{WER}$. We obtain $\mathrm{BER} = \mathrm{WER}/k$ when each erroneously decoded codeword contains exactly one information bit error, and $\mathrm{BER} = \mathrm{WER}$ when in each erroneously decoded codeword all k information bit are wrong. In a similar way, one can define related performance measures such as the byte error rate, the packet error rate,...

The performance of error control schemes depends on a number of design parameters, such as the number of parity bits in a codeword, and the constellation sizes used when mapping codewords to data symbols. These design parameters also affect the resulting information bitrate. Typically, for a given signal bandwidth, the transmitted information bitrate increases when reducing the number of parity bits and increasing the constellation sizes, at the expense of a worse error performance. Hence, there is a trade-off between information bitrate and error performance. The goodput (GP) is a performance indicator involving both the transmitted information bitrate and the error performance: the GP denotes the rate of information bits associated with correctly decoded codewords. Denoting by R_b and WER the transmitted information bitrate and the WER for a given set of design parameters, we obtain

$$\mathrm{GP} = R_\mathrm{b}(1 - \mathrm{WER}) \tag{2.19}$$

It is interesting to note that the GP does not depend on whether or not ARQ has been used. Let us denote by K_0 the number of codewords transmitted for the first time, and by K_n the number of these codewords that must be retransmitted for the nth time (because they have been erroneously decoded during the first transmission and the subsequent $n-1$ retransmissions). Table 2.2 indicates that, of the K_n codewords (re)transmitted ($n = 0, ..., N_\mathrm{retr}$), K_{n+1} of them are erroneously decoded, and the remaining $K_n - K_{n+1}$ of them are correctly decoded; note that $K_0 \geq K_1 \geq ... \geq K_{N_\mathrm{retr}+1}$. The total number of codewords transmitted (including retransmissions) equals $K_\mathrm{tot} = \sum_{n=0}^{N_\mathrm{retr}} K_n$,

Table 2.2: Determination of total number of transmitted, correct and erroneous DTUs, respectively.

	# DTUs transm.	# correct DTUs	# erroneous DTUs
1st transm.	K_0	$K_0 - K_1$	K_1
1st retransm.	K_1	$K_1 - K_2$	K_2
...	...	...	...
$N_{\text{retr}}^{\text{th}}$ retr.	$K_{N_{\text{retr}}}$	$K_{N_{\text{retr}}} - K_{N_{\text{retr}}+1}$	$K_{N_{\text{retr}}+1}$
total	$K_{\text{tot}} = \sum_{i=0}^{N_{\text{retr}}} K_i$	$K_0 - K_{N_{\text{retr}}+1}$	$K_{\text{e}} = \sum_{i=0}^{N_{\text{retr}}} K_{i+1}$

of which $K_0 - K_{N_{\text{retr}}+1}$ are correctly decoded and $K_e = \sum_{n=1}^{N_{\text{retr}}+1} K_n$ are erroneously decoded, with $K_{\text{tot}} = (K_0 - K_{N_{\text{retr}}+1}) + K_e$. Hence, the ratio of the number of information bits associated with the correctly decoded codewords to the time duration corresponding to the total number of codewords transmitted is given by

$$\frac{k(K_0 - K_{N_{\text{retr}}+1})}{\frac{k}{R_{\text{b}}} K_{\text{tot}}} = R_{\text{b}} \left(1 - \frac{K_e}{K_{\text{tot}}}\right) \tag{2.20}$$

where k is the number of information bits per codeword, R_{b} is the transmitted information bitrate and k/R_{b} is the duration of a codeword. When the number of transmitted codewords tends to infinity, the ratio K_e/K_{tot} converges to the WER, yielding $\text{GP} = R_{\text{b}}(1 - \text{WER})$, independently of the maximum number of retransmissions N_{retr}; hence, the same result is obtained for $N_{\text{retr}} = 0$, i.e., in the absence of ARQ. The difference between using ARQ and not using ARQ is in the ratio of the number of codewords correctly decoded to the number of codewords transmitted for the *first* time. This ratio equals $1 - (K_{N_{\text{retr}}+1}/K_0)$ when using ARQ and $1 - (K_1/K_0)$ in the absence of ARQ; hence, using ARQ yields the larger number of correctly decoded codewords. When decoding errors occur independently, the ratio

$$1 - \frac{K_{N_{\text{retr}}+1}}{K_0} = 1 - \prod_{i=0}^{N_{\text{retr}}} \frac{K_{i+1}}{K_i} \tag{2.21}$$

converges to $1 - \text{WER}^{N_{\text{retr}}+1}$ when the number of transmitted codewords goes to infinity.

Part I

Transmission over the DSL Channel

3 Introduction to DSL Communication

This chapter gives an introduction on the transmission over Digital Subscriber Line (DSL) channels. We provide in Section 3.1 a brief history about its origin and evolution. The transmission over DSL channel is prone to noise ingress such as crosstalk and impulsive noise. In Sections 3.2 and 3.3, we explain how these noise sources arise, and which actions are taken to maintain a good performance.

3.1 History

In this dissertation, our algorithms and protection strategies are applied to a recent DSL technology, called Fast Access to Subscriber Terminals or G.fast. This G.fast technology has been preceded by a whole series of DSL technologies, of which we present here a brief history.

DSL stands for a group of technologies that are commonly used to provide broadband internet access by transmitting digital data over telephone lines [28]. The public switched telephone netword (PSTN) was originally designed to carry voice signals. Each home is connected to the telephone network by twisted pair, invented by Alexander Graham Bell. Twisted pair consists of two isolated copper

wires which are twisted to reduce interference from external noise sources. The bandwidth of the voice signal is limited to the frequency range from 0.3 to 3.4 kHz. A number of twisted pairs is bundled in a cable binder.

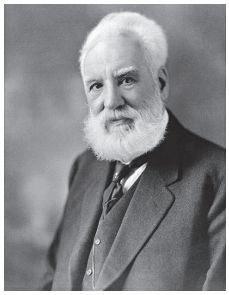

Figure 3.1: Alexander Graham Bell (March 3, 1847, Scotland – August 2, 1922).

The digital data is transmitted in a frequency band higher than the voice band. At the customer premises, the data is filtered from the voice service. A general diagram of a DSL system is depicted in Figure 3.2. The DSL loop corresponds to the twisted pair loop between the local central office and the customer premises, and a DSL modem is required at each end of the loop. The DSL access multiplexer (DSLAM) in the CO contains several DSL modems serving multiple customers. The DSLAM connects the twisted pair to both the data network and the PSTN.

We give here a brief outline of a selection of some notable DSL technologies from the past until today:

- A precursor of DSL is integrated services digital network (ISDN), offering 160 kb/s over distances up to 5.5 km [29]. The data rates are limited due to the large near-end crosstalk (NEXT) caused between signals in telephone lines traveling in opposite directions.

- High-bitrate digital subscriber line (HDSL) provides the symmetric bitrate of 1.544 Mbit/s to customers over two copper twisted pairs. It operates in the baseband and it does not allow for coexistence with telephone traffic. It is typically used to serve sites with business customers. HDSL2, a successor of HDSL, includes TCM as error correction code to enhance performance.

- NEXT is circumvented in asymmetric DSL (ADSL) by applying frequency division duplexing (FDD); the upstream and downstream frequency bands are separated. The term 'asymmetric' is used because of the different data rates in upstream (low data rate up to 896 kb/s) and downstream (high data rate up to 8 Mb/s) directions [29]. Generally, upstream transmission uses the frequency band from 25 kHz up to 138 kHz and downstream transmission goes from 138 kHz up to 1.1 MHz.

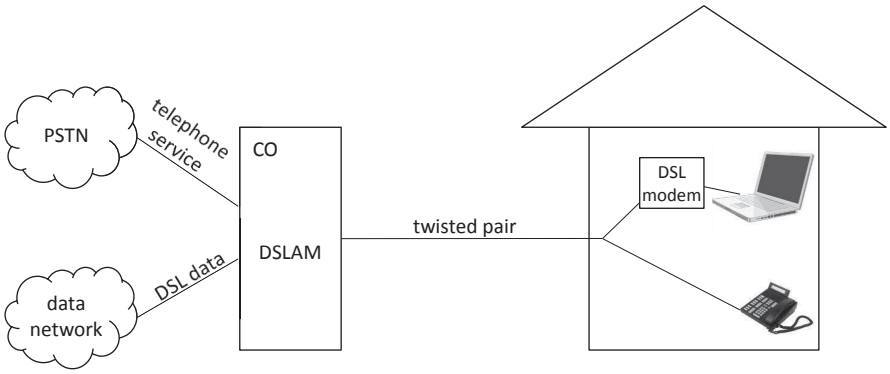

Figure 3.2: Generic DSL architecture block diagram.

In the 1990s, discrete multitone transmission (DMT) was proposed by John Cioffi, known as 'the father of DSL'. DMT transmission refers to a multi-carrier system with variable constellation sizes that makes DSL highly adaptive to the variation in loop length and external noise sources. Multi-carrier systems partition the available bandwidth into a (large) number of subchannels, called tones; each tone supports a number of bits depending on its SNR and the desired bit error probability. During each DMT symbol period, a multi-carrier signal is transmitted, which contains one constellation symbol per tone. More details about DMT transmission are given in Section 4.1.

Figure 3.3: John Cioffi (November 7, 1956, Illinois).

- Very-high bit rate DSL (VDSL) uses a bandwidth up to 12 MHz. As signal attenuation increases with frequency, the distances that can be covered become shorter. The network is upgraded by introducing optical fiber. The resulting network architecture consists of a fiber link between the

central office and a optical network unit in the street, needed to convert the optical signal to an electrical signal. The remaining distance (typically less than 1 km) to the customer premises is covered using VDSL transmission over twisted copper pair. Due to this hybrid fiber-copper connection with the central office, the length of the copper loops is shortened and higher bit rates are achieved (up to 52 Mb/s downstream and 6.4 Mb/s upstream transmission for a 300 m loop). Thanks to these higher speeds, VDSL is able to support capacity-demanding applications such as HDTV.

- VDSL2 is currently the highest-speed DSL variant available and uses the frequency band up to 30 MHz on loop lengths similar to those used with VDSL [5]. A signal coordination technique called vectoring can be used to cancel the far-end crosstalk (FEXT), caused by signals on telephone lines traveling in the same direction. VDSL2 typically achieves a throughput of 30-40 Mb/s in the presence of crosstalk, which is increased beyond 100 Mb/s if the crosstalk is removed by vectoring [30, 31]. The VDSL2-based DMT modulation uses for frequencies up to 12 MHz, a tone spacing of 4.3125 kHz and a DMT symbol rate of 4000 symbols/s. For frequencies up to 30 MHz, a tone spacing of 8.625 kHz is used and a DMT symbol rate of 8000 symbols/s.

Users continue to increase the bit rate demand for many different applications. As the core network mainly consists of fiber, the twisted pair access network to the core network has become a bottleneck that limits the available data rates, as the fiber network has a capacity far beyond the capacity of the copper network [32]. The ultimate goal is fiber access for every user, also called fiber-to-the-home (FTTH). The bandwidth offered by FTTH would be enough to even satisfy the most demanding user. But the cost of the installation is considerable; the twisted pair access network is the major part of the installed network and replacing it with fiber is very expensive. A full FTTH network will not be achieved in the near future, the investments are rather spread over a longer time period. Therefore, internet service providers expressed interest to further boost the capacity of copper access networks to enable fiber-like speeds without full FTTH deployment.

- Fast Access to Subscriber Terminals (G.fast), is a copper technology for fiber-to-the-distribution-point (FTTdp) deployments with copper loops of 30 m to 250 m, targeting a 1 Gb/s downstream bitrate on very short loops and several hundred Mb/s for longer loops (e.g., 500 Mb/s and 200 Mb/s on 100 m and 200 m, respectively) [33–35]. Optical fiber connects the central office with distribution points in street cabinets or basements of buildings, located very close to the user as visualized in Figure 3.4.

  The existing copper telephone wiring provides the connection from the distribution points to the homes. This FTTdp architecture is continuing

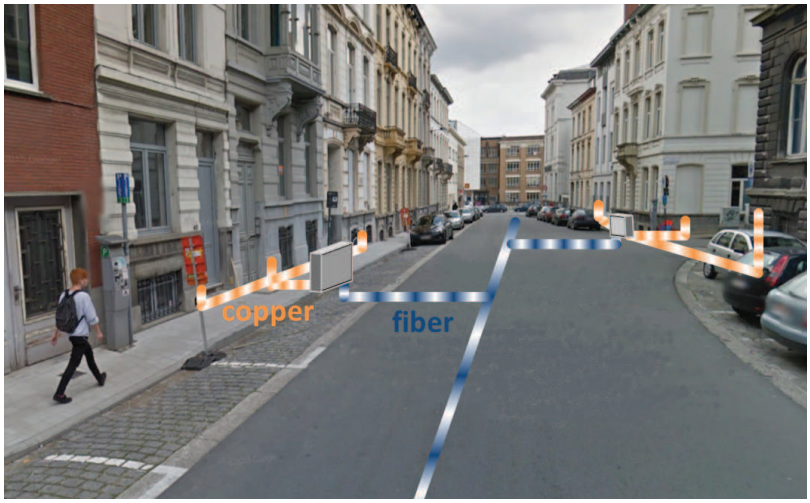

Figure 3.4: FTTdp architecture with a fiber core network and a twisted pair access network.

the gradual deployment of fiber, making the fiber termination move closer to the home, and hence reducing the length of the twisted pair segment. On short DSL loops, higher bit rates can be achieved through higher signal bandwidths, because signal attenuation increases less with frequency as compared to longer loops.

The G.fast-based DMT modulation uses frequencies up to 212 MHz, a tone spacing of 51.75 kHz and a DMT symbol rate of 48000 symbols/s. It carries up to 12 (coded) bits per tone in the DMT symbol. As opposed to previous DSL technologies, G.fast uses time division duplexing (TDD) to separate downstream and upstream traffic.

In the next chapters of this first part of the dissertation, we will apply our results and algorithms to this type of channel.

3.2 Crosstalk and Precoding

As multiple copper pairs are usually bundled in binders, the transmission suffers from crosstalk among twisted pairs. Crosstalk is the primary noise source in DSL systems [29]. Vectoring can substantially cancel the crosstalk by means of proper signal coordination [36]. In downstream communications, the transmitting modems are co-located in the same distribution point, allowing for the joint processing of the signals before transmission. The customer premises of the end users are geographically separated, which makes joint processing of the received signals infeasible. Before transmission, the signals are precoded in such

a way that the combination of the precoding and crosstalk cancels out. The resulting gain can be significant; e.g., VDSL2 typically achieves a throughput of 30-40 Mb/s in the presence of crosstalk; this throughput is increased beyond 100 Mb/s if the crosstalk is canceled by means of precoding [30, 31].

In the case of linear precoding (LP), the columns of the precoding matrix are made proportional to the columns of the inverse of the channel matrix. In the spectrum range of VDSL2, which is limited to 30 MHz, the channel matrix and the corresponding precoder are diagonally dominant, in which case the linear precoder is near-optimal in terms of data rate [37]. However, as in G.fast the spectrum band goes up to 212 MHz, the channel matrix is no longer diagonally dominant [38]; in this case the large off-diagonal elements of the linear precoding matrix (i.e. the additional power used by the crosstalk precompensation signals) increase the transmit power spectral density (PSD), hence deteriorating the resulting data rate under a given transmit PSD constraint. This problem can be circumvented by using nonlinear precoding (NLP), which reduces the transmit PSD as compared to LP by introducing a modulo operation at the transmitting side [36]; however, NLP requires also a modulo operation to be applied at the receiving side, which to some extent degrades the error performance of the receiver since the modulo operation creates additional nearest neighbours for the outer constellation points. This will be elaborated in Chapter 4.

3.3 Impulsive Noise

Impulsive noise (IN) manifests itself as occasional short noise pulses of significant energy. This type of noise is typically caused by electrical power switching originating from human activity near the user's premises. E.g., the switching of a fluorescent lamp, an electrical traction motor like used in trams, trains and trolley buses, telephone ringing... Also weather conditions such as lightning can cause the occurrence of IN. The shape, duration, energy and inter arrival time of the pulses vary widely. There is no commonly accepted model for the noise impulses, in this dissertation we will use a simple model, the parameters of which are fitted to the experimental data reported in [39] and [40].

The reduction of the DMT symbol duration in G.fast increases the vulnerability to impulsive noise as compared to VDSL2, because an individual noise impulse is likely to hit consecutive DMT symbols in G.fast. Error-control techniques that have proven useful for long DMT symbols (as used in VDSL2) might be inadequate for short DMT symbols. This motivates the investigation of error-control techniques for short DSL loops.

When using a static configuration of error correcting codes to cope with non stationary impulsive noise, the coding must be dimensioned to handle large noise levels that occur only occasionally, thereby introducing a large parity overhead and/or a large interleaving delay. This can be avoided by using an adaptive error-control protocol such as a retransmission protocol (automatic

repeat request (ARQ) protocol). ARQ enables the receiver to request the retransmission of a packet or data transfer unit (DTU) that has been erroneously received. With ARQ, the additional overhead and latency caused by retransmissions occur only when the noise level occasionally exceeds the error correcting capability of the code. For a VDSL2 system with frequencies used up to 17 MHz and $T_{\mathrm{DMT}} = 250\,\mu s$, it has been shown in [41] that using ARQ instead of interleaving can considerably reduce the latency.

The reader is referred to Section 4.5 for more information about the statistical model used for the IN.

4 DSL System Description

In this chapter, we describe the system model as it is used throughout the first part of this dissertation about the transmission over the DSL channel. Section 4.1 provides a mathematical description of the communication over the channel that suffers from crosstalk. Section 4.2 gives an overview of the constellation types used. The crosstalk in DSL transmission is tackled by means of precoding. Section 4.3 addresses linear precoding (LP), which is adequate in moderate crosstalk environments. When the crosstalk is strong, we might require nonlinear precoding (NLP), which is presented in Section 4.4. Section 4.5 provides a statistical model for the impulsive noise.

4.1 DSL Channel Model

We consider the downlink transmission from a distribution point (DP) to N_{u} users at different locations; each user is connected to the DP by means of a twisted pair cable. Denoting by $x_i(t)$ and $y_i(t)$ the signals transmitted and

received on the ith twisted pair cable ($i = 1, ..., N_{\text{u}}$), we have

$$y_i(t) = \sum_{j=1}^{N_{\text{u}}} \int h_{i,j}(t-u) x_j(u) du + n_i(t) \tag{4.1}$$

where $h_{i,j}(t)$ is the impulse response from the input of the jth cable to the output of the ith cable, and $n_i(t)$ represents stationary Gaussian noise[1]; the noise terms corresponding to different user indices are statistically independent. The impulse responses $h_{i,i}(t)$ correspond to the direct channels, whereas the impulse responses $h_{i,j}(t)$ with $j \neq i$ denote the crosstalk channels. We assume causal impulse responses with a finite duration not exceeding T_{ch}, i.e., $h_{i,j}(t) = 0$ for $t < 0$ and for $t > T_{\text{ch}}$.

The N_{u} transmitters at the DP make use of discrete multitone (DMT) modulation which includes a cyclic prefix to avoid interference between successive DMT symbols. Basically, the signal $x_i(t)$ consists of blocks of duration T_{DMT}; these blocks are referred to as DMT symbols. During the lth DMT symbol, a vector $\mathbf{x}_i(l) = \left(x_i^{(1)}(l), ..., x_i^{(N_{\text{t}})}(l)\right)$ of N_{t} zero-mean uncorrelated components is transmitted, with each component modulating a different subcarrier or tone. The frequencies of the N_{t} tones are multiples of a frequency spacing F. The quantities T_{DMT} and F are related by $T_{\text{DMT}} = T_{\text{CP}} + \frac{1}{F}$, with $T_{\text{CP}} > T_{\text{ch}}$; typically, we select F such that $T_{\text{CP}}F \ll 1$ (in G.fast, we have $T_{\text{DMT}} = 1/48000$ s $= 20.833$ μs and $F = 51.75$ kHz, so that $T_{\text{CP}} = 1.510$ μs and $T_{\text{CP}}F = 0.078$). The real-valued transmit signal during the lth DMT symbol interval is given by

$$x_i(t) = \sqrt{\frac{2}{T_{\text{DMT}}}} \sum_{k=1}^{N_{\text{t}}} \Re\left(x_i^{(k)}(l) e^{j2\pi kFt}\right) \qquad t \in [lT_{\text{DMT}}, (l+1)T_{\text{DMT}}) \tag{4.2}$$

with kF denoting the frequency of the kth tone. As the signal segments in the intervals $t \in [lT_{\text{DMT}}, lT_{\text{DMT}} + T_{\text{CP}})$ and $t \in [(l+1)T_{\text{DMT}} - T_{\text{CP}}, (l+1)T_{\text{DMT}})$ are identical, the first segment of duration T_{CP} is referred to as the cyclic prefix of the DMT symbol.

The normalization factor $\sqrt{\frac{2}{T_{\text{DMT}}}}$ in (4.2) ensures that the transmit energy on the kth tone during the lth DMT symbol equals $\mathbb{E}[|x_i^{(k)}(l)|^2]$. Assuming that $\mathbb{E}[|x_i^{(k)}(l)|^2]$ does not depend on the DMT symbol index l, the transmit power associated with the kth tone equals $\mathbb{E}[|x_i^{(k)}|^2]/T_{\text{DMT}}$, and the aggregate transmit power (sum over all tones) on the ith cable equals $\sum_{k=1}^{N_{\text{t}}} \mathbb{E}[|x_i^{(k)}|^2]/T_{\text{DMT}}$. The (two-sided) power spectral density $S_{x_i}(f)$ of $x_i(t)$ is given by

$$S_{x_i}(f) = \frac{1}{2} \sum_{k=1}^{N_{\text{t}}} \mathbb{E}[|x_i^{(k)}|^2] \left(S(f - kF) + S(k + F)\right) \tag{4.3}$$

[1]The occurrence of impulsive noise will be considered in Section 4.5.

where

$$S(f) = \left(\frac{\sin(\pi f T_{\mathrm{DMT}})}{\pi f T_{\mathrm{DMT}}}\right)^2$$

is concentrated near $f = 0$, and has a mainlobe width of $2/T_{\mathrm{DMT}}$. Taking into account that $T_{\mathrm{CP}}F \ll 1$ yields $S(mF) \ll S(0)$ for any nonzero integer m, it follows that $S_{x_i}(kF) = S_{x_i}(-kF) \approx \frac{1}{2}\mathbb{E}[|x_i^{(k)}|^2]$. Hence, the one-sided power spectral density (defined as $2S_{x_i}(f)$ for $f \geq 0$ and zero otherwise) at $f = kF$ is accurately approximated by $\mathbb{E}[|x_i^{(k)}|^2]$.

The response $y_{i,j}(t)$ of the ith cable to the transmitted DMT signal $x_j(t)$ is obtained by convolving $x_j(t)$ with the impulse response $h_{i,j}(t)$. At the receiver, the cyclic prefix of each DMT symbol is discarded, and only the remaining part (of duration $\frac{1}{F}$) of each DMT symbol is processed; hence, the useful part of the lth DMT symbol at the receiver is obtained as

$$y_{i,j}(t) = \int h_{i,j}(t-u)x_j(u)\mathrm{d}u \quad t \in [lT_{\mathrm{DMT}} + T_{\mathrm{CP}}, (l+1)T_{\mathrm{DMT}}) \tag{4.4}$$

Because of discarding the cyclic prefix at the receiver, the transmitted DMT symbols having an index less than l do not contribute to $y_{i,j}(t)$ in (4.4) during the considered interval; similarly, transmitted DMT symbols with an index larger than l do not contribute either, because the impulse responses are causal. Hence, only the lth transmitted DMT symbol must be taken into account, so that the integration in (4.4) can be restricted to the interval $u \in [lT_{\mathrm{DMT}}, (l+1)T_{\mathrm{DMT}})$. As the duration of $h_{i,j}(t)$ is less than T_{CP}, (4.4) can be transformed into

$$y_{i,j}(t) = \sqrt{\frac{2}{T_{\mathrm{DMT}}}} \sum_{k=1}^{N_{\mathrm{t}}} \Re\left(H_{i,j}(kF)x_j^{(k)}(l)e^{j2\pi kFt}\right) \quad t \in [lT_{\mathrm{DMT}} + T_{\mathrm{CP}}, (l+1)T_{\mathrm{DMT}}) \tag{4.5}$$

where $H_{i.j}(f)$ is the Fourier transform of $h_{i,j}(t)$. The received signal $y_i(t)$ during the useful part of the lth DMT symbol is then obtained as

$$y_i(t) = \sum_{j=1}^{N_{\mathrm{u}}} y_{i,j}(t) + n_i(t) \tag{4.6}$$

where $y_{i,j}(t)$ is given by (4.5).

The receiver performs a demodulation according to

$$y_i^{(k)}(l) = \sqrt{2T_{\mathrm{DMT}}}F \int_{lT_{\mathrm{DMT}}+T_{\mathrm{CP}}}^{(l+1)T_{\mathrm{DMT}}} y_i(t)e^{-j2\pi kFt}\mathrm{d}t \tag{4.7}$$

Taking into account that $F\int_{1/F} e^{j2\pi mFt}\mathrm{d}t = \delta_m$ with δ_m denoting the Kro-

necker delta function, we obtain

$$y_i^{(k)}(l) = \sum_{j=1}^{N_u} H_{i,j}(kF) x_j^{(k)}(l) + n_i^{(k)}(l) \tag{4.8}$$

Considering that $n_i(t)$ is zero-mean stationary Gaussian noise with (two-sided) power spectral density $S_{n_i}(f)$, the complex random variables $n_i^{(k)}(l)$ are zero-mean Gaussian and circular symmetric with $\mathbb{E}\left[n_i^{(k)}(l)\left(n_i^{(k')}(l)\right)^*\right] = N_{0,i}^{(k)}\delta_{k-k'}$, where $N_{0,i}^{(k)} = 2(1+T_{\mathrm{CP}}F)S_{n_i}(kF)$. Hence, the noise contributions $n_i^{(k)}(l)$ are independent across tones, with variances $N_{0,i}^{(k)}$. When $n_i(t)$ is *white* Gaussian noise, $S_{n_i}(f)$ does not depend on f, so that $N_{0,i}^{(k)}$ becomes independent of the tone index k (and will be denoted $N_{0,i}$; in this case, the noise contributions $n_i^{(k)}(l)$ are i.i.d. across tones, with variance $N_{0,i}$).

As there is no interference between consecutive DMT symbols, we will drop the DMT symbol index for notational convenience. The resulting observation model can be compactly represented as

$$\mathbf{y}^{(k)} = \mathbf{H}^{(k)}\mathbf{x}^{(k)} + \mathbf{n}^{(k)} \tag{4.9}$$

where the superscript k refers to the tone index ($k = 1, ..., N_{\mathrm{t}}$), the received vector $\mathbf{y}^{(k)}$, the transmitted vector $\mathbf{x}^{(k)}$ and the noise vector $\mathbf{n}^{(k)}$ have dimension $N_{\mathrm{u}} \times 1$, and the channel matrix $\mathbf{H}^{(k)}$ has dimension $N_{\mathrm{u}} \times N_{\mathrm{u}}$. The ith components ($i = 1, ..., N_{\mathrm{u}}$) of the vectors $\mathbf{x}^{(k)}$, $\mathbf{y}^{(k)}$ and $\mathbf{n}^{(k)}$ represent the quantities $x_i^{(k)}$, $y_i^{(k)}$ and $n_i^{(k)}$, respectively. The element $H_{i,j}^{(k)}$ of $\mathbf{H}^{(k)}$ equals $H_{i,j}(kF)$. The components of $\mathbf{n}^{(k)}$ are zero-mean statistically independent circular symmetric Gaussian random variables with $\mathbb{E}\left[|n_i^{(k)}|^2\right] = N_{0,i}^{(k)}$.

In a practical implementation, the modulation (4.2) is replaced by first obtaining the N-point IFFT of the vector

$$\left(0, x_i^{(1)}(l), ..., x_i^{(N_{\mathrm{t}})}(l), 0, ...0, \left(x_i^{(N_{\mathrm{t}})}(l)\right)^*, ..., \left(x_i^{(1)}(l)\right)^*\right)$$

consisting of a zero, the N_t constellation symbol values of the considered DMT symbol, $N - 2N_t - 1$ zeroes, and the complex conjugates of the N_t constellation symbol values in reversed order (implying that N must satisfy the inequality $N \geq 2N_t + 1$). Next, a vector of $N + \nu$ samples is constructed by prepending the last ν samples of the IFFT to the beginning of the IFFT, and finally these $N + \nu$ samples are applied at a rate $1/T$ to a transmit filter. The sampling interval T satisfies $T = 1/(NF)$ with F denoting the tone spacing, and ν should be selected such that $\nu T > T_{\mathrm{ch}}$; the resulting durations of the DMT symbol and the cyclic prefix are $T_{\mathrm{DMT}} = (N + \nu)T$ and $T_{\mathrm{CP}} = \nu T$, respectively. The demodulation operation (4.7) is replaced by applying $y_i(t)$ to a receive filter, sampling the receiver filter output at the rate $1/T$, keeping the N samples that

correspond to the useful part of the considered DMT symbol, and performing an N-point FFT on these samples. Typically, N is an integer power of 2. The resulting observation model can again be represented by (4.9).

Each of the N_u transmit signals is subject to a maximum power spectral density (PSD) and a maximum aggregate transmission power (ATP) constraint per line. These constraints translate into

$$\mathbb{E}\left[|x_i^{(k)}|^2\right] \leq E_\mathrm{PSD}^{(k)} \quad \text{all}\,(i,k) \tag{4.10}$$

$$\sum_{k=1}^{N_\mathrm{t}} \mathbb{E}\left[|x_i^{(k)}|^2\right] \leq E_\mathrm{ATP} \qquad \text{all}\ i \tag{4.11}$$

where $E_\mathrm{PSD}^{(k)}$ is the maximum one-sided PSD at frequency kF, and $E_\mathrm{ATP} = P_\mathrm{ATP}/T_\mathrm{DMT}$, with P_ATP denoting the maximum aggregate transmit power.

4.2 Constellation Types

The information bit sequences to be sent to the N_u users are (possibly) encoded for increased robustness against transmission errors, and the resulting coded bits are mapped to constellation symbols. The constellation symbol for the ith user sent on the kth tone is denoted $a_i^{(k)}$, and belongs to a constellation $\mathcal{A}_i^{(k)}$ with $M_i^{(k)} = 2^{b_i^{(k)}}$ points. For mathematical convenience, we consider normalized constellations, i.e., the constellation symbols have unit energy: $\mathbb{E}[|a_i^{(k)}|^2] = 1$; a non-normalized constellation symbol with energy $E_i^{(k)}$ is then denoted as $\sqrt{E_i^{(k)}}a_i^{(k)}$. The number of points in the constellations $\mathcal{A}_i^{(k)}$ and the constellation symbol energies $E_i^{(k)}$ are design parameters. For further use, we introduce the $N_\mathrm{u} \times 1$ vector $\mathbf{a}^{(k)}$, the ith component of which equals $a_i^{(k)}$ $(i = 1, ..., N_\mathrm{u})$ and the $N_\mathrm{u} \times N_\mathrm{u}$ diagonal matrix $\mathbf{E}^{(k)}$ with $\left(\mathbf{E}^{(k)}\right)_{i,i} = E_i^{(k)}$; the vector of non-normalized symbols is then given by $\left(\mathbf{E}^{(k)}\right)^{1/2}\mathbf{a}^{(k)}$.

If we simply set $x_i^{(k)} = \sqrt{E_i^{(k)}}a_i^{(k)}$, the users would suffer from crosstalk, because of the nonzero off-diagonal elements of the channel matrix $\mathbf{H}^{(k)}$ from (4.9). As the users are at different locations, they cannot cooperate to cancel this interference. Therefore, we allow the N_u transmitters at the DP to cooperate in order to avoid interference at the users' receivers. This transmitter-side signal coordination (or *vectoring*) is referred to as precoding; the corresponding $x_i^{(k)}$ is a function of $\mathbf{a}^{(k)}$ rather than only $a_i^{(k)}$.

We restrict our attention to quadrature amplitude modulation (QAM) constellations C_M containing $M = 2^b$ points, with $b \in \{1, 2, ..., b_\mathrm{max}\}$; b represents the number of coded bits per constellation symbol. The real and imaginary

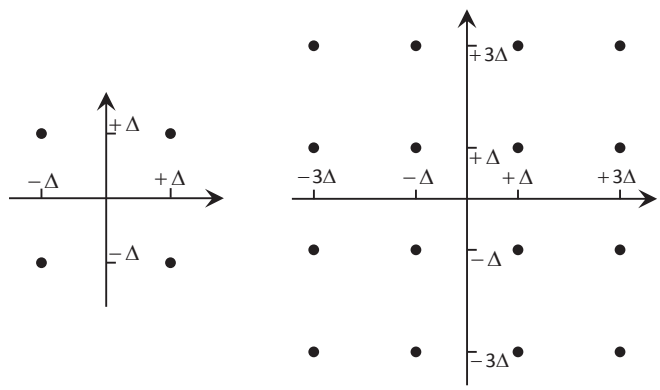

Figure 4.1: Constellations for 4-QAM and 16-QAM.

parts of the constellation points are odd multiples of the quantity Δ. Now we discuss the different QAM constellations in more detail.

4.2.1 Square-QAM

A square-QAM constellation consists of $M = 2^b$ points, where b is an even integer. The set of constellation symbols for the M-QAM constellation ($M = 2^b$) is given by $C_M = \{((2m+1) + j(2n+1))\Delta; m, n \in \{-\sqrt{M}/2, \ldots, \sqrt{M}/2 - 1\}\}$, with minimum distance d between constellation points given by $d = 2\Delta$. The average energy per symbol associated with this constellation is equal to

$$\begin{aligned} E_M &= \frac{1}{M} \sum_{a \in C_M} |a|^2 \\ &= \frac{2}{3} (M-1) \Delta^2 \end{aligned} \tag{4.12}$$

For a normalized constellation, we have $E_M = 1$, yielding

$$d^2 = \frac{6}{M-1} \tag{4.13}$$

Figure 4.1 shows the constellations for 4-QAM ($b = 2$) and 16-QAM ($b = 4$).

4.2.2 Cross-QAM

A cross-QAM constellation consists of $M = 2^b$ points, where b is an odd integer larger than 3. The set of constellation points for the 2^b-QAM constellation is given by $C_M = C_{M,1} \cup C_{M,2} \cup C_{M,3}$ with

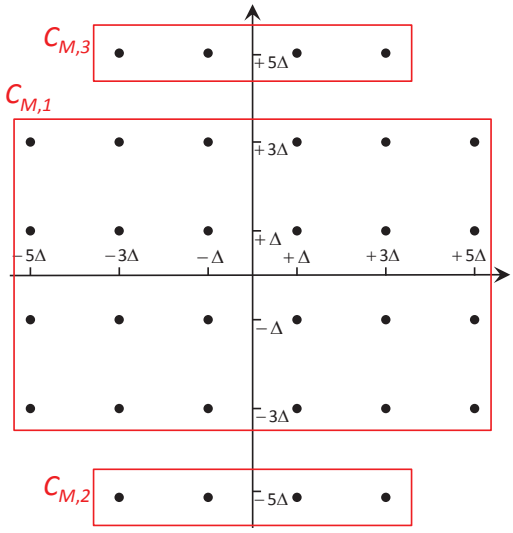

Figure 4.2: Constellation for 32-QAM.

$$
\begin{aligned}
C_{M,1} = \{((2m+1) \quad &+ \quad j(2n+1))\Delta; \\
m \quad &\in \quad \{-\frac{3}{4}M', \ldots, \frac{3}{4}M'-1\}, n \in \{-\frac{1}{2}M', \ldots, \frac{1}{2}M'-1\}\} \\
C_{M,2} = \{((2m+1) \quad &+ \quad j(2n+1))\Delta; \\
m \quad &\in \quad \{-\frac{1}{2}M', \ldots, \frac{1}{2}M'-1\}, n \in \{-\frac{3}{4}M', \ldots, -\frac{1}{2}M'-1\}\} \\
C_{M,3} = \{((2m+1) \quad &+ \quad j(2n+1))\Delta; \\
m \quad &\in \quad \{-\frac{1}{2}M', \ldots, \frac{1}{2}M'-1\}, n \in \{\frac{1}{2}M', \ldots, \frac{3}{4}M'-1\}\}
\end{aligned}
$$

where we have set $M' = \sqrt{\frac{M}{2}}$ for conciseness; these constellations are referred to as cross-QAM. The minimum distance d between constellation points is given by $d = 2\Delta$. For the average energy per symbol we obtain

$$E_M = \frac{2}{3}\left(\frac{31}{32}M - 1\right)\Delta^2 \tag{4.14}$$

For a normalized constellation, we have $E_M = 1$, yielding

$$d^2 = \frac{6}{\frac{31}{32}M - 1} \tag{4.15}$$

Figure 4.2 shows the 32-QAM constellation ($b = 5$).

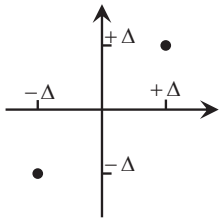

Figure 4.3: Constellation for 2-QAM.

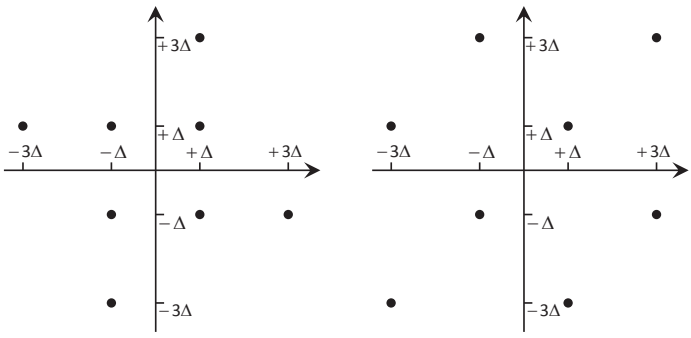

Figure 4.4: Two types of 8-QAM constellations: 8-QAM-VDSL (left) and 8-QAM-G.fast (right).

4.2.3 2-QAM

The 2-QAM constellation ($b = 1$) is shown in Figure 4.3, with $C_2 = \{(1 + j)\Delta, -(1 + j)\Delta\}$. The minimum distance d between constellation points is given by $d = 2\sqrt{2}\Delta$. For the average energy per symbol, we obtain

$$E_2 = 2\Delta^2 \tag{4.16}$$

For a normalized constellation, we have $E_2 = 1$, yielding $d^2 = 4$.

4.2.4 8-QAM

We consider two types of 8-QAM constellations, referred to as 8-QAM-VDSL and 8-QAM-G.fast, which are shown in Figure 4.4.

The 8-QAM-VDSL constellation is part of the DSL standards since the introduction of DMT transmission in ADSL and still used in the VDSL standard [5]. The minimum distance d between constellation points is equal to $d = 2\Delta$. For the average energy per symbol, we obtain

$$E_{8,\mathrm{VDSL}} = 6\Delta^2 \tag{4.17}$$

For a normalized constellation, we have $E_{8,\mathrm{VDSL}} = 1$, yielding $d^2 = 2/3$.

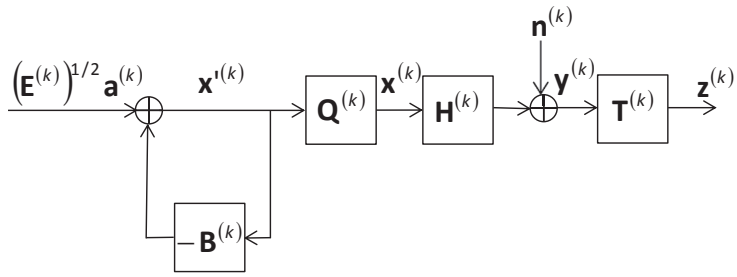

Figure 4.5: Block diagram of the system with linear precoder.

The 8-QAM-G.fast constellation has been proposed for the G.fast standard [42]. The minimum distance d between constellation points is equal to $d = 2\sqrt{2}\Delta$. For the average energy per symbol, we obtain

$$E_{8,\text{G.fast}} = 10\Delta^2 \tag{4.18}$$

For a normalized constellation, we have $E_{8,\text{G.fast}} = 1$, yielding $d^2 = 4/5$.

4.3 Linear Precoding

4.3.1 Derivation of Linear Precoding Structure

In the case of linear precoding (LP), we counteract the crosstalk generated by the off-diagonal elements of $\mathbf{H}^{(k)}$ by applying at the transmitter a linear transformation to the non-normalized constellation symbol vector $\left(\mathbf{E}^{(k)}\right)^{1/2}\mathbf{a}^{(k)}$, i.e., the transmit vector $\mathbf{x}^{(k)}$ is given by $\mathbf{x}^{(k)} = \mathbf{P}^{(k)}\left(\mathbf{E}^{(k)}\right)^{1/2}\mathbf{a}^{(k)}$, where $\mathbf{P}^{(k)}$ is a $N_\text{u} \times N_\text{u}$ precoding matrix.

We select the precoding matrix $\mathbf{P}^{(k)}$ such that the crosstalk is completely canceled at the users' receivers; this is referred to as zero-forcing (ZF) linear precoding. First we consider the QR decomposition of $\left(\mathbf{H}^{(k)}\right)^\text{H}$, i.e., $\left(\mathbf{H}^{(k)}\right)^\text{H} = \mathbf{Q}^{(k)}\mathbf{R}^{(k)}$, where $\mathbf{Q}^{(k)}$ is a unitary matrix ($\left(\mathbf{Q}^{(k)}\right)^\text{H}\mathbf{Q}^{(k)} = \mathbf{Q}^{(k)}\left(\mathbf{Q}^{(k)}\right)^\text{H} = \mathbf{I}$) and $\mathbf{R}^{(k)}$ is an upper triangular matrix. From this, we obtain the following decomposition of $\mathbf{H}^{(k)}$: $\mathbf{H}^{(k)} = \mathbf{L}^{(k)}\left(\mathbf{Q}^{(k)}\right)^\text{H}$, with $\mathbf{L}^{(k)} = \left(\mathbf{R}^{(k)}\right)^\text{H}$ denoting a lower triangular matrix. Next, we decompose $\mathbf{L}^{(k)}$ as $\mathbf{L}^{(k)} = \mathbf{D}_\mathbf{L}^{(k)}(\mathbf{I} + \mathbf{B}^{(k)})$, where $\mathbf{D}_\mathbf{L}^{(k)}$ is a diagonal matrix with $\left(\mathbf{D}_\mathbf{L}^{(k)}\right)_{i,i} = L_{i,i}^{(k)}$, $\mathbf{I}$ denotes the identity matrix, and $\mathbf{B}^{(k)}$ is a lower triangular matrix with zeroes on its diagonal. This yields $\mathbf{H}^{(k)} = \mathbf{D}_\mathbf{L}^{(k)}(\mathbf{I} + \mathbf{B}^{(k)})\left(\mathbf{Q}^{(k)}\right)^\text{H}$. Now we select the precoding matrix as

$\mathbf{P}^{(k)} = \mathbf{Q}^{(k)}(\mathbf{I}+\mathbf{B}^{(k)})^{-1}$; this gives rise to $\mathbf{H}^{(k)}\mathbf{P}^{(k)} = \mathbf{D}_{\mathbf{L}}^{(k)}$, which indicates that the cascade of the precoder $\mathbf{P}^{(k)}$ and the channel $\mathbf{H}^{(k)}$ acts as a diagonal channel matrix $\mathbf{D}_{\mathbf{L}}^{(k)}$, so that crosstalk is indeed eliminated.

Because of the particular structure of the precoding matrix, the transmit vector can be represented as $\mathbf{x}^{(k)} = \mathbf{Q}^{(k)}\mathbf{x}'^{(k)}$, with $(\mathbf{I}+\mathbf{B}^{(k)})\mathbf{x}'^{(k)} = \left(\mathbf{E}^{(k)}\right)^{1/2}\mathbf{a}^{(k)}$. Considering that $\mathbf{B}^{(k)}$ is lower triangular with zero diagonal, the elements of $\mathbf{x}'^{(k)}$ can be computed recursively as follows:

$$x_1'^{(k)} = \sqrt{E_1^{(k)}}a_1^{(k)} \tag{4.19}$$

$$x_i'^{(k)} = \sqrt{E_i^{(k)}}a_i^{(k)} - \sum_{j=1}^{i-1} B_{i,j}^{(k)} x_j'^{(k)} \qquad (i>1) \tag{4.20}$$

The resulting system with linear precoding is shown in Figure 4.5.

At the input of the receiver of the ith user we obtain

$$y_i^{(k)} = L_{i,i}^{(k)}\sqrt{E_i^{(k)}}a_i^{(k)} + n_i^{(k)} \tag{4.21}$$

The symbol energy $E_i^{(k)}$ is a design parameter, which should be selected such that the resulting $\mathbb{E}\left[|x_i^{(k)}|^2\right]$ satisfy the constraints (4.10) and (4.11). Define $\mathbf{N}_0^{(k)}$ as the diagonal matrix with $\left(\mathbf{N}_0^{(k)}\right)_{i,i} = N_{0,i}^{(k)}$. For mathematical convenience, we apply to $\mathbf{y}^{(k)}$ a reversible transformation $\mathbf{z}^{(k)} = \mathbf{T}^{(k)}\mathbf{y}^{(k)}$, consisting of a rotation and a scaling of the components $y_i^{(k)}$, i.e., $z_i^{(k)} = y_i^{(k)}\exp(-j\arg(L_{i,i}^{(k)}))/\sqrt{N_{0,i}^{(k)}}$; hence, $\mathbf{T}^{(k)}$ is a diagonal matrix with

$$(\mathbf{T}^{(k)})_{i,i} = \exp(-j\arg(L_{i,i}^{(k)}))/\sqrt{N_{0,i}^{(k)}} \tag{4.22}$$

Because of the reversibility of the transformation, $\mathbf{z}^{(k)}$ and $\mathbf{y}^{(k)}$ contain the same information about the constellation symbol vector $\mathbf{a}^{(k)}$. The resulting $z_i^{(k)}$ can be decomposed as

$$z_i^{(k)} = \sqrt{\mathrm{SNR}_i^{(k)}}a_i^{(k)} + w_i^{(k)} \tag{4.23}$$

where

$$\mathrm{SNR}_i^{(k)} = \frac{|L_{i,i}^{(k)}|^2 E_i^{(k)}}{N_{0,i}^{(k)}} \tag{4.24}$$

denotes the SNR on the kth tone at the input of the ith receiver, and $\mathbb{E}[|w_i^{(k)}|^2] = 1$.

4.3.2 Transmit Energy

Considering that the components of $\mathbf{a}^{(k)}$ are uncorrelated, the energy of the components of the transmit vector $\mathbf{x}^{(k)} = \mathbf{P}^{(k)}\mathbf{a}^{(k)}$ at the output of the precoding matrix can be computed as

$$\mathbb{E}\left[|x_i^{(k)}|^2\right] = \sum_{j=1}^{N_\mathrm{u}} |P_{i,j}^{(k)}|^2 E_j^{(k)} \tag{4.25}$$

Hence, in view of (4.25), when some components of the precoding matrix $\mathbf{P}^{(k)}$ have a large magnitude, some of the $E_i^{(k)}$ should be small in order to satisfy the constraints (4.10)-(4.11), yielding small values of the corresponding $\mathrm{SNR}_i^{(k)}$; in this case, only small constellation sizes can be used to achieve a satisfactory error performance at small $\mathrm{SNR}_i^{(k)}$, which limits the information bitrate. This problem can be circumvented by using nonlinear precoding, which is described in the next section.

4.3.3 Alternative Linear Precoding Structure

An alternative formulation of the zero-forcing linear precoding involves decomposing the channel matrix as $\mathbf{H}^{(k)} = \mathbf{D}_\mathbf{H}^{(k)}\left(\mathbf{I} + \mathbf{C}^{(k)}\right)$ [36], where $\mathbf{D}_\mathbf{H}^{(k)}$ is a diagonal matrix with $\left(\mathbf{D}_\mathbf{H}^{(k)}\right)_{i,i} = H_{i,i}^{(k)}$, $\mathbf{I}$ denotes the identity matrix, and the diagonal elements of $\mathbf{C}^{(k)}$ are zero; the off-diagonal elements of $\mathbf{C}^{(k)}$ are indicative of the coupling between the twisted pair cables. The precoder is selected as $\mathbf{P}'^{(k)} = \left(\mathbf{I} + \mathbf{C}^{(k)}\right)^{-1}$, which yields $\mathbf{H}^{(k)}\mathbf{P}'^{(k)} = \mathbf{D}_\mathbf{H}^{(k)}$. Denoting in this alternative formulation the non-normalized symbol related the kth tone and the ith user as $\sqrt{E_i'^{(k)}}a_i^{(k)}$, it is easily verified that the original formulation and the alternative formulation are equivalent (i.e., they yield the same transmit and receive vectors $\mathbf{x}^{(k)}$ and $\mathbf{y}^{(k)}$, so they satisfy the same transmit power constraints and achieve the same error performance) when $\sqrt{E_i'^{(k)}} = \sqrt{E_i^{(k)}}L_{i,i}^{(k)}/H_{i,i}^{(k)}$. The alternative system avoids the QR decomposition of $\left(\mathbf{H}^{(k)}\right)^\mathrm{H}$, but requires the inversion of $\mathbf{I} + \mathbf{C}^{(k)}$. In the case of small to moderate crosstalk, this inverse can be approximated by truncating the infinite series expansion

$$\left(\mathbf{I} + \mathbf{C}^{(k)}\right)^{-1} = \sum_{l\geq 0}(-1)^l\left(\mathbf{C}^{(k)}\right)^l$$

to only a few terms, e.g., $\left(\mathbf{I} + \mathbf{C}^{(k)}\right)^{-1} \approx \mathbf{I} - \mathbf{C}^{(k)}$. However, such approximation is no longer accurate in the presence of strong crosstalk; moreover, when $\mathbf{C}^{(k)}$ has one or more eigenvalues with a magnitude exceeding 1, the above infinite series does not converge.

Table 4.1: Values of d^2 for the 2^b-QAM constellations with $b = 1, \ldots, 12$.

b	1	2	3 (VDSL)	3 (G.fast)	4	5	6
d^2	4	2	2/3	4/5	2/5	1/5	2/21

b	7	8	9	10	11	12
d^2	2/41	2/85	2/165	2/341	2/661	2/1365

4.3.4 Comparison of Different Constellations

Considering (4.23), we will show in Chapter 5 that the error performance for uncoded transmission and for TCM is mainly determined by the minimum distance between the scaled constellation symbols $\sqrt{\mathrm{SNR}_i^{(k)}}a_i^{(k)}$, with $a_i^{(k)} \in \mathcal{A}_i^{(k)}$; this distance equals $\sqrt{\mathrm{SNR}_i^{(k)}}d_i^{(k)}$, where $d_i^{(k)}$ is the minimum distance of the normalized constellation $\mathcal{A}_i^{(k)}$. For a given channel matrix $\mathbf{H}^{(k)}$, let us replace for all (i,k) the normalized constellations $\mathcal{A}_i^{(k)}$ (with minimum distance $d_i^{(k)}$) and the symbol energies $E_i^{(k)}$ by the normalized constellations $\mathcal{A}_i'^{(k)}$ (with minimum distance $d_i'^{(k)}$) and the symbol energies $E_i'^{(k)}$. In order that the transmitted energies $\mathbb{E}\left[|x_i^{(k)}|^2\right]$ from (4.25) remain the same, we select $E_i'^{(k)} = E_i^{(k)}$, yielding $\mathrm{SNR}_i'^{(k)} = \mathrm{SNR}_i^{(k)}$ (see (4.24)). Hence, compared to the constellation $\mathcal{A}_i^{(k)}$ scaled by the factor $\sqrt{\mathrm{SNR}_i^{(k)}}$, the minimum distance associated with the constellation $\mathcal{A}_i'^{(k)}$ scaled by the factor $\sqrt{\mathrm{SNR}_i'^{(k)}}$ is increased by the factor $d_i'^{(k)}/d_i^{(k)}$. Table 4.1 summarizes the values of d^2 for the various constellations. Comparing 8-QAM-VDSL (with $d^2 = 2/3$) and 8-QAM-G.fast (with $d^2 = 4/5$), it follows that the latter has an advantage of $10\log(6/5) = 0.79$ dB over the former, in terms of Euclidean distance of the scaled constellations.

4.4 Nonlinear Precoding

4.4.1 Derivation of Nonlinear Precoding Structure

We consider zero-forcing nonlinear precoding (NLP) of the Tomlinson-Harashima type [36]. This type of precoding makes use of modulo operations at the transmitter.

For any real-valued u and any positive divisor A, we define u modulo A as $[u]_A = u + lA$, where l is the unique integer such that $-A/2 \leq u + lA < A/2$. This definition is easily extended to complex-valued u, by defining $[u]_A = [\Re(u)]_A + j[\Im(u)]_A$. For $\alpha > 0$, we have the relation $[\alpha u]_{\alpha A} = \alpha[u]_A$. The

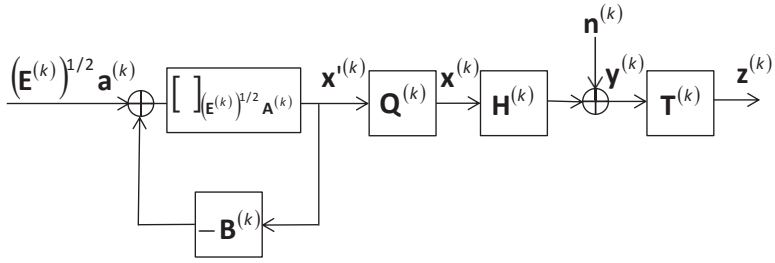

Figure 4.6: Block diagram of the system with nonlinear precoder.

modulo operation on a complex-valued vector $\mathbf{u}$ corresponds to the component-wise application of the modulo operation; denoting by A_i the divisor related to u_i, the ith component of $\mathbf{u}$, we have $([\mathbf{u}]_{\mathbf{A}})_i = [u_i]_{A_i}$, with $(\mathbf{A})_i = A_i$.

The only difference between the nonlinear precoder and the linear precoder from Figure 4.5 is in the way $\mathbf{x}'^{(k)}$ is obtained. In the case of NLP, we have

$$x_1'^{(k)} = \left[\sqrt{E_1^{(k)}}a_1^{(k)}\right]_{\sqrt{E_1^{(k)}}A_1^{(k)}} \tag{4.26}$$

$$x_i'^{(k)} = \left[\sqrt{E_i^{(k)}}a_i^{(k)} - \sum_{j=1}^{i-1} B_{i,j}^{(k)} x_j'^{(k)}\right]_{\sqrt{E_i^{(k)}}A_i^{(k)}} \qquad (i > 1) \tag{4.27}$$

The selection of the quantity $A_i^{(k)}$ will be discussed shortly. Because of the modulo operation, the real and imaginary parts of $x_i'^{(k)}$ have a magnitude not exceeding $\sqrt{E_i^{(k)}}A_i^{(k)}/2$, which indicates that the value of $A_i^{(k)}$ affects $\mathbb{E}[|x_i'^{(k)}|^2]$. Alternatively, $x_i'^{(k)}$ can be represented as

$$x_1'^{(k)} = \sqrt{E_1^{(k)}}\left(a_1^{(k)} + l_1^{(k)}A_1^{(k)}\right) \tag{4.28}$$

$$x_i'^{(k)} = \sqrt{E_i^{(k)}}\left(a_i^{(k)} + l_i^{(k)}A_i^{(k)}\right) - \sum_{j=1}^{i-1} B_{i,j}^{(k)} x_j'^{(k)} \qquad (i > 1) \tag{4.29}$$

where $l_i^{(k)} = l_{R,i}^{(k)} + jl_{I,i}^{(k)}$, and $l_{R,i}^{(k)}$ and $l_{I,i}^{(k)}$ are the unique integers such that the real and imaginary parts of the right-hand sides of (4.28) and (4.29) are in the interval $\left[-\sqrt{E_i^{(k)}}A_i^{(k)}/2, \sqrt{E_i^{(k)}}A_i^{(k)}/2\right)$. Considering the similarity between (4.28)-(4.29) and (4.19)-(4.20), it follows that the receive vector $\mathbf{y}^{(k)}$ is

determined by

$$y_i^{(k)} = L_{i,i}^{(k)} \sqrt{E_i^{(k)}} \left(a_i^{(k)} + l_i^{(k)} A_i^{(k)} \right) + n_i^{(k)} \tag{4.30}$$

Applying to $\mathbf{y}^{(k)}$ the same transformation matrix $\mathbf{T}^{(k)}$ as in the case of LP yields $\mathbf{z}^{(k)} = \mathbf{T}^{(k)}\mathbf{y}^{(k)}$, with

$$z_i^{(k)} = \sqrt{\mathrm{SNR}_i^{(k)}} \left(a_i^{(k)} + l_i^{(k)} A_i^{(k)} \right) + w_i^{(k)} \tag{4.31}$$

where $\mathrm{SNR}_i^{(k)}$ is given by (4.24) and $\mathbb{E}[|w_i^{(k)}|^2] = 1$. The corresponding system with NLP is shown in Figure 4.6.

4.4.2 Selection of $A_i^{(k)}$

When $a_i^{(k)}$ belongs to an $M_i^{(k)}$-QAM constellation $\mathcal{A}_i^{(k)}$, then $a_i^{(k)} + l_i^{(k)} A_i^{(k)}$ belongs to a periodically extended $M_i^{(k)}$-QAM constellation (extension with period $A_i^{(k)}$ in the horizontal and vertical directions), which we denote as $\mathcal{A}_{\mathrm{ext},i}^{(k)}$. The value of $A_i^{(k)}$ affects the Euclidean distance between points of the constellation $\mathcal{A}_{\mathrm{ext},i}^{(k)}$. We take the smallest $A_i^{(k)}$ for which the minimum Euclidean distance in $\mathcal{A}_{\mathrm{ext},i}^{(k)}$ is the largest possible, i.e., equal to the minimum Euclidean distance in the original constellation $\mathcal{A}_i^{(k)}$. A larger $A_i^{(k)}$ would maintain the minimum Euclidean distance in $\mathcal{A}_{\mathrm{ext},i}^{(k)}$, but yield an increase of $\mathbb{E}[|x_i'^{(k)}|^2]$ and of the resulting transmit power. When $M_i^{(k)} = M = 2^b$, we take $A_i^{(k)} = A$, according to

$$A = \begin{cases} 4\Delta & b = 1 \\ 8\Delta & b = 3\,(\mathrm{VDSL, G.fast}) \\ 2\sqrt{M}\Delta & \text{square} \\ 3\sqrt{\frac{M}{2}}\Delta & \text{cross} \end{cases} \tag{4.32}$$

which is illustrated in Figures 4.7-4.11 for the different constellation types. Note that the 2^b-QAM constellations are enclosed by a square with side A from (4.32), so that in (4.26) the modulo operation can be dropped and, equivalently, in (4.28) we have $l_1^{(k)} = 0$.

4.4.3 Transmit Energy

For given k, the quantities $x_i'^{(k)}$ are well approximated as independent random variables which, for $i > 1$, are uniformly distributed inside a square with side $\sqrt{E_i^{(k)}} A_i^{(k)}$ [36]. This yields

$$\mathbb{E}\left[|x_i^{(k)}|^2\right] = \sum_{j=1}^{N_\mathrm{u}} |Q_{i,j}^{(k)}|^2 \mathbb{E}\left[|x_j'^{(k)}|^2\right]$$

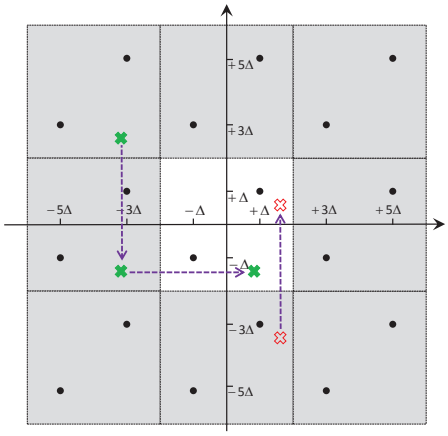

Figure 4.7: Illustration of the modulo operation for 2-QAM.

where

$$\mathbb{E}\left[|x_i'^{(k)}|^2\right] = \begin{cases} E_1^{(k)} & i = 1 \\ \rho_i^{(k)} E_i^{(k)} & i > 1 \end{cases} \tag{4.33}$$

and $\rho_i^{(k)} = \left(A_i^{(k)}\right)^2/6$. When $M_i^{(k)} = M = 2^b$, we obtain $\rho_i^{(k)} = \rho$, with

$$\rho = \begin{cases} \frac{4}{3} & b = 1 \\ \frac{16}{9} & b = 3, \text{VDSL} \\ \frac{16}{15} & b = 3, \text{G.fast} \\ \frac{M}{M-1} & \text{square} \\ \frac{9}{8}\frac{M}{\frac{31}{32}M-1} & \text{cross} \end{cases} \tag{4.34}$$

The factor $\rho_i^{(k)}$ is the ratio of the energy of $x_i'^{(k)}$ to the energy of $\sqrt{E_i^{(k)}}a_i^{(k)}$; we note from (4.34) that the maximum value of this ratio amounts to 16/9 (2.5 dB), which occurs for 8-QAM-VDSL. For square and cross constellations, ρ decreases with increasing M, and converges to 1 (0 dB) and 36/31 (0.65 dB), respectively.

4.4.4 Comparison of Constellations

Considering (4.31), we will show in Chapter 5 that the error performance for uncoded transmission and for TCM is mainly determined by the minimum distance between the scaled constellation symbols $\sqrt{\text{SNR}_i^{(k)}}a_i^{(k)}$, with $a_i^{(k)} \in \mathcal{A}_i^{(k)}$; this distance equals $\sqrt{\text{SNR}_i^{(k)}}d_i^{(k)}$, where $d_i^{(k)}$ is the minimum distance of the normalized constellation $\mathcal{A}_i^{(k)}$. For a given channel matrix $\mathbf{H}^{(k)}$,

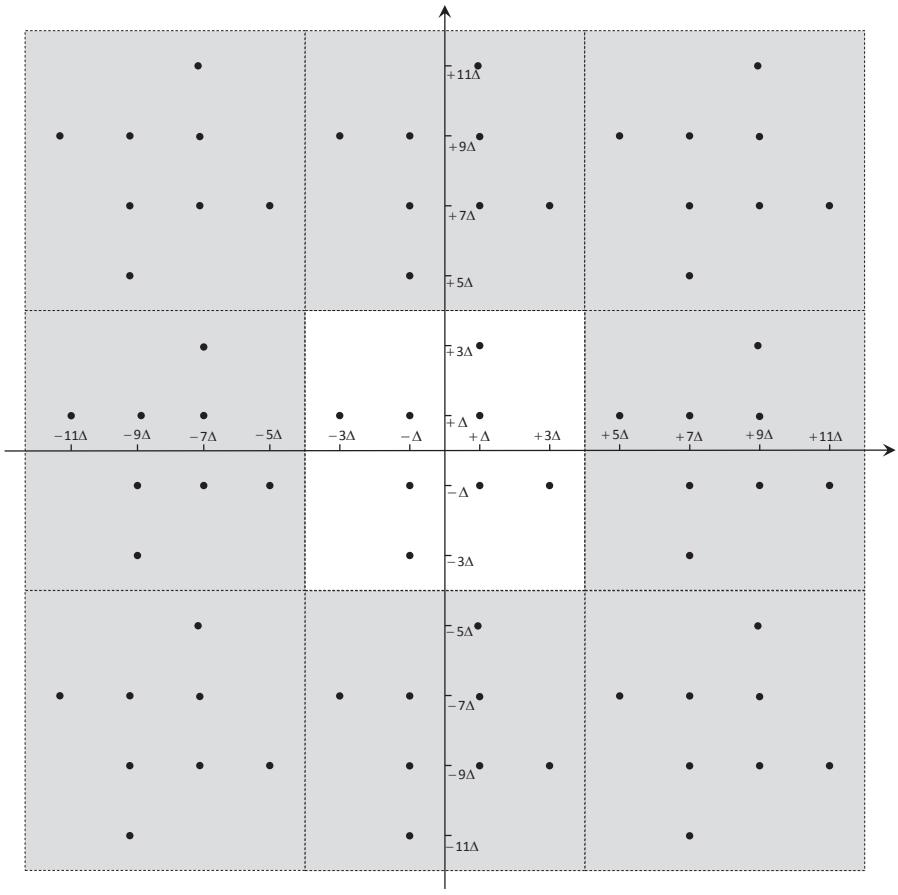

Figure 4.8: Illustration of the modulo operation for the 8-QAM-VDSL constellation with $A = 8\Delta$.

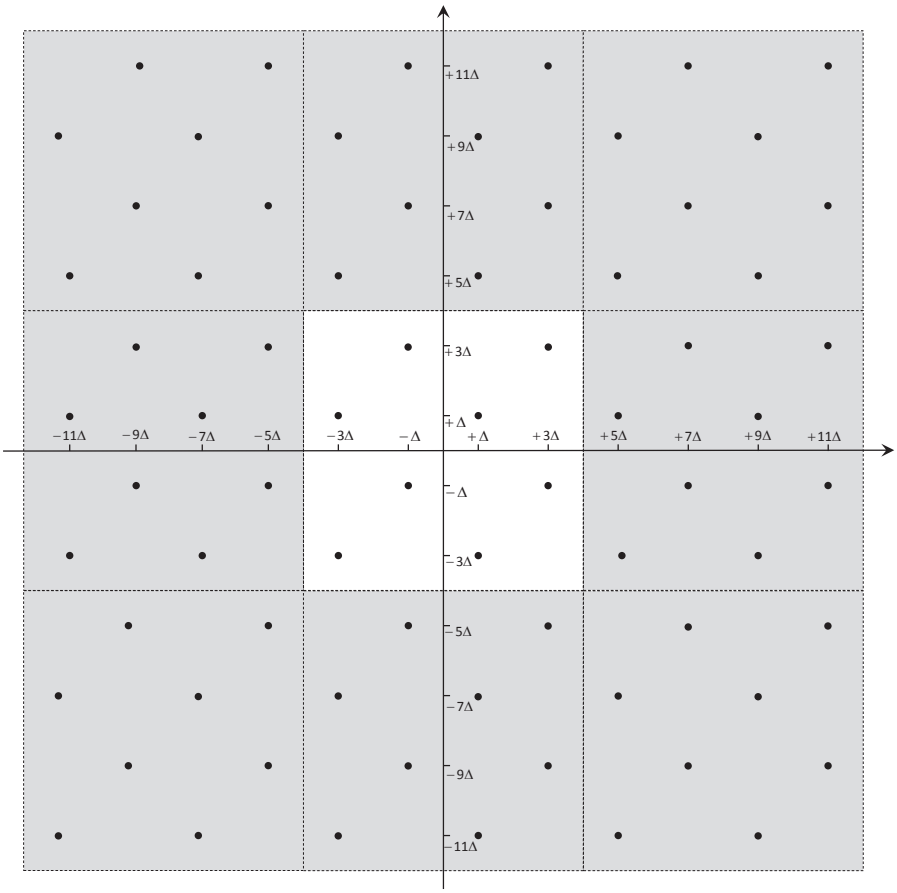

Figure 4.9: Illustration of the modulo operation for the 8-QAM-G.fast constellation.

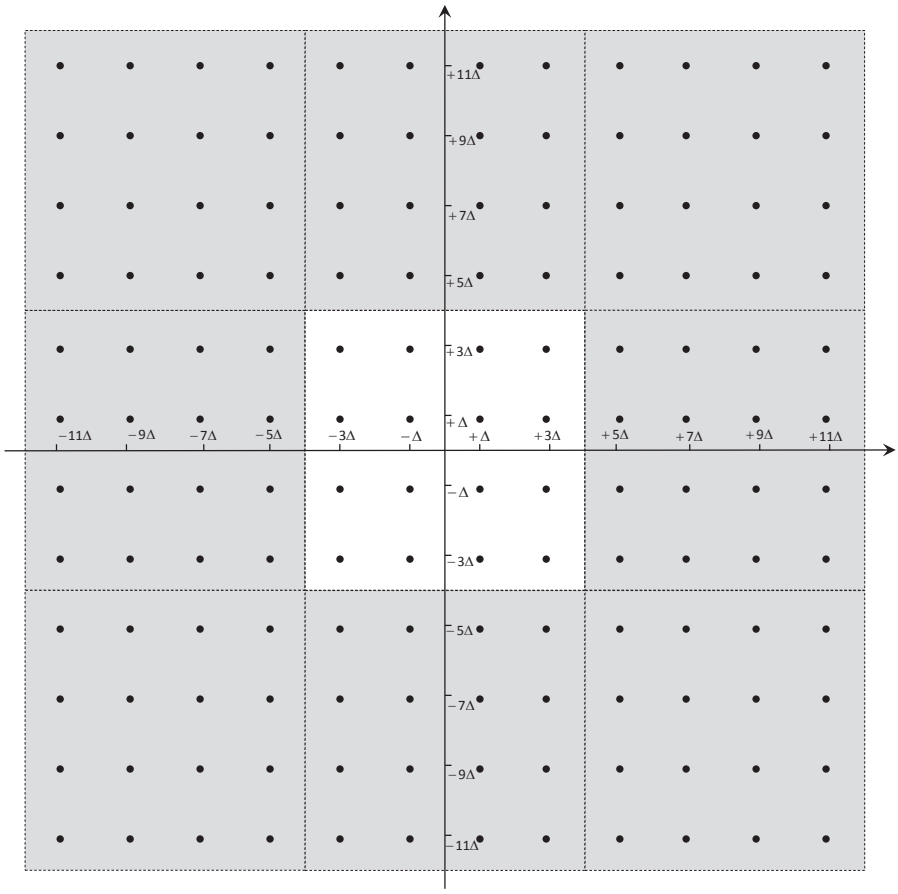

Figure 4.10: Illustration of the modulo operation for square constellations.

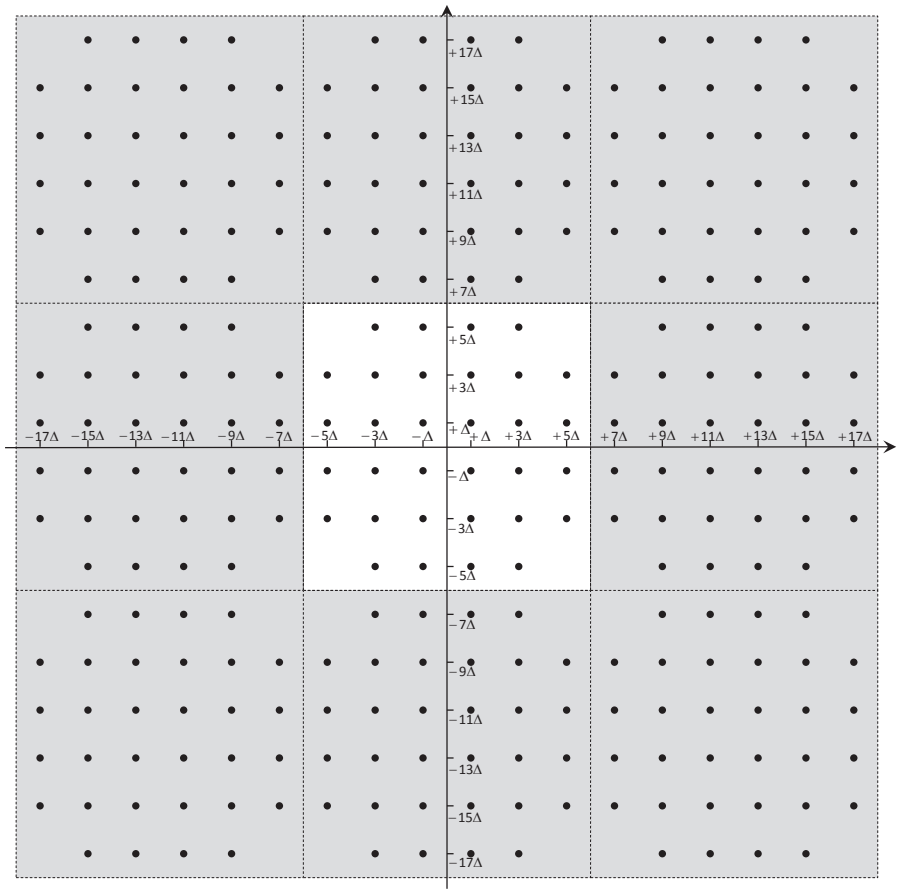

Figure 4.11: Illustration of the modulo operation for cross constellations.

Table 4.2: Values of d^2/ρ for the 2^b-QAM constellations with $b = 1, \ldots, 12$.

b	1	2	3 (VDSL)	3 (G.fast)	4	5	6
d^2/ρ	3	3/2	3/8	3/4	3/8	1/6	3/32

b	7	8	9	10	11	12
d^2/ρ	1/24	3/128	1/96	3/512	1/384	3/2048

let us replace for all (i, k) the normalized constellations $\mathcal{A}_i^{(k)}$ (with minimum distance $d_i^{(k)}$) and the symbol energies $E_i^{(k)}$ by the normalized constellations $\mathcal{A}_i'^{(k)}$ (with minimum distance $d_i'^{(k)}$) and the symbol energies $E_i'^{(k)}$. In order that the transmitted energies $\mathbb{E}\left[|x_i^{(k)}|^2\right]$ from (4.33) remain the same, we select $E_1'^{(k)} = E_1^{(k)}$ and $E_i'^{(k)} = (\rho_i^{(k)}/\rho_i'^{(k)})E_i^{(k)}$ for $i > 0$, yielding $\mathrm{SNR}_1'^{(k)} = \mathrm{SNR}_1^{(k)}$ and $\mathrm{SNR}_i'^{(k)} = (\rho_i^{(k)}/\rho_i'^{(k)})\mathrm{SNR}_i^{(k)}$ (see (4.24)). Hence, compared to the constellation $\mathcal{A}_i^{(k)}$ scaled by the factor $\sqrt{\mathrm{SNR}_i^{(k)}}$, the minimum distance associated with the constellation $\mathcal{A}_i'^{(k)}$ scaled by the factor $\sqrt{\mathrm{SNR}_i'^{(k)}}$ is increased by the factor $\left(d_i'^{(k)}/\sqrt{\rho_i'^{(k)}}\right) / \left(d_i^{(k)}/\sqrt{\rho_i^{(k)}}\right)$ for $i > 1$. Table 4.2 summarizes the values of d^2/ρ for the various constellations. Comparing 8-QAM-VDSL (with $d^2/\rho = 3/8$) and 8-QAM-G.fast (with $d^2/\rho = 3/4$), it follows that the latter has an advantage of $10\log(2) = 3$ dB over the former in terms of Euclidean distance of the scaled constellations; this is a strong motivation to use 8-QAM-G.fast instead of 8-QAM-VDSL in the case of NLP.

Strictly speaking, for 8-QAM-VDSL, $A = 6\Delta$ (instead of $A = 8\Delta$ from (4.32)) is the smallest value that maintains $d^2 = 2/3$ for the extended constellation, as shown in Figure 4.12; this yields $\rho = 1$ and $d^2/\rho = 2/3$ (instead of $\rho = 16/9$ and $d^2/\rho = 3/8$) for 8-QAM-VDSL, so that the corresponding advantage of 8-QAM-G.fast over 8-QAM-VDSL in the case of NLP becomes $10\log((3/4)\cdot(3/2))\cdot(3/2)) = 0.51$ dB (instead of 3 dB for $A = 8\Delta$). In the case of uncoded transmission (where the error performance is determined by the minimum Euclidean distance in the extended constellation), $A = 6\Delta$ would be the proper choice for 8-QAM-VDSL. However, for trellis-coded modulation, we will point out in Section 5.2 that the error performance is determined by the minimum Euclidean distance in extended *subsets* of the original constellation. For 8-QAM-VDSL it can be verified that taking $A = 6\Delta$ actually reduces the minimum Euclidean distance in the extended subset as compared to the original subset; it turns out that $A = 8\Delta$ is needed in order not to reduce the distance in the extended subsets. For the other QAM constellations, the minimum value of A (given in (4.32)) that maintains the distance in the extended constellation

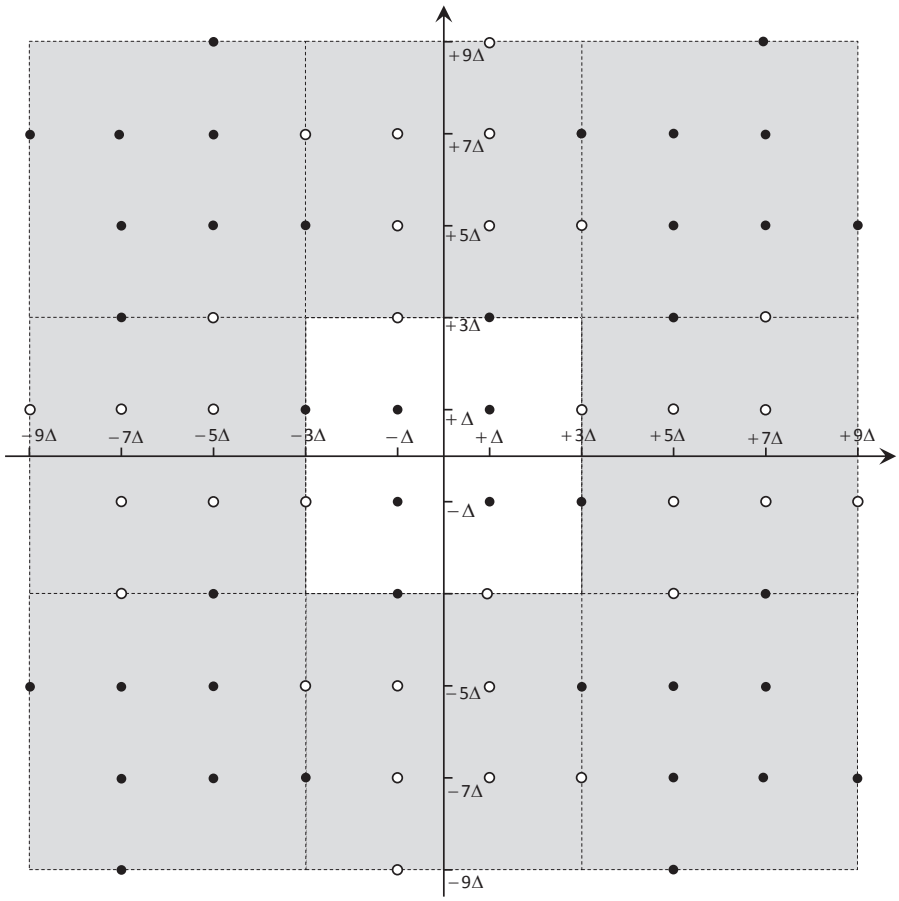

Figure 4.12: Illustration of the modulo operation for the 8-QAM-VDSL constellation with $A = 6\Delta$.

also maintains the distance in the extended subsets.

4.5 Impulsive Noise

Accurate modeling of impulsive noise (IN) is very difficult, because IN measurements made on different locations and/or at different times can show strong variations of the IN statistics regarding pulse duration, inter-arrival times, pulse energy and spectral content. For this reason, we will consider in this dissertation a simple IN model with only a small number of parameters. Although simple, the model captures the main features of IN, and, therefore, serves to assess the robustness of various error control schemes against IN, rather than to accurately predict system performance in an actual IN environment.

We model the IN as gated white Gaussian noise [43–45], which is switched on and off at DMT symbol boundaries, and is added to the received signal $y_i(t)$ from (4.1). Hence, a DMT symbol is either not hit by IN, or hit by IN during the entire DMT symbol duration; such model implies that the average duration of a noise impulse exceeds the DMT symbol duration, which is the case in G.fast. Taking IN into account, the demodulator output $y_i^{(k)}(l)$ from (4.8) should be replaced by

$$y_i^{(k)}(l) = \sum_{j=1}^{N_u} H_{i,j}(kF) x_j^{(k)}(l) + n_{\text{act},i}^{(k)}(l) \tag{4.35}$$

In (4.35),

$$n_{\text{act},i}^{(k)}(l) = n_i^{(k)}(l) + I(l) n_{\text{imp},i}^{(k)}(l) \tag{4.36}$$

denotes the instantaneous noise on the considered tone, consisting of the stationary noise contribution $n_i^{(k)}(l)$ and the IN contribution $I(l)n_{\text{imp},i}^{(k)}(l)$. We have $I(l) = 1$ when the considered DMT symbol interval is hit by IN and $I(l) = 0$ otherwise. The quantity $n_{\text{imp},i}^{(k)}(l)$ is zero-mean Gaussian and circular symmetric with $\mathbb{E}\left[n_{\text{imp},i}^{(k)}(l)\left(n_{\text{imp},i}^{(k')}(l)\right)^*\right] = N_{0,\text{imp},i}\delta_{k-k'}$; note that the variance $N_{0,\text{imp},i}$ of $n_{\text{imp},i}^{(k)}(l)$ does not depend on the tone index k, because of the assumption that the IN is white when present. The instantaneous variance $N_{0,\text{act},i}^{(k)}(l)$ of $n_{\text{act},i}^{(k)}(l)$ is given by $N_{0,\text{act},i}^{(k)}(l) = N_{0,i}^{(k)} + I(l)N_{0,\text{imp},i}$. The noise term $n_{\text{act},i}^{(k)}(l)$ is distributed according to a mixture of two zero-mean Gaussian distributions, with variances $N_{0,i}^{(k)}$ (when $I(l) = 0$) and $N_{0,i}^{(k)} + N_{0,\text{imp},i}$ (when $I(l) = 1$), respectively; the instantaneous noise contributions $n_{\text{act},i}^{(k)}(l)$ are independent across tones.

When considering the lth DMT symbol interval in the case of LP or NLP, we have to replace in Sections 4.3 and 4.4 the noise variance $N_{0,i}^{(k)}$ by the instantaneous noise variance $N_{0,\text{act},i}^{(k)}(l)$ in order to take the IN into account. Consequently, the observation equations (4.23) and (4.31) for LP and NLP are still

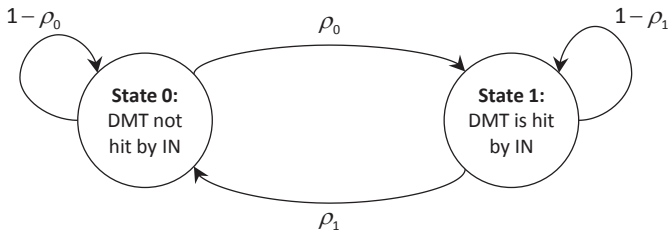

Figure 4.13: Two-state Markov IN model.

valid, provided that $\mathrm{SNR}_i^{(k)}$ is replaced by the instantaneous SNR given by

$$\mathrm{SNR}_{\mathrm{act},i}^{(k)} = \frac{|L_{i,i}^{(k)}|^2 E_i^{(k)}}{N_{0,\mathrm{act},i}^{(k)}} \tag{4.37}$$

(the DMT symbol index l is suppressed for notational convenience). The instantaneous SNR can be expressed as

$$\mathrm{SNR}_{\mathrm{act},i}^{(k)} = \begin{cases} \mathrm{SNR}_i^{(k)} & \text{if DMT is not hit by IN} \\ \dfrac{\mathrm{SNR}_i^{(k)}}{1+\kappa_i^{(k)}} & \text{if DMT is hit by IN} \end{cases} \tag{4.38}$$

where $\mathrm{SNR}_i^{(k)}$ is given by (4.24) and denotes the SNR in the absence of IN, and

$$\kappa_i^{(k)} = \frac{N_{0,\mathrm{imp},i}}{N_{0,i}^{(k)}} \tag{4.39}$$

denotes the strength of the IN *relative* to the stationary noise for the considered tone and user.

Whether or not a DMT symbol is hit by IN is governed by the 2-state Markov model shown in Figure 4.13, which captures the burstiness of the IN process [46,47]. The states 1 and 0 indicate that a DMT symbol is hit ($I(l) = 1$) and not hit ($I(l) = 0$) by IN. Denoting the state of the lth DMT symbol by s_l, we define the following state transition probabilities

$$\begin{aligned} \Pr[s_{l+1} = 0 | s_l = 0] &= 1 - \rho_0 \\ \Pr[s_{l+1} = 1 | s_l = 0] &= \rho_0 \\ \Pr[s_{l+1} = 0 | s_l = 1] &= \rho_1 \\ \Pr[s_{l+1} = 1 | s_l = 1] &= 1 - \rho_1 \end{aligned}$$

where ρ_0 and ρ_1 denote the probabilities of leaving the states 0 and 1, respectively. Defining the state probabilities corresponding to the lth DMT symbol as

$p_0(l) = \Pr[s_l = 0]$ and $p_1(l) = \Pr[s_l = 1]$, we introduce the state probability vector $\mathbf{p}(l)$ with $(\mathbf{p}(l))_0 = p_0(l)$ and $(\mathbf{p}(l))_1 = p_1(l)$. The state probability vectors corresponding to the lth and $(l+1)^{\text{th}}$ DMT symbol are related by $\mathbf{p}(l+1) = \mathbf{\Pi}\mathbf{p}(l)$, where

$$\mathbf{\Pi} = \begin{pmatrix} 1-\rho_0 & \rho_1 \\ \rho_0 & 1-\rho_1 \end{pmatrix}$$

is referred to as the stochastic matrix of the Markov model. The steady-state probability vector $\mathbf{p}$ satisfies $\mathbf{p} = \mathbf{\Pi}\mathbf{p}$, which yields $p_0 = \rho_1/(\rho_0+\rho_1)$ and $p_1 = \rho_0/(\rho_0+\rho_1)$. The state probability vectors corresponding to the lth and $(l+i)^{\text{th}}$ DMT symbol are related by $\mathbf{p}(l+i) = \mathbf{\Pi}^i\mathbf{p}(l)$. Using the eigenvalue-decomposition $\mathbf{\Pi} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, we obtain $\mathbf{\Pi}^i = \mathbf{V}\mathbf{\Lambda}^i\mathbf{V}^{-1}$. For the case at hand, the two eigenvalues of $\mathbf{\Pi}$ are 1 and $\lambda = 1-\rho_0-\rho_1$; note that $|\lambda| < 1$ (when excluding the pathological cases corresponding to $\rho_0 = \rho_1 = 0$ and $\rho_0 = \rho_1 = 1$, which give rise to $\lambda = 1$ and $\lambda = -1$, respectively). This yields

$$\mathbf{\Pi}^i = \frac{1}{\rho_0+\rho_1}\begin{pmatrix} \rho_1+\rho_0\lambda^i & \rho_1(1-\lambda^i) \\ \rho_0(1-\lambda^i) & \rho_0+\rho_1\lambda^i \end{pmatrix}$$

Hence, when the lth DMT symbol is hit by IN, the probability $\Pr[s_{l+i} = 1|s_l = 1]$, that the $(l+i)^{\text{th}}$ DMT symbol is also hit by IN, equals $\frac{\rho_0+\rho_1\lambda^i}{\rho_0+\rho_1}$, which for large i converges to the steady-state probability $p_1 = \frac{\rho_0}{\rho_0+\rho_1}$. Similarly, when the lth DMT symbol is not hit by IN, the probability $\Pr[s_{l+i} = 0|s_l = 0]$, that the $(l+i)^{\text{th}}$ DMT symbol is also not hit by IN, equals $\frac{\rho_1+\rho_0\lambda^i}{\rho_0+\rho_1}$, which for large i converges to the steady-state probability $p_0 = \frac{\rho_1}{\rho_0+\rho_1}$.

The IN is characterized by an alternation of on-intervals (during which the IN is present) and off-intervals (during which the IN is absent). The probability that an on-interval has a duration of n_{on} DMT symbol intervals is given by $(1-\rho_1)^{n_{\text{on}}-1}\rho_1$ ($n_{\text{on}} = 1, 2, \ldots$), which indicates that n_{on} has a geometrical distribution; the corresponding average duration of an on-interval equals T_{DMT}/ρ_1. Similarly, the probability that an off-interval has a duration of n_{off} DMT symbol intervals is given by $(1-\rho_0)^{n_{\text{off}}-1}\rho_0$ ($n_{\text{off}} = 1, 2, \ldots$), so that the average duration of an off-interval equals T_{DMT}/ρ_0. Hence, according to this model, the average duration of a noise impulse is T_{DMT}/ρ_1, and the average inter-arrival time of the noise impulses is $T_{\text{DMT}}(\rho_0^{-1}+\rho_1^{-1})$. We select the probabilities ρ_0 and ρ_1 so that the resulting average IN duration and inter-arrival time match with experimental IN data. Measurements reported in [39,40] show an average noise impulse duration and inter-arrival time of 35 μs and 1.3 ms, corresponding to 1.68 T_{DMT} and 60.73 T_{DMT}, respectively (for G.fast, $1/T_{\text{DMT}} = 48000\,s^{-1}$, so $T_{\text{DMT}} = 20.83\ \mu s$); this yields the transition probabilities $\rho_0 = 0.016$, $\rho_1 = 0.595$, the eigenvalue $\lambda = 0.389$, and the steady-state probabilities $p_0 = 0.974$, $p_1 = 0.026$. Figures 4.14 and 4.15 show as a function of l the probabilities $\Pr[s_{l+i} = 1|s_l = 1]$ and $\Pr[s_{l+i} = 0|s_l = 0]$, respectively,

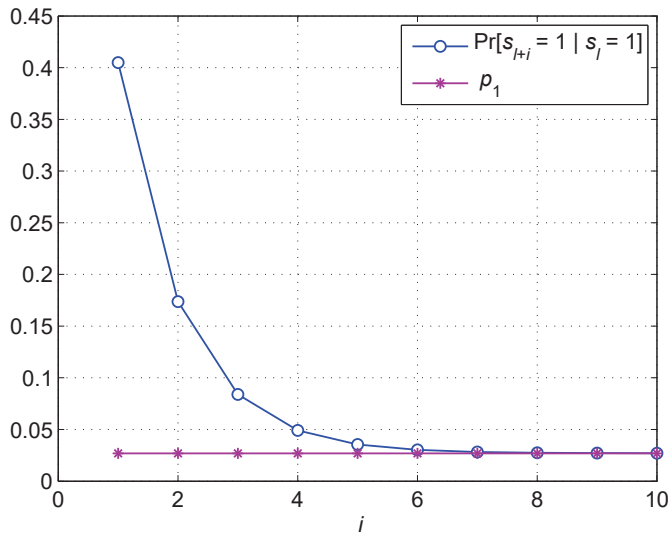

Figure 4.14: The probability $\Pr[s_{l+i} = 1|s_l = 1]$ versus i along with its steady-state value p_1.

along with their steady-state values p_1 and p_0; we observe that the steady-state probabilities are reached after about 8 DMT symbol intervals.

At the transmitter, the constellation sizes on the individual tones are determined from the per-tone SNRs. These SNRs are measured at the receiver, and fed back to the transmitter. The SNR measurement is affected by the presence of IN. Taking into account that the SNR measurement is obtained as the ratio of the average signal power over the average noise power, each averaged over many DMT symbol intervals, we obtain

$$\mathrm{SNR}_{\mathrm{avg},i}^{(k)} = \frac{\mathrm{SNR}_i^{(k)}}{1 + \kappa_i^{(k)} p_1} \tag{4.40}$$

where $\mathrm{SNR}_{\mathrm{avg},i}^{(k)}$ is the average SNR related to the kth tone of the ith user, which is measured by the receiver. In (4.40), $\mathrm{SNR}_i^{(k)}$ is the SNR on the same tone for a system without IN, and $\kappa_i^{(k)} = N_{0,\mathrm{imp},i}/N_{0,i}^{(k)}$ is given by (4.39). Indicating by $\mathrm{SNR}_{\mathrm{act},i}^{(k)}$ the instantaneous SNR related to the kth tone of the ith user for a

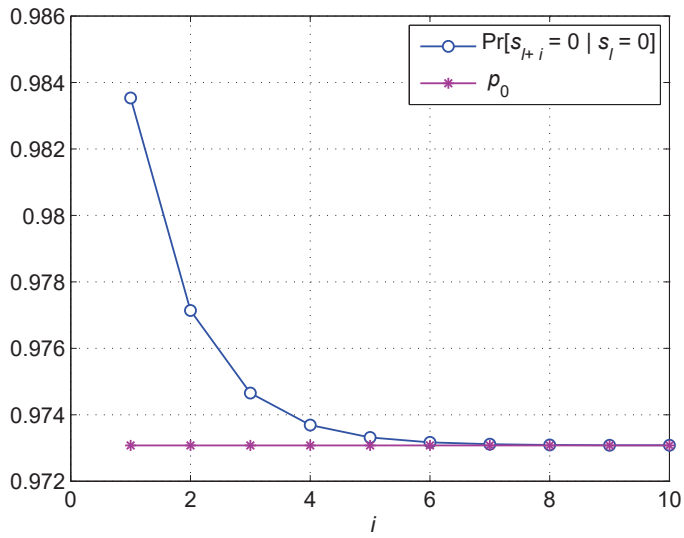

Figure 4.15: The probability $\Pr[s_{l+i} = 0|s_l = 0]$ versus i along with its steady-state value p_0.

particular DMT symbol, $\mathrm{SNR}^{(k)}_{\mathrm{act},i}$ can be expressed in terms of $\mathrm{SNR}^{(k)}_{\mathrm{avg},i}$:

$$\mathrm{SNR}^{(k)}_{\mathrm{act},i} = \begin{cases} (1+\kappa_i^{(k)} p_1)\mathrm{SNR}^{(k)}_{\mathrm{avg},i} & \text{if DMT is not hit by IN} \\ \dfrac{1+\kappa_i^{(k)} p_1}{1+\kappa_i^{(k)}}\mathrm{SNR}^{(k)}_{\mathrm{avg},i} & \text{if DMT is hit by IN} \end{cases} \tag{4.41}$$

Hence, the measured SNR, denoted by $\mathrm{SNR}^{(k)}_{\mathrm{avg},i}$, overestimates (underestimates) the instantaneous SNR, denoted by $\mathrm{SNR}^{(k)}_{\mathrm{act},i}$, when the DMT is hit (is not hit) by IN.

5 Error Performance with Linear and Nonlinear Precoding

In this chapter, we investigate uncoded transmission and various types of coded transmission in the presence of linear precoding (LP) or nonlinear precoding (NLP), as described in Sections 4.3 and 4.4. We use a single-carrier and single-user model. The obtained results will be used in Chapters 7 and 8, where we investigate the performance and goodput of the G.fast system, which corresponds to a multi-carrier (different SNR per tone) and multi-user scenario.

In the absence of IN, the resulting observation model for a given user (for notational convenience, we remove the user index) in the case of LP is

$$z^{(k)} = \sqrt{\mathrm{SNR}^{(k)}} a^{(k)} + w^{(k)} \tag{5.1}$$

where k is the tone index, $\mathrm{SNR}^{(k)}$ is the SNR for the considered user and tone at the input of the receiver; we recall the normalizations $\mathbb{E}[|a^{(k)}|^2] = \mathbb{E}[|w^{(k)}|^2] = 1$ for all k. In the case of NLP, the observation model becomes

$$z^{(k)} = \sqrt{\mathrm{SNR}^{(k)}} \left(a^{(k)} + l^{(k)} A^{(k)} \right) + w^{(k)} \tag{5.2}$$

where $l^{(k)} \in \mathbb{Z} + j\mathbb{Z}$. $\mathbb{Z} + j\mathbb{Z}$ denotes the set of complex numbers with integer real and imaginary parts, $A^{(k)}$ is the divisor related to the modulo operation

at the transmitter, and the remaining quantities are the same as in (5.1). In the presence of IN, the above observation models are still valid, provided that $\mathrm{SNR}^{(k)}$ is replaced by $\mathrm{SNR}_{\mathrm{act}}^{(k)}$, the instantaneous SNR given by (4.38).

For both cases of LP and NLP, we outline for uncoded transmission and for each type of coded transmission the algorithm for detecting from $\{z^{(k)}, k = 1, ..., N_{\mathrm{t}}\}$ the information bits associated with the data symbols $\{a^{(k)}, k = 1, ..., N_{\mathrm{t}}\}$, and investigate the resulting bit error rate (BER) using analytical approximations and computer simulations. Whereas the observation model (5.1) for LP gives rise to "conventional" detection algorithms, we point out that for NLP some modifications to these detection algorithms are needed in order to cope with the occurrence of the unknown $\{l^{(k)}, k = 1, ..., N_{\mathrm{t}}\}$ in the observation model (5.2). As far as the BER is concerned, we show that for given SNR the BER for NLP is (slightly) worse than for LP. However, in order to achieve a given SNR, LP and NLP require a different level of transmit power; typically, in the case of strong crosstalk, NLP requires less transmit power, compared to LP. A performance comparison in terms of given transmit power will be conducted in Chapters 7 and 8, where the findings from the present chapter will be used as an intermediate result.

5.1 Uncoded Transmission

In the case of uncoded transmission, the kth tone conveys a symbol $a^{(k)}$ from a normalized $M^{(k)}$-QAM constellation $\mathcal{A}^{(k)}$, which represents $\log_2(M^{(k)})$ information bits. For now we restrict our attention to the case where IN is absent; the effect of IN will be considered in Section 5.4.

An approximate analytical expression for the error performance of uncoded transmission is derived in Sections 5.1.1 and 5.1.2, for LP and NLP respectively. Section 5.1.4 presents some numerical results.

5.1.1 Linear Precoding

5.1.1.1 Detection Algorithm

First we consider the case of linear precoding, as explained in Section 4.3. Maximum-likelihood (ML) detection of the symbol $a^{(k)}$ based on the observation $z^{(k)}$ from (5.1) yields the symbol decision $\hat{a}^{(k)}$, with

$$\hat{a}^{(k)} = \arg \min_{\tilde{a} \in \mathcal{A}^{(k)}} \left| z^{(k)} - \sqrt{\mathrm{SNR}^{(k)}} \tilde{a} \right|^2 \tag{5.3}$$

where the minimization is over all points of the normalized constellation $\mathcal{A}^{(k)}$.

5.1.1.2 Error Performance

Denoting the minimum Euclidean distance between points of the normalized $M^{(k)}$-QAM constellation as $d^{(k)}$, the corresponding BER related to the detection of the symbol $a^{(k)}$ is well approximated as [20]

$$\mathrm{BER}^{(k)} \approx \frac{1}{M^{(k)}} \sum_{\substack{a \in \mathcal{A}^{(k)}, \hat{a} \in \mathcal{A}^{(k)} \\ |\hat{a}-a| = d^{(k)}}} \frac{N_{\mathrm{bit}}(a,\hat{a})}{\log_2(M^{(k)})} Q\left(\sqrt{\frac{\left(d^{(k)}\right)^2}{2}\mathrm{SNR}^{(k)}}\right) \tag{5.4}$$

In the above expression, the summation is over constellation points a and $\hat{a}$ which are at distance $d^{(k)}$ from one another, and $N_{\mathrm{bit}}(a,\hat{a})$ denotes the number of bits in which the binary labels of the constellation points a and $\hat{a}$ are different. Note that $d^{(k)}$ depends on the type and size of the considered constellation (see Section 4.2). The BER averaged over all N_t tones is obtained as

$$\mathrm{BER} = \frac{\sum_{k=1}^{N_t} \log_2(M^{(k)})\mathrm{BER}^{(k)}}{\sum_{k=1}^{N_t} \log_2(M^{(k)})} \tag{5.5}$$

When all N_t tones have the same constellation and the same SNR, i.e., $\mathrm{SNR}^{(k)} = \mathrm{SNR}$ and $M^{(k)} = M$ for all k, we obtain

$$\mathrm{BER} = \mathrm{BER}^{(k)} \approx \frac{1}{M} \sum_{\substack{a \in C_M, \hat{a} \in C_M \\ |\hat{a}-a| = d}} \frac{N_{\mathrm{bit}}(a,\hat{a})}{\log_2(M)} Q\left(\sqrt{\frac{d^2}{2}\mathrm{SNR}}\right) \tag{5.6}$$

with d denoting the minimum Euclidean distance of the normalized M-QAM constellation C_M.

For 2-QAM, 8-QAM-VDSL and square-QAM constellations, we use Gray mapping, which yields $N_{\mathrm{bit}}(a,\hat{a}) = 1$ for all constellation points a and $\hat{a}$ satisfying $|\hat{a}-a| = d^{(k)}$, hence minimizing (5.4). For 8-QAM-G.fast and cross-QAM constellations, a Gray mapping does not exist. For 8-QAM-G.fast we use the mappings shown in Figures 5.1 and 5.2 for LP and NLP, respectively. For cross-QAM with $2n+1$ bits ($n \geq 2$), we first apply Gray mapping to a rectangular $2^n \times 2^{n+1}$ QAM constellation, and subsequently turn the $2^n \times 2^{n+1}$ rectangular QAM constellation into a 2^{2n+1} cross-QAM constellation, by rotating over 90 degrees the leftmost and rightmost 2^{n-2} columns (each column containing 2^n constellation points), changing the order of the points as explained in [48, 49] and moving them above and below the 2^n center columns, respectively; this procedure is illustrated for 32-QAM in Figure 5.3. The resulting mapping minimizes the average number of bit differences between labels associated with constellation points at minimum distance.

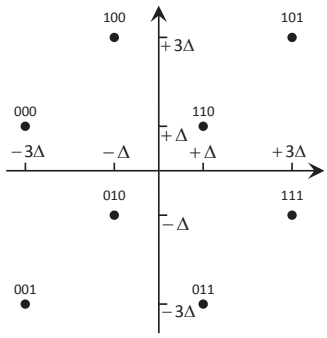

Figure 5.1: Mapping for the 8-QAM-G.fast constellation for uncoded transmission with LP.

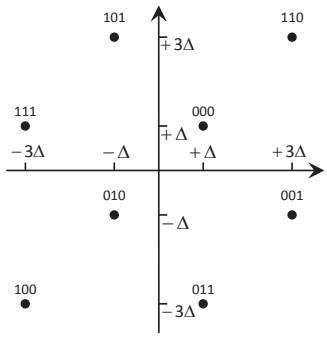

Figure 5.2: Mapping for the diamond 8-QAM constellation for uncoded transmission with NLP.

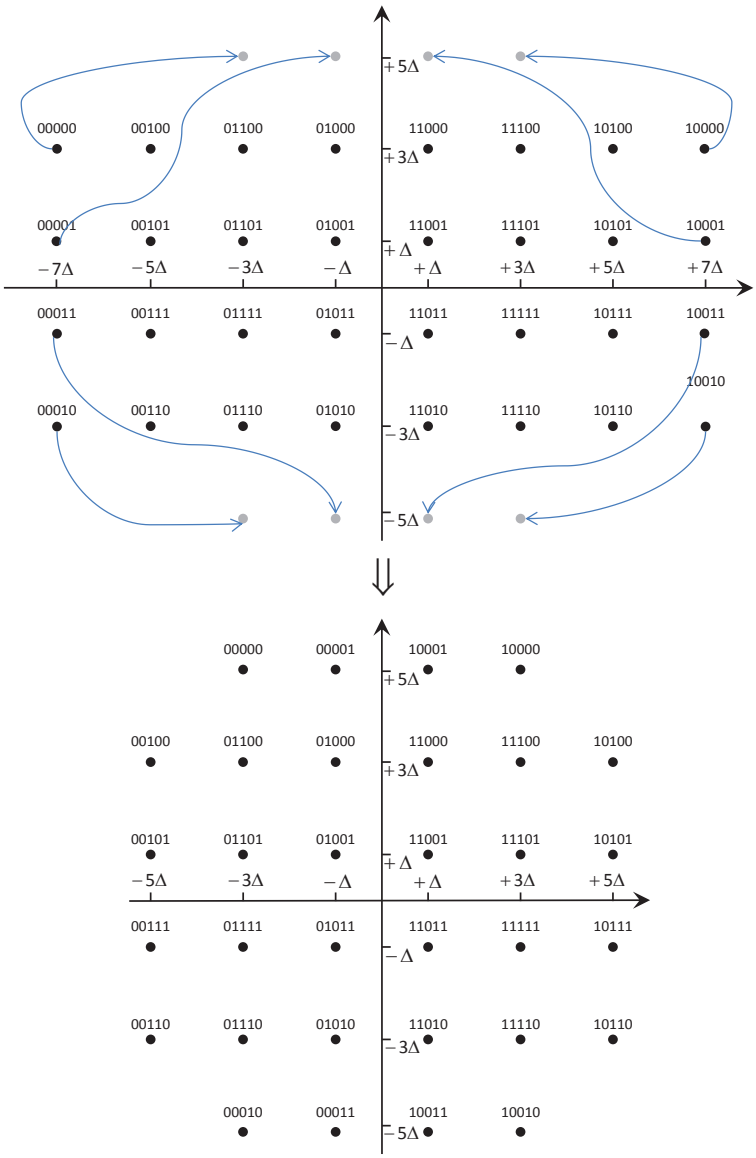

Figure 5.3: Transformation of a rectangular 32-QAM constellation into a cross 32-QAM constellation.

5.1.2 Nonlinear Precoding

5.1.2.1 Detection Algorithm

In the case of nonlinear precoding, as explained in Section 4.4, the scaled and rotated observation $z^{(k)}$ is given by (5.2) where $a^{(k)} + l^{(k)} A^{(k)}$ can be interpreted as a symbol belonging to the periodic extension of the normalized $M^{(k)}$-QAM constellation $\mathcal{A}^{(k)}$; this extended constellation is denoted $\mathcal{A}^{(k)}_{\text{ext}}$. The symbol decision $\hat{a}^{(k)}$ is given by

$$\hat{a}^{(k)} = \arg \min_{\tilde{a} \in \mathcal{A}^{(k)}} \left(\min_{\tilde{l} \in \mathbb{Z} + j\mathbb{Z}} \left| z^{(k)} - \sqrt{\text{SNR}^{(k)}} \left(\tilde{a} + \tilde{l} A^{(k)} \right) \right|^2 \right) \tag{5.7}$$

$$= \arg \min_{\tilde{a} \in \mathcal{A}^{(k)}} \left(\min_{\tilde{l} \in \mathbb{Z} + j\mathbb{Z}} \left| \left[z^{(k)} \right]_{\sqrt{\text{SNR}^{(k)}} A^{(k)}} - \sqrt{\text{SNR}^{(k)}} \left(\tilde{a} + \tilde{l} A^{(k)} \right) \right|^2 \right) \tag{5.8}$$

It follows from (5.8) that $\left[z^{(k)} \right]_{\sqrt{\text{SNR}^{(k)}} A^{(k)}}$, which results from a modulo operation on $z^{(k)}$, is a sufficient statistic for performing the ML decision.

5.1.2.2 Error Performance

The resulting BER related to the detection of the symbol $a^{(k)}$ is given by an expression similar to (5.4), i.e.,

$$\text{BER}^{(k)} \approx \frac{1}{M^{(k)}} \sum_{\substack{a \in \mathcal{A}^{(k)}, \hat{a} \in \mathcal{A}^{(k)}_{\text{ext},i} \\ |\hat{a} - a| = d^{(k)}}} \frac{N_{\text{bit}}(a, \hat{a})}{\log_2(M^{(k)})} Q \left(\sqrt{\frac{\left(d^{(k)}\right)^2}{2} \text{SNR}^{(k)}} \right) \tag{5.9}$$

The difference between (5.4) and (5.9) is in the summation over $\hat{a}$: for LP and NLP, $\hat{a}$ belongs to the original constellation $\mathcal{A}^{(k)}$ and to the extended constellation $\mathcal{A}^{(k)}_{\text{ext}}$, respectively; hence, for given constellation sizes and SNRs per tone, NLP yields the larger BER because $\mathcal{A}^{(k)} \subset \mathcal{A}^{(k)}_{\text{ext}}$. The BER averaged over all N_t tones is given by (5.5), with $\text{BER}^{(k)}$ given by (5.9).

5.1.3 Rule of Thumb

The approximate BER expressions (5.4) for LP and (5.9) for NLP indicate that the selection of the constellation affects the error performance through the minimum distance $d^{(k)}$ of the normalized constellation and through the factor in front of the function $Q(.)$. For large $\text{SNR}^{(k)}$, the dependence of $d^{(k)}$ on the constellation is the more important factor. Hence, when replacing the normalized constellation $\mathcal{A}^{(k)}$ with minimum distance $d^{(k)}$ by the normalized

constellation $\mathcal{A}'^{(k)}$ with minimum distance $d'^{(k)}$, one should change the SNR from $\mathrm{SNR}^{(k)}$ to $\mathrm{SNR}'^{(k)} = (d^{(k)}/d'^{(k)})^2\mathrm{SNR}^{(k)}$ in order to maintain essentially the same BER at high SNR.

For normalized square-QAM constellations with *large* M, (4.13) simplifies to $d^2 \approx 6/M$. For normalized cross-QAM with large M, (4.15) also yields $d^2 \approx 6/M$, taking into account that $31/32 \approx 1$. Hence, when increasing the constellation size from $M = 2^b$ to $M' = 2^{b'}$ with $b' > b$, the error performance is maintained when increasing the SNR by a factor M'/M, which corresponds to an increase (expressed in dB) of $10\log(M'/M) = 10\log(2)\cdot(b'-b) \approx 3(b'-b)$. Hence, when increasing the number of bits per QAM symbol by 1 (i.e., when doubling the number of constellation points), the resulting SNR penalty for maintaining a certain error performance equals about 3 dB.

5.1.4 Numerical Results

In the remainder of this dissertation, we will use the G.fast constellation in the case of 8-QAM, unless explicitly mentioned otherwise.

Considering the observation models (5.1) and (5.2) for LP and NLP, Figure 5.4 and Figure 5.5 show the corresponding BER for uncoded transmission for QAM constellations with $b = 1, ..., 12$. We observe that the BER approximations (5.4) and (5.9), accurately match the simulations results. For large constellation sizes, the BER curve for 2^{b+1}-QAM is obtained by shifting the BER curve for 2^b-QAM by about 3 dB to the right; this is according to the rule of thumb from Section 5.1.3, stating that adding 1 bit to the constellation gives rise to a SNR penalty of about 3 dB. Actually, taking the factor 31/32 occuring in the expression (4.15) for the minimum distance for cross-QAM into account, we have for large constellation sizes $d^2_{2^{2n}}/d^2_{2^{2n+1}} = 31/16$ (2.87 dB)) and $d^2_{2^{2n+1}}/d^2_{2^{2n+2}} = 64/31$ (3.14 dB); hence, for small BER, the BER curve for 2^{2n+1}-QAM is not exactly halfway the curves for 2^{2n}-QAM and 2^{2n+2}-QAM, but slightly closer to the curve for 2^{2n}-QAM.

Figure 5.6 shows the BER for NLP with uncoded transmission using the following 8-QAM constellations: 8-QAM-G.fast, 8-QAM-VDSL with $A = 8\Delta$ and 8-QAM-VDSL with $A = 6\Delta$. 8-QAM-G.fast gives the best performance at high SNR, as the minimum Euclidean distance between constellation points d is larger for G.fast ($d^2 = 4/5$) as compared to VDSL ($d^2 = 2/3$, for both $A = 8\Delta$ and $A = 6\Delta$). The better performance of 8-QAM-VDSL with $A = 8\Delta$ as compared to $A = 6\Delta$ is caused by the larger number of neighbours at minimal distance when $A = 6\Delta$.

Table 5.1 lists the SNR values that are required to achieve BER $= 10^{-7}$ for the various constellation sizes, for both linear and nonlinear precoding. We observe that nonlinear precoding needs a slightly larger SNR, because the extended constellation $\mathcal{A}^{(k)}_{\mathrm{ext},i}$ has more neighbours at minimum Euclidean distance from the transmitted symbol, when the latter is an outer point of the original

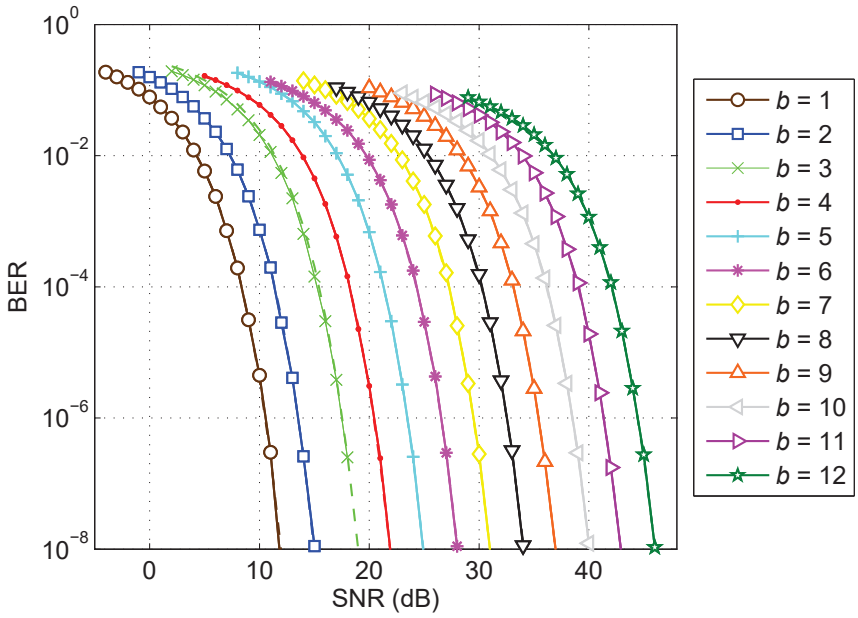

Figure 5.4: BER for LP and uncoded transmission for QAM constellations with $b = 1, \ldots, 12$; simulations (solid lines with markers) versus approximation (dashed lines).

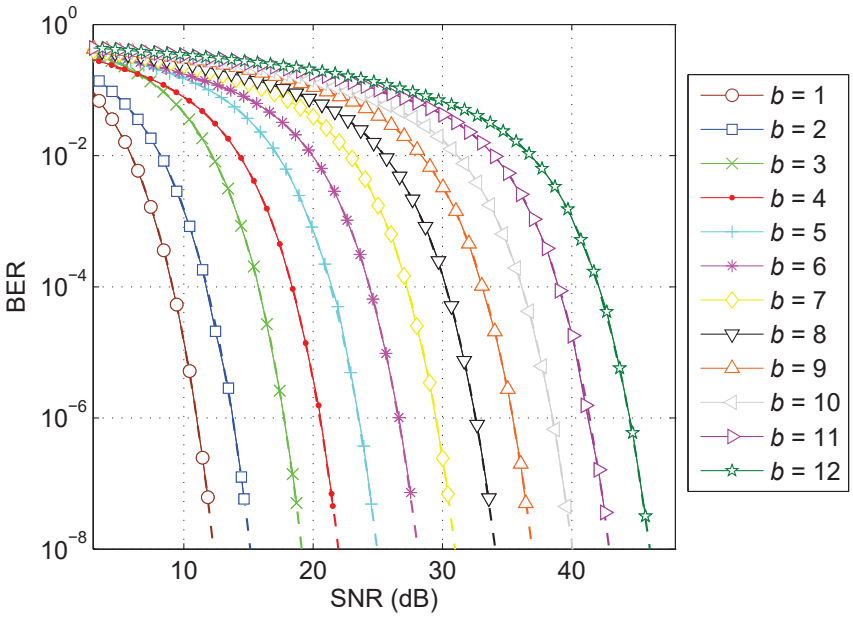

Figure 5.5: BER for NLP and uncoded transmission for QAM constellations with $b = 1, \ldots, 12$; simulations (solid lines with markers) versus approximation (dashed lines).

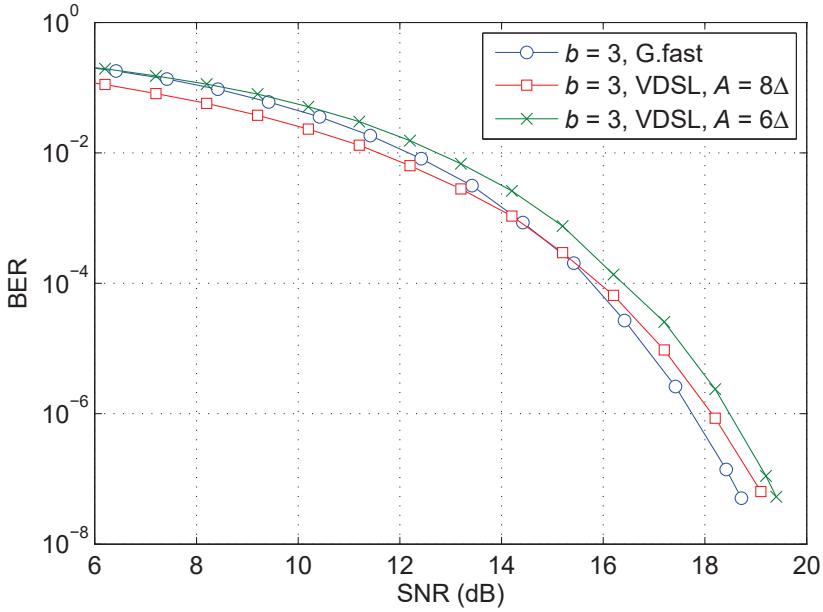

Figure 5.6: BER for NLP, uncoded transmission and the proposed 8-QAM constellations; simulations results.

Table 5.1: Required SNR (in dB) to achieve the target BER of 10^{-7} for uncoded transmission with LP and NLP.

b	1	2	3	4	5	6
LP	11.3	14.3	18.3	21.2	24.2	27.4
NLP	11.7	14.5	18.5	21.3	24.3	27.4

b	7	8	9	10	11	12
LP	30.3	33.4	36.2	39.3	42.2	45.3
NLP	30.3	33.4	36.2	39.4	42.2	45.3

constellation $\mathcal{A}^{(k)}$; this effect decreases with increasing constellation size, because of the decreasing relative importance of the outer constellation points of $\mathcal{A}^{(k)}$.

5.2 Trellis-Coded Modulation

In Section 5.2.1, we describe the trellis code used in G.fast and in the VDSL standard. The detection algorithm and an approximate analytical expression for the error performance of TCM is derived in Sections 5.2.2 and 5.2.3, for linear and nonlinear precoding, respectively. Numerical results are presented in Section 5.2.5. Here we restrict our attention to the case where IN is absent; the effect of IN will be considered in Section 5.4.

5.2.1 Trellis Code Description

The trellis codes introduced by Ungerboeck typically make use of 2^{m+1}-point complex-valued constellations, such as PSK or QAM constellations, to convey m information bits per constellation; these constellations are referred to as two-dimensional (2-D), because they involve two "real" dimensions. The concept of TCM has been extended to convey m information bits per 2^{m+1}-point $2N_{\text{dim}}$-dimensional constellations (with $N_{\text{dim}} > 1$), which are obtained as the cartesian product of N_{dim} 2-D constellations [50]. For a given number of information bits per 2-D symbol, multi-dimensional TCM can be shown to provide a better error performance, because of the smaller number of parity bits (i.e. $(1/N_{\text{dim}})$ parity bits) per 2-D symbol (see Section 5.2.4).

The trellis code, depicted in Figure 5.7, is defined in ITU-T Recommendation G.993.2 [5] and is similar to Wei's 16-state 4-dimensional trellis code [50]; only the symbol mapping is different. At the ℓth trellis section, the trellis encoder transforms an input vector $\mathbf{u}^{(\ell)}$ of $z^{(\ell)}$ information bits into $z^{(\ell)} + 1$ coded bits, i.e., one redundant bit is added. These $z^{(\ell)} + 1$ coded bits are split into a set

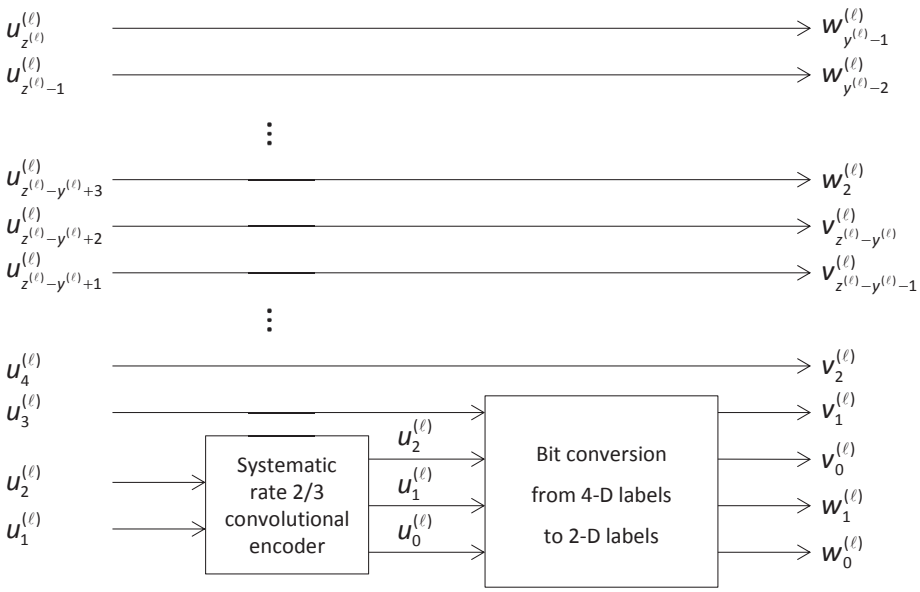

Figure 5.7: Structure of the trellis encoder in G.fast.

of $z^{(\ell)} + 1 - y^{(\ell)}$ coded bits and a set of $y^{(\ell)}$ coded bits, which are grouped in the output vectors $\mathbf{v}^{(\ell)}$ and $\mathbf{w}^{(\ell)}$, respectively. The vectors $\mathbf{v}^{(\ell)}$ and $\mathbf{w}^{(\ell)}$ are mapped to symbols from a $2^{z^{(\ell)}+1-y^{(\ell)}}$-QAM constellation and a $2^{y^{(\ell)}}$-QAM constellation, respectively, which are transmitted using two tones of the DMT system. As the output of the trellis encoder consists of two QAM symbols per trellis section, the trellis code is referred to as 4-dimensional (the real and imaginary parts of a QAM symbol are considered as two separate dimensions).

The redundant bit $u_0^{(\ell)}$ in Figure 5.7 is produced by a 16-state systematic recursive rate 2/3 convolutional encoder, the block diagram of which is shown in Figure 5.8. The trellis section for the 16-state convolutional code from Figure 5.8 is given in Figure 5.9. The next state $(S_3^{(\ell+1)}, S_2^{(\ell+1)}, S_1^{(\ell+1)}, S_0^{(\ell+1)})$ is determined solely by the current state $(S_3^{(\ell)}, S_2^{(\ell)}, S_1^{(\ell)}, S_0^{(\ell)})$ and the input $(u_2^{(\ell)}, u_1^{(\ell)})$ at time instant ℓ; the two information bits $(u_2^{(\ell)}, u_1^{(\ell)})$ which enter the convolutional encoder are referred to as the encoded information bits, while the remaining $z^{(\ell)} - 2$ information bits which bypass the convolutional encoder are termed the uncoded information bits. Each state has 4 outgoing edges corresponding to the 4 possible values of $(u_2^{(\ell)}, u_1^{(\ell)})$, that end in 4 different states; similarly, each state has 4 incoming edges that originate from 4 different states. The values to the left of the trellis section, in front of each state, list the decimal input values of $(u_2^{(\ell)}, u_1^{(\ell)})$ corresponding to the outgoing branches from top to bottom. Similarly, to the right of the trellis, behind each state, the decimal val-

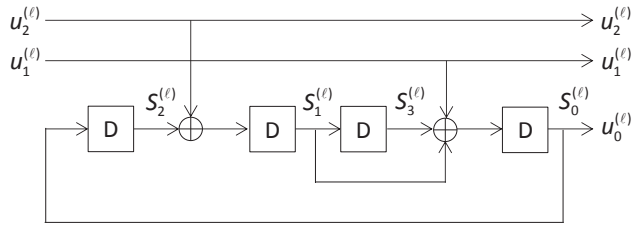

Figure 5.8: Rate-2/3 16-state systematic convolutional encoder for TCM in G.fast.

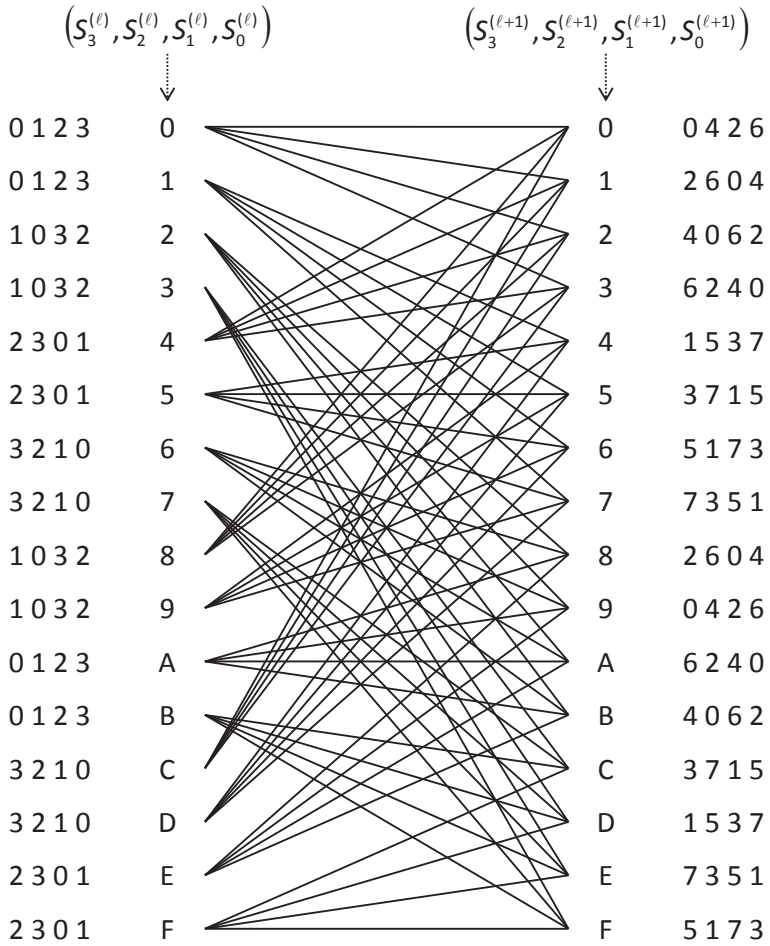

Figure 5.9: Trellis diagram for the 16-state convolutional code.

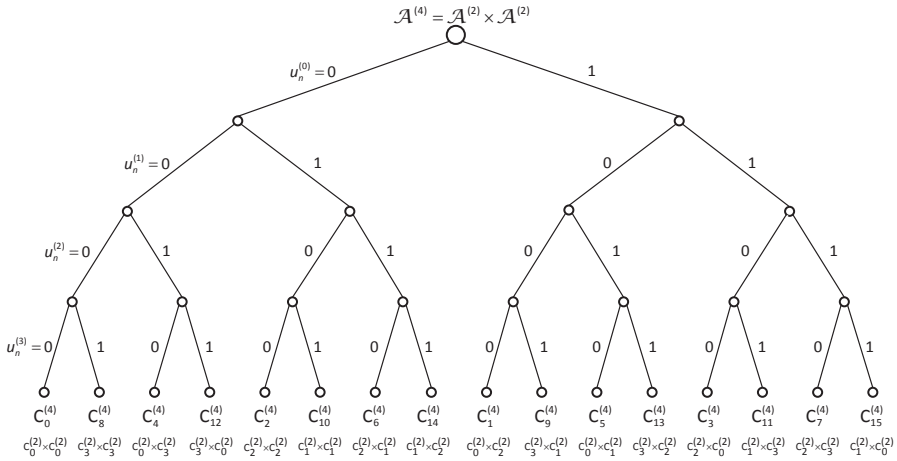

Figure 5.10: Set partitioning of the 4-D constellation $\mathcal{A}^4 = \mathcal{A}^2 \times \mathcal{A}^2$ as in VDSL2.

ues of $(u_2^{(\ell)}, u_1^{(\ell)}, u_0^{(\ell)})$ are indicated that correspond to the incoming branches in the state from top to bottom.

Figure 5.10 illustrates the set partitioning of the 4-D constellation $\mathcal{A}^{(4)} = \mathcal{A}_1^{(2)} \times \mathcal{A}_2^{(2)}$, where $\mathcal{A}_1^{(2)}$ and $\mathcal{A}_2^{(2)}$ are 2-D constellations with 2^{b_1} and 2^{b_2} points, respectively, so that the 4-D constellation has $2^{b_1+b_2}$ points; b_1 and b_2 are short-hand notations for $z^{(\ell)} + 1 - y^{(\ell)}$ and $y^{(\ell)}$. This set partitioning results in 16 subsets $\{C_n^{(4)}, n = 0, ..., 15\}$ from the original 4-D constellation, each containing $2^{b_1+b_2-4}$ points. Each of the 4-D subsets $C_n^{(4)}$ is obtained as $C_n^{(4)} = C_{1,m_1}^{(2)} \times C_{2,m_2}^{(2)}$, where $C_{1,m_1}^{(2)}$ and $C_{2,m_2}^{(2)}$ are subsets resulting from two levels of set partioning of the 2-D constellations $\mathcal{A}_1^{(2)}$ and $\mathcal{A}_2^{(2)}$; the subsets $C_{1,m_1}^{(2)}$ and $C_{1,m_2}^{(2)}$contain 2^{b_1-2} and 2^{b_2-2} points. The indices n, m_1, and m_2 are the decimal values corresponding to $(u_3^{(\ell)}, u_2^{(\ell)}, u_1^{(\ell)}, u_0^{(\ell)})$, $(v_1^{(\ell)}, v_0^{(\ell)})$ and $(w_1^{(\ell)}, w_0^{(\ell)})$, respectively; the remaining bits $(v_{b_1-1}^{(\ell)}, ..., v_2^{(\ell)})$ and $(w_{b_2-1}^{(\ell)}, ..., w_2^{(\ell)})$ determine which points from the 2-D subsets $C_{1,m_1}^{(2)}$ and $C_{2,m_2}^{(2)}$ are actually transmitted. Figure 5.11 illustrates the set partitioning of the 2-D constellations for 64-QAM; the constellation points are labeled by the decimal value of the two bits (either $(v_1^{(\ell)}, v_0^{(\ell)})$ or $(w_1^{(\ell)}, w_0^{(\ell)})$) according to the subset they belong to. Representing by d the minimum Euclidean distance of the 2-D constellation, we note that the minimum distance within a same subset equals $2d$, and the minimum distance between subsets determined by the decimal values 0 and 3 (or 1 and 2) equals $\sqrt{2}d$.

The trellis encoder output bits $(v_1^{(\ell)}, v_0^{(\ell)})$ and $(w_1^{(\ell)}, w_0^{(\ell)})$ are obtained from

1	3	1	3	1	3	1	3
0	2	0	2	0	2	0	2
1	3	1	3	1	3	1	3
0	2	0	2	0	2	0	2
1	3	1	3	1	3	1	3
0	2	0	2	0	2	0	2
1	3	1	3	1	3	1	3
0	2	0	2	0	2	0	2

Figure 5.11: Mapping of 2-D subsets in 64-QAM.

the bits $(u_3^{(\ell)}, u_2^{(\ell)}, u_1^{(\ell)}, u_0^{(\ell)})$ according to the following modulo-2 additions:

$$\begin{aligned} v_1^{(\ell)} &= u_1^{(\ell)} \oplus u_3^{(\ell)} \\ v_0^{(\ell)} &= u_3^{(\ell)} \\ w_1^{(\ell)} &= u_0^{(\ell)} \oplus u_1^{(\ell)} \oplus u_2^{(\ell)} \oplus u_3^{(\ell)} \\ w_0^{(\ell)} &= u_2^{(\ell)} \oplus u_3^{(\ell)} \end{aligned}$$

The remaining $z^{(\ell)} - 3$ trellis encoder output bits are equal to the $z^{(\ell)} - 3$ information bits that are obtained by removing the information bits $(u_3^{(\ell)}, u_2^{(\ell)}, u_1^{(\ell)})$ from the information bit vector $\mathbf{u}^{(\ell)}$.

From the above description, it follows that the minimum number of information bits in a trellis section equals three (i.e., the bits $(u_3^{(\ell)}, u_2^{(\ell)}, u_1^{(\ell)})$), which corresponds to two 4-QAM symbols, determined by the bits $(v_1^{(\ell)}, v_0^{(\ell)})$ and $(w_1^{(\ell)}, w_0^{(\ell)})$, respectively. When some of the tones convey 2-QAM symbols, two such symbols are 'paired' to form a 4-QAM symbol. More specifically, denoting the two (normalized) 2-QAM symbols as $a_{2\text{QAM},l} = \alpha_l(1+j)/\sqrt{2}$ where $\alpha_l \in \{-1, 1\}$ and $l = 1, 2$, the corresponding (normalized) 4-QAM symbol obtained after pairing is given by $a_{4\text{QAM}} = (\alpha_1 + j\alpha_2)/\sqrt{2}$.

The QAM symbols from the TCM codeword are transmitted using the DMT communication system. Each subcarrier of the DMT system conveys a QAM symbol from the TCM codeword. The number of QAM symbols in the TCM codeword equals the number of tones in the DMT symbol, so that one TCM codeword corresponds to one DMT symbol.

For the square-QAM and cross-QAM constellations, we use the mappings as proposed in [42]. In Figure 5.2, we show the mapping for 8-QAM-G.fast, where the last two bits represent either $(v_1^{(\ell)}, v_0^{(\ell)})$ or $(w_1^{(\ell)}, w_0^{(\ell)})$, which determine the 2-D subset to which the constellation point belongs.

5.2.2 Linear Precoding

5.2.2.1 Detection Algorithm

We consider the observation model (5.1) in the case of LP, with the user index removed for notational convenience. Applying ML sequence detection, the detected symbol sequence $\hat{\mathbf{a}} = \left(\hat{a}^{(1)}, \hat{a}^{(2)}, ..., \hat{a}^{(N_\mathrm{t})}\right)$ is obtained as $\hat{\mathbf{a}} = \arg\min_{\tilde{\mathbf{a}} \in \mathcal{C}} D_\mathrm{LP}(\tilde{\mathbf{a}})$, where

$$D_\mathrm{LP}(\tilde{\mathbf{a}}) = \sum_{k=1}^{N_\mathrm{t}} \left| z^{(k)} - \sqrt{\mathrm{SNR}^{(k)}} \tilde{a}^{(k)} \right|^2 \tag{5.10}$$

$\tilde{\mathbf{a}} = \left(\tilde{a}^{(1)}, \tilde{a}^{(2)}, ..., \tilde{a}^{(N_\mathrm{t})}\right)$ and the minimization is over the set $\mathcal{C}$ of valid coded symbol sequences. This minimization is executed efficiently by applying the Viterbi algorithm to the trellis that describes the operation of the trellis encoder. As the state transition at the ℓth trellis section is determined only by the current state $(S_3^{(\ell)}, S_2^{(\ell)}, S_1^{(\ell)}, S_0^{(\ell)})$ and the information bits $(u_2^{(\ell)}, u_1^{(\ell)})$, it follows that for a given current state the different information bit vectors $\mathbf{u}^{(\ell)}$ which have the same $(u_2^{(\ell)}, u_1^{(\ell)})$ give rise to the same state transition. Hence, each state transition in the trellis of the rate 2/3 convolutional code (see Figure 5.9) turns into $2^{z^{(\ell)}-2}$ parallel transitions (corresponding to the uncoded information bits) in the trellis of the trellis code. Each of these parallel transitions corresponds to a different 4-D symbol; these 4-D symbols have the bits $(u_2^{(\ell)}, u_1^{(\ell)}, u_0^{(\ell)})$ in common. The branch metric to be used in the Viterbi algorithm for a given state transition at the ℓth trellis section is given by $\min_{\tilde{a}^{(2\ell)}, \tilde{a}^{(2\ell+1)}} \left(D_\mathrm{LP}^{(2\ell)}(\tilde{a}^{(2\ell)}) + D_\mathrm{LP}^{(2\ell+1)}(\tilde{a}^{(2\ell+1)})\right)$, where

$$D_\mathrm{LP}^{(k)}(\tilde{a}^{(k)}) = \left| z^{(k)} - \sqrt{\mathrm{SNR}^{(k)}} \tilde{a}^{(k)} \right|^2 \tag{5.11}$$

and the minimization is over the 4-D symbols $(\tilde{a}^{(2\ell)}, \tilde{a}^{(2\ell+1)})$ associated with the parallel transitions corresponding to the considered state transition. In the case where a 4-QAM symbol $a^{(k)}$ is the result of the pairing of two 2-QAM symbols $a^{(k_1)}$ and $a^{(k_2)}$ with associated observations $z^{(k_1)}$ and $z^{(k_2)}$, we have

$$D_\mathrm{LP}^{(k)}(\tilde{a}^{(k)}) = \left| z^{(k_1)} - \sqrt{\mathrm{SNR}^{(k_1)}} \tilde{a}^{(k_1)} \right|^2 + \left| z^{(k_2)} - \sqrt{\mathrm{SNR}^{(k_2)}} \tilde{a}^{(k_2)} \right|^2$$

where $\tilde{a}^{(k_1)} = (1+j)\Re(\tilde{a}^{(k)})$ and $\tilde{a}^{(k_2)} = (1+j)\Im(\tilde{a}^{(k)})$.

5.2.2.2 Error Performance

The pairwise error probability $\mathrm{PEP}(\hat{\mathbf{a}}|\mathbf{a}) = \Pr[D_\mathrm{LP}(\hat{\mathbf{a}}) < D_\mathrm{LP}(\mathbf{a})|\mathbf{a}]$ is an upper bound on the probability that the ML decision equals $\hat{\mathbf{a}}$, when the transmitted

symbol vector equals $\mathbf{a}$. It is easily verified that

$$\mathrm{PEP}(\hat{\mathbf{a}}|\mathbf{a}) = Q\left(\sqrt{\frac{1}{2}\sum_{k=1}^{N_\mathrm{t}} |\Delta a^{(k)}|^2 \mathrm{SNR}^{(k)}}\right) \tag{5.12}$$

where $\Delta a^{(k)} = \hat{a}^{(k)} - a^{(k)}$. The resulting BER is upper bounded as

$$\mathrm{BER} \leq \frac{\sum_{\mathbf{a},\hat{\mathbf{a}}} N_\mathrm{info}(\mathbf{a},\hat{\mathbf{a}})\mathrm{PEP}(\hat{\mathbf{a}}|\mathbf{a})\,\Pr[\mathbf{a}]}{\sum_{k=1}^{N_\mathrm{t}} \left(\log_2(M^{(k)}) - 1/2\right)} \tag{5.13}$$

where the summation in the numerator is over all allowed coded symbol sequences $\mathbf{a} \in \mathcal{C}$ and $\hat{\mathbf{a}} \in \mathcal{C}$, $N_\mathrm{info}(\mathbf{a},\hat{\mathbf{a}})$ denotes the number of information bits that are different between the coded sequences $\mathbf{a}$ and $\hat{\mathbf{a}}$, $\Pr[\mathbf{a}] = 2^{N_t/2}\prod_{k=1}^{N_t}\left(M^{(k)}\right)^{-1}$ is the a priori probability that the transmitted coded sequence equals $\mathbf{a}$, and the denominator of (5.13) equals the number of information bits within the DMT symbol.

Now we restrict our attention to the case where the constellations and the SNRs are the same for all N_t tones, i.e., $M^{(k)} = M$ and $\mathrm{SNR}^{(k)} = \mathrm{SNR}$ for all k, so that (5.12) and (5.13) reduce to

$$\mathrm{PEP}(\hat{\mathbf{a}}|\mathbf{a}) = Q\left(\sqrt{\frac{\mathrm{SNR}}{2}|\Delta\mathbf{a}|^2}\right)$$

$$\mathrm{BER} \leq \frac{\sum_{\mathbf{a},\hat{\mathbf{a}}} N_\mathrm{info}(\mathbf{a},\hat{\mathbf{a}})\mathrm{PEP}(\hat{\mathbf{a}}|\mathbf{a})\,\Pr[\mathbf{a}]}{(\log_2(M) - 1/2)\,N_\mathrm{t}} \tag{5.14}$$

and $\Pr[\mathbf{a}] = \left(\sqrt{2}/M\right)^{N_t}$. For a given transmitted coded symbol sequence $\mathbf{a}$, the upper bound on BER is dominated by the coded symbol sequences $\hat{\mathbf{a}}$ with $\hat{\mathbf{a}} \neq \mathbf{a}$, for which the Euclidean distance $|\Delta\mathbf{a}|^2$ is the smallest. Let us consider the case where $\hat{\mathbf{a}}$ differs from $\mathbf{a}$ in the ℓth trellis section only: $(\hat{a}^{(2\ell)}, \hat{a}^{(2\ell+1)})$ represents a 4-D symbol on a branch connecting the same states as the branch that carries the 4-D symbol $(a^{(2\ell)}, a^{(2\ell+1)})$, i.e., these two branches correspond to parallel transitions. Hence, these two 4-D symbols have the bits $(u_2^{(\ell)}, u_1^{(\ell)}, u_0^{(\ell)})$ in common, so they differ only in the uncoded information bits. The following cases can be distinguished.

1. The two 4-D symbols have also the bit $u_3^{(\ell)}$ in common. The 2-D symbols $a^{(2\ell)}$ and $\hat{a}^{(2\ell)}$ belong to a same subset $C_{1,m_1}^{(2)}$, and $a^{(2\ell+1)}$ and $\hat{a}^{(2\ell+1)}$ belong to a same subset $C_{2,m_2}^{(2)}$; a point from either subset has (at most) 4 nearest neighbours at distance $2d$ (with d representing the minimum Euclidean distance of the M-QAM constellation). Hence, the corresponding minimum $|\Delta\mathbf{a}|^2$ equals $4d^2$, which is obtained when either $\hat{a}^{(2\ell)} = a^{(2\ell)}$, and $\hat{a}^{(2\ell+1)}$ is a nearest neighbour of $a^{(2\ell+1)}$ (at most 4 possibilities for

$\hat{a}^{(2\ell+1)}$), or $\hat{a}^{(2\ell+1)} = a^{(2\ell+1)}$, and $\hat{a}^{(2\ell)}$ is a nearest neighbor of $a^{(2\ell)}$ (at most 4 possibilities for $\tilde{a}^{(2\ell)}$). This corresponds to a total of (at most) 8 possibilities to obtain $|\Delta\mathbf{a}|^2 = 4d^2$; the number of possibilities is less than 8 when the transmitted 2-D constellation points are near the edges of the constellation.

2. The bit $u_3^{(\ell)}$ is different for the two 4-D symbols. The 2-D symbols $a^{(2\ell)}$ and $\hat{a}^{(2\ell)}$ belong to different subsets $C_{1,m_1}^{(2)}$, with (at most) 4 neighbours at minimum interset distance $\sqrt{2}d$; the same holds for $a^{(2\ell+1)}$ and $\hat{a}^{(2\ell+1)}$. Hence, the corresponding minimum $|\Delta\mathbf{a}|^2$ equals $4d^2$, which is obtained when $\hat{a}^{(2\ell)}$ is at distance $\sqrt{2}d$ from $a^{(2\ell)}$ (at most 4 possibilities), and $\hat{a}^{(2\ell+1)}$ is at distance $\sqrt{2}d$ from $a^{(2\ell+1)}$ (at most 4 possibilities). This corresponds to a total of (at most) 16 possibilities to obtain $|\Delta\mathbf{a}|^2 = 4d^2$; the number of possibilities is less than 16 when the transmitted 2-D constellation points are near the edges of the constellation.

When $\hat{\mathbf{a}}$ differs from $\mathbf{a}$ in more than one trellis section, it can be verified from the trellis that $|\Delta\mathbf{a}|^2 \geq 5d^2$. Hence, at high SNR, the upper bound (5.14) on BER is dominated by the coded sequences $\hat{\mathbf{a}}$ that differ from $\mathbf{a}$ in one trellis section only, yielding $|\Delta\mathbf{a}|^2 = 4d^2$. Approximating BER by keeping in (5.14) only the terms with $|\Delta\mathbf{a}|^2 = 4d^2$, we obtain

$$\mathrm{BER} \approx \sum_{\substack{\mathbf{a},\hat{\mathbf{a}} \\ |\Delta\mathbf{a}| = 2d}} \frac{N_{\mathrm{info}}(\mathbf{a},\hat{\mathbf{a}})}{(\log_2(M) - 1/2)\, N_{\mathrm{t}}} \Pr[\mathbf{a}] Q\left(\sqrt{2d^2\mathrm{SNR}}\right) \tag{5.15}$$

A simple estimate of the factor in front of $Q\left(\sqrt{2d^2\mathrm{SNR}}\right)$ in (5.15) is obtained by assuming that for each $\mathbf{a}$ there are $12N_{\mathrm{t}}$ sequences $\hat{\mathbf{a}}$ for which $|\Delta\mathbf{a}|^2 = 4d^2$ (i.e., $N_{\mathrm{t}}/2$ trellis sections and 24 parallel transitions at minimum distance from the correct transition), and that half the number of information bits in the considered trellis section is wrong (i.e., $(\log_2(M) - 1/2)$ erroneous information bits); this yields

$$\mathrm{BER} \approx 12Q\left(\sqrt{2d^2\mathrm{SNR}}\right) \tag{5.16}$$

Note that the factor of 12 involves an overestimation of the number of coded sequences $\hat{\mathbf{a}}$ at minimum distance from $\mathbf{a}$, especially for small constellations.

5.2.3 Nonlinear Precoding

5.2.3.1 Detection Algorithm

In the case of nonlinear precoding, the observation model (5.2) holds, with the user index removed for notational convenience. The symbol sequence $\hat{\mathbf{a}} = \left(\hat{a}^{(1)}, \hat{a}^{(2)}, ..., \hat{a}^{(N_{\mathrm{t}})}\right)$ resulting from ML sequence detection is obtained as $\hat{\mathbf{a}} =$

$\arg\min_{\tilde{\mathbf{a}}\in\mathcal{C}} D_{\mathrm{NLP}}(\tilde{\mathbf{a}})$, where the minimization of $D_{\mathrm{NLP}}(\tilde{\mathbf{a}})$ is over the set $\mathcal{C}$ of valid coded symbol sequences. $D_{\mathrm{NLP}}(\tilde{\mathbf{a}})$ is given by

$$D_{\mathrm{NLP}}(\tilde{\mathbf{a}}) = \min_{\{\tilde{l}^{(k)}\}} \sum_{k=1}^{N_{\mathrm{t}}} \left| z^{(k)} - \sqrt{\mathrm{SNR}^{(k)}}(\tilde{a}^{(k)} + \tilde{l}^{(k)} A^{(k)}) \right|^2 \tag{5.17}$$

$$= \min_{\{\tilde{l}^{(k)}\}} \sum_{k=1}^{N_{\mathrm{t}}} \left| \left[z^{(k)} \right]_{\sqrt{\mathrm{SNR}^{(k)}} A^{(k)}} - \sqrt{\mathrm{SNR}^{(k)}}(\tilde{a}^{(k)} + \tilde{l}^{(k)} A^{(k)}) \right|^2 \tag{5.18}$$

and the minimization in (5.17) is over the complex numbers with integer real and imaginary parts. It follows from (5.18) that $\{\left[z^{(k)}\right]_{\sqrt{\mathrm{SNR}^{(k)}}A^{(k)}}, k = 1, ..., N_{\mathrm{t}}\}$ is a sufficient statistic for the detection of $\mathbf{a}$.

The branch metric to be used in the Viterbi algorithm for a given state transition at the ℓth trellis section equals $\min_{\tilde{a}^{(2\ell)},\tilde{a}^{(2\ell+1)}} \left(D_{\mathrm{NLP}}^{(2\ell)}(\tilde{a}^{(2\ell)}) + D_{\mathrm{NLP}}^{(2\ell+1)}(\tilde{a}^{(2\ell+1)}) \right)$, where the minimization is over the 4-D symbols $(\tilde{a}^{(2\ell)}, \tilde{a}^{(2\ell+1)})$ associated with the parallel transitions corresponding to the considered state transition, and

$$D_{\mathrm{NLP}}^{(k)}(\tilde{a}^{(k)}) = \min_{\tilde{l}^{(k)}} \left| \left[z^{(k)} \right]_{\sqrt{\mathrm{SNR}^{(k)}} A^{(k)}} - \sqrt{\mathrm{SNR}^{(k)}}(\tilde{a}^{(k)} + \tilde{l}^{(k)} A^{(k)}) \right|^2 \tag{5.19}$$

In the case where a 4-QAM symbol $a^{(k)}$ is the result of the pairing of two 2-QAM symbols $a^{(k_1)}$ and $a^{(k_2)}$ with associated observations $z^{(k_1)}$ and $z^{(k_2)}$, we have

$$D_{\mathrm{NLP}}^{(k)}(\tilde{a}^{(k)}) = \min_{\tilde{l}^{(k_1)}} \left| \left[z^{(k_1)} \right]_{\sqrt{\mathrm{SNR}^{(k_1)}} A^{(k_1)}} - \sqrt{\mathrm{SNR}^{(k_1)}}(\tilde{a}^{(k_1)} + \tilde{l}^{(k_1)} A^{(k_1)}) \right|^2$$
$$+ \min_{\tilde{l}^{(k_2)}} \left| \left[z^{(k_2)} \right]_{\sqrt{\mathrm{SNR}^{(k_2)}} A^{(k_2)}} - \sqrt{\mathrm{SNR}^{(k_2)}}(\tilde{a}^{(k_2)} + \tilde{l}^{(k_2)} A^{(k_2)}) \right|^2$$

where $\tilde{a}^{(k_1)} = (1+j)\Re(\tilde{a}^{(k)})$ and $\tilde{a}^{(k_2)} = (1+j)\Im(\tilde{a}^{(k)})$.

5.2.3.2 Error Performance

The upper bound on the BER is still given by (5.13), but in the case of NLP we have

$$\mathrm{PEP}(\hat{\mathbf{a}}|\mathbf{a}) = \sum_{\{l^{(k)}\}} Q\left(\sqrt{\frac{1}{2} \sum_{k=1}^{N_{\mathrm{t}}} |\Delta a^{(k)} + l^{(k)} A^{(k)}|^2 \mathrm{SNR}^{(k)}} \right)$$

where $l^{(k)}$ (with $k = 1, ..., N_{\mathrm{t}}$) takes values from $\mathbb{Z} + j\mathbb{Z}$, When the constellations and the SNRs are the same for all N_{t} tones, i.e., $M^{(k)} = M$ and $\mathrm{SNR}^{(k)} = \mathrm{SNR}$ for all k, we obtain again the BER approximation (5.15), but

now $\hat{\mathbf{a}}$ belongs to the set $\mathcal{C}_{\text{ext}}$ containing not only all allowed coded symbol sequences from $\mathcal{C}$ but also the sequences for which the modulo reduction (the kth component $\hat{a}^{(k)}$ is reduced modulo $A^{(k)}$, $k = 1, ..., N_{\text{t}}$) belongs to $\mathcal{C}$. As there are more sequences in this augmented set $\mathcal{C}_{\text{ext}}$ at minimum distance $2d$ from the transmitted sequence than in the set $\mathcal{C}$, the resulting BER for NLP is larger than for LP, because of a larger coefficient in front of $Q\left(\sqrt{2d^2\text{SNR}}\right)$.

5.2.4 Rule of Thumb

Making a similar reasoning as in Section 5.1.3, it follows from the approximate BER expression (5.15) that doubling the number of constellation points from 2^b to 2^{b+1} requires the SNR to be increased by about 3 dB in order to maintain the same TCM error performance (assuming SNR and 2^b to be large).

Now let us compare the error performances of TCM and uncoded transmission. Comparing the arguments of the function $Q(.)$ in (5.15) and (5.4) or (5.9), we observe that for a given constellation, uncoded transmission requires a 6 dB higher SNR, compared to TCM, in order to achieve the same error performance as TCM (assuming large SNR). However, when uncoded transmission and TCM use the same constellation, they operate at different information bitrates. Taking into account that TCM introduces 1 parity bit per N_{dim} QAM symbols (in G.fast, we have $N_{\text{dim}} = 2$, which is referred to as 4D TCM) a 2^b-QAM constellation carries $b - (1/N_{\text{dim}})$ information bits. Hence, for a fair comparison, we should compare TCM with a 2^b-QAM constellation to uncoded $2^{b-(1/N_{\text{dim}})}$-QAM. It follows from Section 5.1.3 that, compared to uncoded 2^b-QAM, uncoded $2^{b-(1/N_{\text{dim}})}$-QAM requires about $3/N_{\text{dim}}$ dB less SNR in order to achieve the same error performance. Hence, comparing TCM using 2^b-QAM to uncoded $2^{b-(1/N_{\text{dim}})}$-QAM (yielding the same information bitrate), the former has a coding gain of $6 - (3/N_{\text{dim}})$ dB; for N_{dim} equal to 1, 2 and 4, the corresponding coding gains are 3 dB, 4.5 dB and 5.25 dB, respectively, which illustrates the performance advantage of using multi-dimensional TCM compared to the 2D TCM (i.e., $N_{\text{dim}} = 1$) originally proposed by Ungerboeck.

When comparing uncoded 2^b-QAM to uncoded $2^{b-(1/N_{\text{dim}})}$-QAM, we applied the rule of thumb from Section 5.1.3 to uncoded QAM in spite of a *non-integer* number of information bits (i.e., $b - (1/N_{\text{dim}})$ bits) per QAM symbol. In a practical setting, the corresponding information bitrate could be achieved by time-multiplexing uncoded 2^b-QAM and uncoded 2^{b-1}-QAM, with fractions of the time equal to $1 - (1/N_{\text{dim}})$ and $1/N_{\text{dim}}$, respectively. Denoting by E the constellation energy needed for uncoded 2^b-QAM to achieve a certain error performance, an average constellation energy for the time-multiplexed signal equal to $(1 - (1/N_{\text{dim}}))E + (1/N_{\text{dim}})(E/2) = (1 - (1/(2N_{\text{dim}})))E$ yields essentially the same error performance, but at an SNR which is $-10\log(1 - (1/(2N_{\text{dim}})))$ dB less than for uncoded 2^b-QAM; for N_{dim} equal to 1, 2 and 4, the corresponding advantage of the time-multiplex compared to uncoded 2^b-QAM amounts to

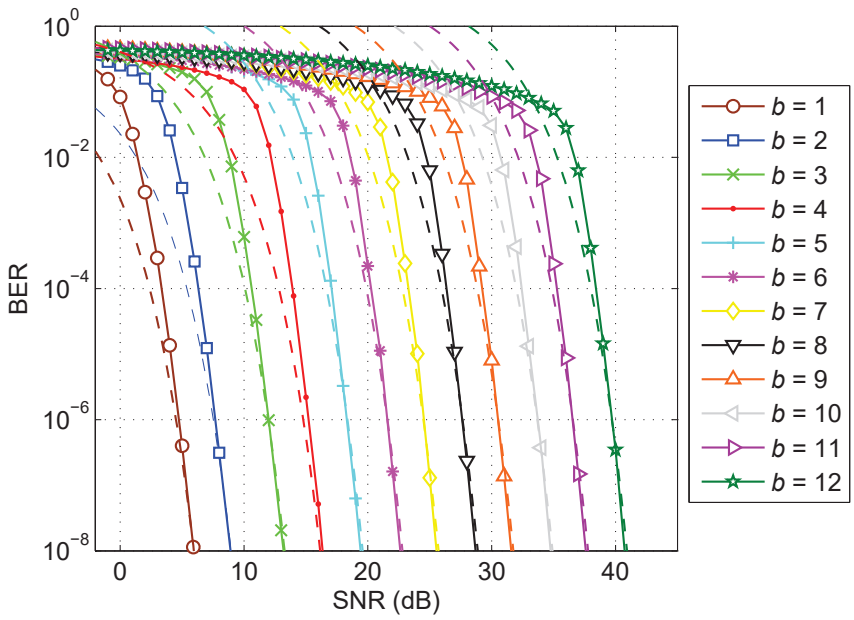

Figure 5.12: BER performance of TCM with LP for constant constellation size, simulations (solid lines with markers) versus approximation (dashed lines).

3 dB, 1.25 dB and 0.58 dB. The rule from Section 5.1.3 indicates an advantage of about $3/N_{\text{dim}}$ dB for uncoded $2^{b-(1/N_{\text{dim}})}$ compared to uncoded 2^b-QAM, which for N_{dim} equal to 1, 2 and 4 corresponds to 3 dB, 1.5 dB and 0.75 dB; note that these results are at most only a few tenths of a dB different from the results for the time-multiplex.

5.2.5 Numerical Results

In Figure 5.12, the BER performance is shown for TCM with LP under the assumption that on each tone a 2^b-QAM constellation is used, and all tones have the same SNR. The considered constellations range from 2-bit QAM to 12-bit QAM. For each constellation size, both the BER resulting from simulations and the approximated BER is given. For $b \geq 5$ we use the approximation $12Q\left(\sqrt{2d^2\text{SNR}}\right)$, which turns out to be quite accurate for small BER. For $b < 5$ this approximation is less accurate (the number of codewords at minimum distance is severely overestimated), so we use (5.15) instead, which provides an accurate approximation at low BER.

Figure 5.13 shows the BER performance for TCM with NLP; again both the BER resulting from simulations and the approximated BER is given. Due to the

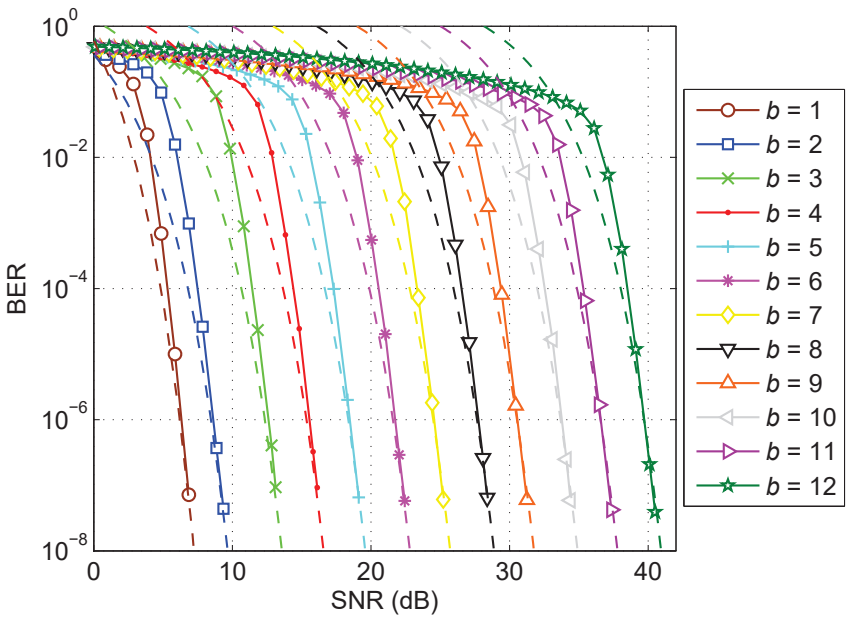

Figure 5.13: BER performance of TCM with NLP for constant constellation size, simulations (solid lines with markers) versus approximation (dashed lines).

augmented set of allowed coded symbol sequences at minimal distance for NLP, the approximation $12Q\left(\sqrt{2d^2\mathrm{SNR}}\right)$ turns out to be accurate at low BER for smaller constellations ($b \geq 3$) than was the case for LP. For $b = 1$ and $b = 2$, we use (5.15) instead. For these constellations, a trellis section represents only the 4 bits $(u_3^{(\ell)}, u_2^{(\ell)}, u_1^{(\ell)}, u_0^{(\ell)})$. The only possible parallel transition would result in an error in bit $u_3^{(\ell)}$, corresponding to erroneous symbols on both tones in the trellis section. The number of codewords at minimum distance is 256 for $b = 1$ and 16 for $b = 2$, and all 3 transmitted information bits in the trellis section are in error. Based on these observations, (5.15) yields BER $= 256Q\left(\sqrt{2d^2\mathrm{SNR}}\right)$ and BER $= 16Q\left(\sqrt{2d^2\mathrm{SNR}}\right)$ for $b = 1$ and $b = 2$ respectively.

To verify the assumption that dominant error events involve only the uncoded bits corresponding to parallel transitions, we performed some additional simulations for LP with 16-QAM and $N_t = 2048$. Figure 5.14 shows the ratio η_{CW} of the number of codewords with decoding errors in the uncoded information bits only, to the total number of erroneously decoded codewords. We observe that η_{CW} increases with increasing SNR, and reaches the value of about 90% for SNR $\approx$ 16 dB (where BER $\approx 10^{-7}$). Figure 5.15 shows the ratio η_{info} of

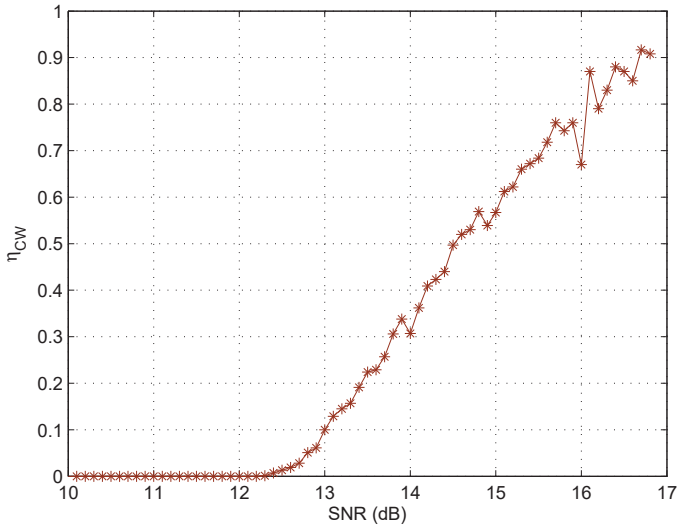

Figure 5.14: η_{CW} for TCM and constant constellation size ($b = 4$).

the number of erroneous uncoded information bits to the total number of erroneous information bits. This ratio increases with increasing SNR, and reaches a value of about 85% for SNR $\approx$ 16 dB. These results illustrate that at large SNR the errors involving coded information bits can be safely ignored compared to the errors involving only uncoded information bits.

It can be verified from the trellis diagram that paths which leave the original path reemerge with the original transmitted path after at least 3 trellis sections and have a minimal squared distance of $5d^2$ to the original transmitted sequence, while the parallel transitions have a minimum squared distance of $4d^2$. We validate this observation by means of computer simulations. For LP with normalized 16-QAM ($d^2 = 2/5$) and $N_t = 2048$, we have obtained by means of simulation the squared distances $\sum_{k=1}^{N_t} |\Delta a_i^{(k)}|^2$ for the codewords containing errors in the encoded information bits. Figure 5.16 shows that the average squared distance for these erroneous codewords approaches $5d^2 = 2$ for large SNR values, which is in agreement with the theoretical analysis.

Again by means of simulations, we have investigated the BER performance of the trellis code, operating on (5.1) for LP and on (5.2) for NLP, assuming that the constellation size and the SNR are the same for all tones. Table 5.2 shows $\mathrm{SNR}_{\mathrm{LP}}(b)$ and $\mathrm{SNR}_{\mathrm{NLP}}(b)$ that are required to achieve BER $= 10^{-7}$ for LP and NLP, respectively, assuming a constellation size $M = 2^b$. We observe that $\mathrm{SNR}_{\mathrm{LP}}(b) \leq \mathrm{SNR}_{\mathrm{NLP}}(b)$, indicating that LP yields the better decoder performance; $\mathrm{SNR}_{\mathrm{LP}}(b)$ and $\mathrm{SNR}_{\mathrm{NLP}}(b)$ are essentially the same for large con-

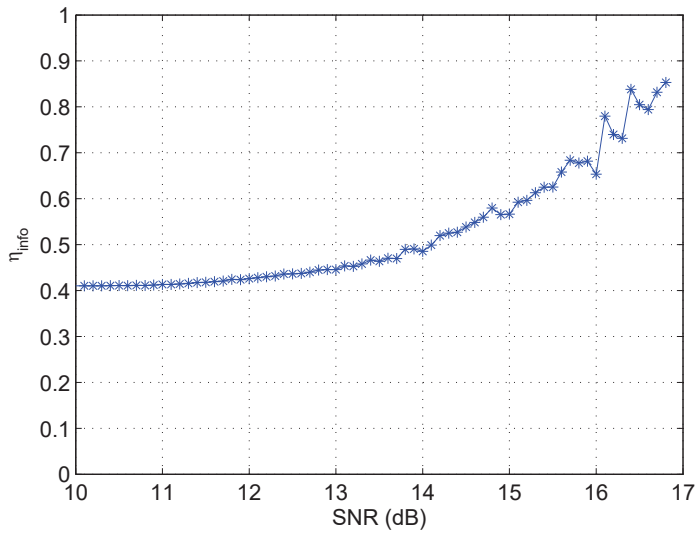

Figure 5.15: η_{info} of TCM for constant constellation size ($b = 4$).

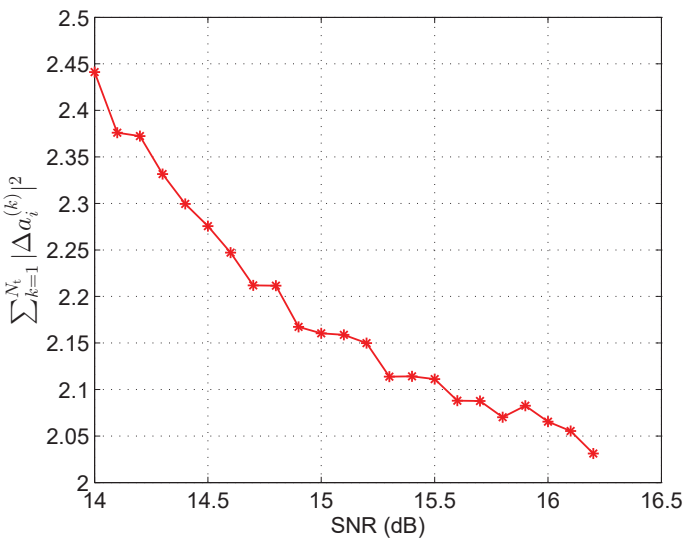

Figure 5.16: Average squared distance of the nonparallel error events, for the normalized 16-QAM constellation.

Table 5.2: Required SNR (in dB) to achieve the target BER of 10^{-7} for TCM with LP and NLP.

b	1	2	3	4	5	6
LP	5.4	8.4	12.6	15.8	18.9	22.1
NLP	6.8	9.1	13.1	16.1	19.0	22.2

b	7	8	9	10	11	12
LP	25.1	28.2	31.1	34.2	37.2	40.2
NLP	25.1	28.3	31.1	34.3	37.2	40.3

stellations, but differ by about 0.5-1.5 dB for $b \leq 3$. As the BER associated with the outer constellation points is affected the most by the constellation expansion caused by the modulo operation, the larger degradation of the smaller constellations is attributed to their larger fraction of outer points.

We have verified that the BER essentially remains at 10^{-7} when the constellation sizes differ among the tones, provided that the SNRs at the decoder input are adjusted such that the SNR for the kth tone ($k = 1, 2, ..., N_{\text{t}}$) equals $\text{SNR}_{\text{LP}}(b^{(k)})$ for LP or $\text{SNR}_{\text{NLP}}(b^{(k)})$ for NLP, when the corresponding constellation size is $M^{(k)} = 2^{b^{(k)}}$. For a specific example of this verification, the reader is referred to Chapters 7 and 8 where the results from this chapter are applied to transmission on real DSL channels.

5.3 LDPC Codes

In Section 5.3.1, we describe the LDPC codes considered in this dissertation. The error performance is analyzed by means of EXIT charts in Section 5.3.4. In Section 5.3.3, we present the relation between LDPC codes and mutual information. We present some numerical results in Section 5.3.7. Here we restrict our attention to the case where IN is absent; the effect of IN will be considered in Section 5.4.

5.3.1 LDPC Code Description

The LDPC codes that we use are the systematic quasi-cyclic LDPC (QC-LDPC) block codes taken from the G.hn standard [51]. Two possible information block lengths are available, $K = 960$ or $K = 4320$. The LDPC code is defined by its parity-check matrix $\mathbf{H}$, specified for following code rates $R_{\text{c}} \in \{1/2, 2/3, 5/6\}$. Furthermore, puncturing patterns are provided to achieve the higher rate codes with $R_{\text{c}} \in \{16/18, 20/21\}$. Those higher rate LDPC codes are obtained from

the 'mother' code with rate $R_c = 5/6$ by discarding both some information and parity bits. The codeword length is then given by $N = K/R_c$.

The check matrix of the QC-LDPC code consists of an array of $\frac{N-K}{h} \times \frac{N}{h}$ circulant $h \times h$ sub-matrices $\mathbf{B}_{i,j}$:

$$\mathbf{H} = \begin{bmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} & \cdots & \mathbf{B}_{1,\frac{N}{h}} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \cdots & \mathbf{B}_{2,\frac{N}{h}} \\ \vdots & & \ddots & \vdots \\ \mathbf{B}_{\frac{N-K}{h},1} & \mathbf{B}_{i,j} & \cdots & \mathbf{B}_{\frac{N-K}{h},\frac{N}{h}} \end{bmatrix} \tag{5.20}$$

The submatrix $\mathbf{B}_{i,j}$ is equal to either a cyclic column shift of the identity matrix or a zero matrix.

As illustration we present the compact form of the check matrix $\mathbf{H}_c$ (the subscript c refers to "compact") corresponding $R_c = 5/6$ and $K = 960$:

$$\mathbf{H}_c = \begin{bmatrix} -1 & 13 & 32 & 47 & 41 & 24 & -1 & 25 & 22 & 40 & 1 & 31 & 8 & 15 & 20 & 15 & 42 & 30 & 13 & 3 & -1 & 0 & -1 & -1 \\ 25 & 46 & 15 & 43 & 45 & 29 & 39 & 47 & 23 & 38 & 39 & 12 & -1 & 21 & -1 & 38 & 33 & 0 & 0 & -1 & 39 & 0 & 0 & -1 \\ 35 & 45 & 45 & 38 & 14 & 16 & 6 & 11 & -1 & 18 & 7 & 41 & 35 & 17 & 32 & 45 & 41 & -1 & 18 & 17 & 0 & -1 & 0 & 0 \\ 9 & 32 & 6 & 22 & 26 & 31 & 9 & 8 & 22 & 32 & 40 & 4 & 18 & 40 & 36 & -1 & -1 & 23 & 31 & 41 & 39 & 20 & -1 & 0 \end{bmatrix} \tag{5.21}$$

where -1 stands for a zero $h \times h$ sub-matrix and each non-negative integer stands for a cyclic column shift of the identity matrix, with the number of right column shifts given by the integer.

Next, the bits in the LDPC codeword are randomly interleaved, and finally the interleaved codeword is mapped to constellation symbols.

5.3.1.1 Encoding Operation

The systematic codeword $\mathbf{c} = (c_N, \ldots, c_1) = (u_K, \ldots, u_1, p_{N-K}, \ldots, p_1)$ consists of an information part and a parity bit part. For this type of LDPC codes, the parity part can easily be generated without determining the systematic generator matrix $\mathbf{G}$ from $\mathbf{H}$ such that $\mathbf{G}\mathbf{H}^{\mathrm{T}} = 0$, normally used for encoding. The codeword $\mathbf{c} = (\mathbf{u}|\mathbf{p})$ satisfies the parity check equations, i.e. $\mathbf{c}\mathbf{H}^{\mathrm{T}} = 0$:

$$(\mathbf{u}|\mathbf{p}) \cdot \mathbf{H}^{\mathrm{T}} = 0 \tag{5.22}$$

$$(\mathbf{u}|\mathbf{p}) \cdot \begin{pmatrix} \mathbf{H}_1^{\mathrm{T}} \\ \mathbf{H}_2^{\mathrm{T}} \end{pmatrix} = 0 \tag{5.23}$$

$$\mathbf{u} \cdot \mathbf{H}_1^{\mathrm{T}} + \mathbf{p} \cdot \mathbf{H}_2^{\mathrm{T}} = 0 \tag{5.24}$$

where the check matrix $\mathbf{H}^{\mathrm{T}}$ is decomposed in 2 sub-matrices $\mathbf{H}_1^{\mathrm{T}}$ and $\mathbf{H}_2^{\mathrm{T}}$ respectively corresponding to the first K and last $N-K$ rows of $\mathbf{H}^{\mathrm{T}}$.

Due to the low number of 1's per row in $\mathbf{H}_2^{\mathrm{T}}$, the system can be solved with low complexity to find the parity bits $(p_{N-K}, \ldots, p_1)$.

5.3.1.2 Puncturing

For the sake of illustration, we give here the two puncturing patterns necessary to obtain the high rate codes with $R_{\mathrm{c}} = 16/18$ and $R_{\mathrm{c}} = 20/21$, from the LDPC code defined by the check matrix in (5.21) for $K = 960$:

$$\mathbf{pp}_{1080} = [\underbrace{11\ldots1}_{720}\underbrace{00\ldots0}_{36}\underbrace{11\ldots1}_{360}\underbrace{00\ldots0}_{36}] \tag{5.25}$$

$$\mathbf{pp}_{1008} = [\underbrace{11\ldots1}_{720}\underbrace{00\ldots0}_{48}\underbrace{11\ldots1}_{240}\underbrace{00\ldots0}_{96}\underbrace{11\ldots1}_{48}] \tag{5.26}$$

where the puncturing pattern serves as a mask and the jth bit is omitted from $\mathbf{c}$ if $pp_{N,j} = 0$.

5.3.2 LDPC Decoding

In the absence of IN, the LDPC decoding is executed according to the SPA using the exact computation of the LLRs, as explained in Section 2.3.1.1. In the case of a punctured LDPC code, the decoder executes the SPA on the Tanner graph of the mother code, with the LLRs of the punctured bits set to zero.

Below we outline the soft demapping (i.e., the computation of the LLRs $L_{\mathrm{ch}\to n}$) in the cases of linear and nonlinear precoding.

5.3.2.1 Linear Precoder

In the case of LP , the exact LLR (2.2) of the nth coded bit c_n, which is contained in the observation $z^{(k)}$ from (5.1), is computed as

$$L_{\mathrm{ch}\to n} = \ln \frac{\sum\limits_{\tilde{a}^{(k)} \in \mathcal{A}_{c_n=0}^{(k)}} \exp\left(-\left|z^{(k)} - \sqrt{\mathrm{SNR}^{(k)}}\tilde{a}^{(k)}\right|^2\right)}{\sum\limits_{\tilde{a}^{(k)} \in \mathcal{A}_{c_n=1}^{(k)}} \exp\left(-\left|z^{(k)} - \sqrt{\mathrm{SNR}^{(k)}}\tilde{a}^{(k)}\right|^2\right)} \tag{5.27}$$

where $\mathcal{A}_{c_n=0}^{(k)}$ and $\mathcal{A}_{c_n=1}^{(k)}$ represent the subset of the symbol constellation $\mathcal{A}^{(k)}$ corresponding to $c_n = 0$ and $c_n = 1$, respectively.

We will consider 2^b-QAM constellations (with $b = 1, 2, \ldots, 12$) with the same mapping as for uncoded transmission (i.e., Gray mapping for $b = 1$ and even b; close-to-Gray mapping for odd b with $b \geq 3$), which minimizes the average number of bit differences between constellation points at minimum distance.

5.3.2.2 Nonlinear Precoder

In the case of NLP, the LLR (2.2) of the nth coded bit u_n, which is contained in the observation $z^{(k)}$ from (5.2), is computed as

$$L_{\text{ch}\to n} = \ln \frac{\sum\limits_{\tilde{a}^{(k)} \in \mathcal{A}^{(k)}_{\text{ext},c_n=0}} \exp\left(-\left|z^{(k)} - \sqrt{\text{SNR}^{(k)}}\tilde{a}^{(k)}\right|^2\right)}{\sum\limits_{\tilde{a}^{(k)} \in \mathcal{A}^{(k)}_{\text{ext},c_n=1}} \exp\left(-\left|z^{(k)} - \sqrt{\text{SNR}^{(k)}}\tilde{a}^{(k)}\right|^2\right)} \tag{5.28}$$

where $\mathcal{A}^{(k)}_{\text{ext},c_n=0}$ and $\mathcal{A}^{(k)}_{\text{ext},c_n=1}$ represent the subset of the extended symbol constellation $\mathcal{A}^{(k)}_{\text{ext}}$ corresponding to $c_n = 0$ and $c_n = 1$, respectively.

The system with NLP uses the same constellations and mappings as for LP, with one exception for 8-QAM. For 8-QAM-G.fast, we use the mapping from Figure 5.2, instead of the mapping from Figure 5.1 as used with LP. This choice is justified by the fact that the extended symbol constellation with the mapping from Figure 5.2 yields a lower average bit difference between points at minimum distance, as compared to the extended signal constellation with the mapping from Figure 5.1.

5.3.3 Mutual Information of LLRs

Here we consider the average mutual information I_{avg} between a coded bit c_n and its corresponding LLR $L_{\text{ch}\to n}$. We will point out that this mutual information plays an important role in the analysis of the LDPC decoding performance.

Assume that the coded bit c_n is the jth bit of the symbol $a^{(k)}$ which belongs to a $2^{b^{(k)}}$-QAM constellation $\mathcal{A}^{(k)}$. For notational convenience, we represent c_n and the corresponding LLR $L_{\text{ch}\to n}$ by $c_j^{(k)}$ and $L_j^{(k)}$, respectively. The mutual information $I(c_j^{(k)}; L_j^{(k)})$ between $c_j^{(k)}$ and $L_j^{(k)}$ depends on the bit index j and the constellation $\mathcal{A}^{(k)}$, and is denoted $I_{j,\mathcal{A}^{(k)}}$:

$$I_{j,\mathcal{A}^{(k)}} = I(c_j^{(k)}; L_j^{(k)}) \tag{5.29}$$

$$= \sum_{c_j^{(k)} \in \{0,1\}} \int_{L_j^{(k)}} p_{c,L}(c_j^{(k)}, L_j^{(k)}) \log_2 \left(\frac{p_{c,L}(c_j^{(k)}, L_j^{(k)})}{p_c(c_j^{(k)}) \cdot p_L(L_j^{(k)})} \right) \mathrm{d}L_j^{(k)} \tag{5.30}$$

Averaging $I_{j,\mathcal{A}^{(k)}}$ over the $b^{(k)}$ bits of the considered constellation $\mathcal{A}^{(k)}$ yields the average mutual information $I_{\mathcal{A}^{(k)}}$:

$$I_{\mathcal{A}^{(k)}} = \frac{1}{b^{(k)}} \sum_{j=1}^{b^{(k)}} I_{j,\mathcal{A}^{(k)}}$$

Finally, taking into account that $b^{(k)}$ coded bits each yield on average a mutual information equal to $I_{\mathcal{A}^{(k)}}$, the average mutual information I_{avg} over all coded bits is obtained as

$$I_{\text{avg}} = \frac{\sum_{k=1}^{N_t} b^{(k)} I_{\mathcal{A}^{(k)}}}{\sum_{k=1}^{N_t} b^{(k)}}$$

When $L_{\text{ch}\to n}$ is computed according to the exact expressions (5.27) for LP or (5.28) for NLP, $L_{\text{ch}\to n}$ is a sufficient statistic, implying that $L_{\text{ch}\to n}$ and $z^{(k)}$ contain the same information about the coded bit c_n. In this case, $I_{j,\mathcal{A}^{(k)}}$ can be computed as $I_{j,\mathcal{A}^{(k)}} = I(c_j^{(k)}; z^{(k)})$, which simplifies the computation.

Assuming that all tones use the same constellation and have the same SNR, Figures 5.17 and 5.18 show I_{avg} versus SNR for 2^b-QAM ($b = 1, \ldots, 12$) and respectively LP and NLP; the LLRs are computed according to (5.27) for LP and (5.28) for NLP. Clearly, there is a degradation of I_{avg} when NLP is used, especially for I_{avg} close to 0. This is due to the periodic extensions of the original QAM constellation that need to be considered during computation of the LLRs.

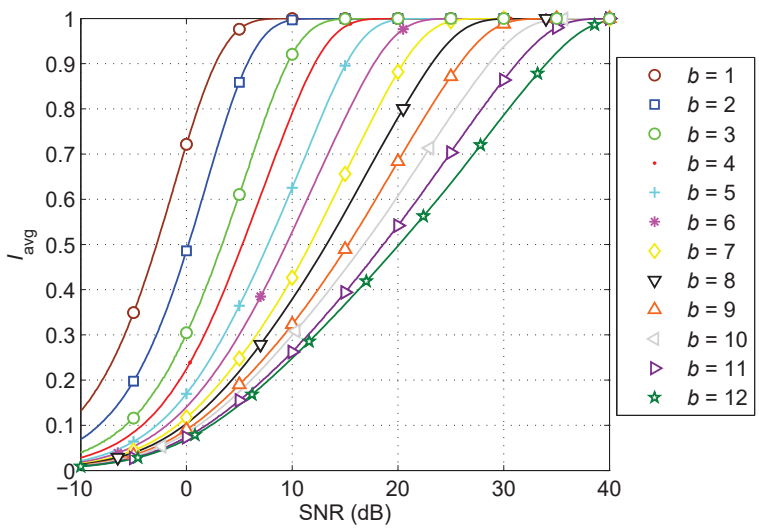

Figure 5.17: Average mutual information per bit I_{avg} for 2^b-QAM ($b = 1, \ldots, 12$), LP.

5.3.4 EXIT Chart Analysis.

The decoding performance of the LDPC codes can be analyzed by examining their EXIT charts (introduced in Section 2.3.2). To approach very low BER,

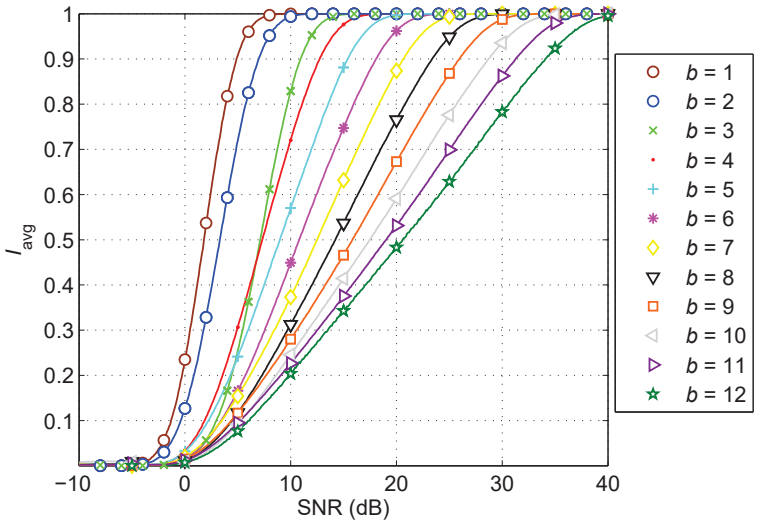

Figure 5.18: Average mutual information per bit I_{avg} for 2^b-QAM ($b = 1, \ldots, 12$), NLP.

the EXIT curves related to the variable nodes and the check nodes must be matched: the former curve must be located above the latter curve, so that their only crossing point is (1,1). If this is the case, we say that 'the tunnel is open', which implies that a mutual information increase can be realized by transferring messages between the variable nodes and the check nodes, with the objective to eventually obtain a mutual information equal to 1. The closer the curves fit together, the smaller the increase in mutual information per iteration and the larger the number of decoding iterations that are needed to reach the convergence point (1,1).

In the following we consider the EXIT charts of the different LDPC codes considered in this dissertation. The mutual information $I_{E,V}\big|_{I_{A,V}=0}$ at the starting point of the variable node EXIT curve equals the average mutual information I_{avg} computed in Section 5.3.3. We assume that all tones use a BPSK constellation and have the same SNR, and that LP is used; hence, the curve for $b = 1$ in Fig. 5.17 shows the relation between I_{avg} and SNR.

For the LDPC code of rate 1/2, Figures 5.19 and 5.20 give the variable node EXIT curves (for SNR equal to 1 dB) and the check node EXIT curves, respectively. The considered LDPC codes are irregular, meaning that multiple variable node degrees and multiple check node degrees occur: the rate 1/2 LDPC code has variable node degrees $d_v \in \{2, 3, 6\}$ and check node degrees $d_c \in \{5, 6, 7\}$. The variable node EXIT curve is the average, over all occurring degrees d_v, of

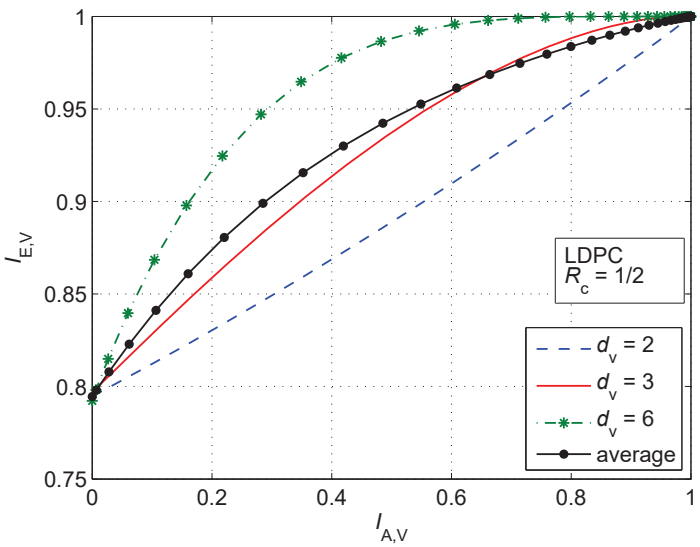

Figure 5.19: Variable node EXIT curves for the LDPC code of rate 1/2 at SNR = 1 dB.

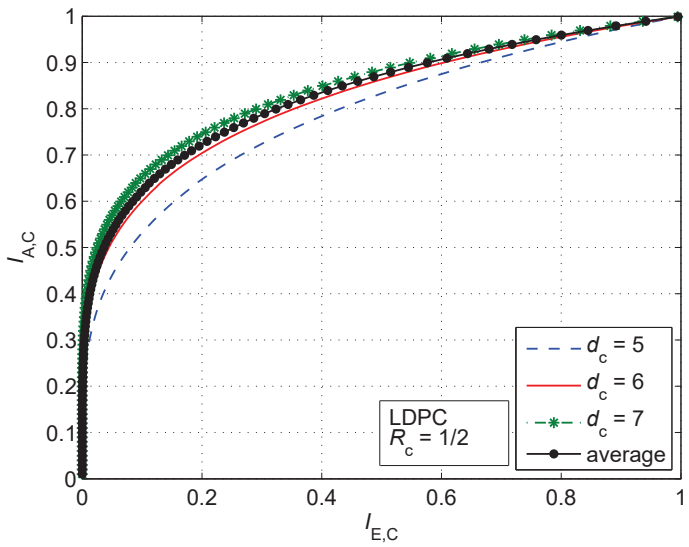

Figure 5.20: Check node EXIT curves for the LDPC code of rate 1/2.

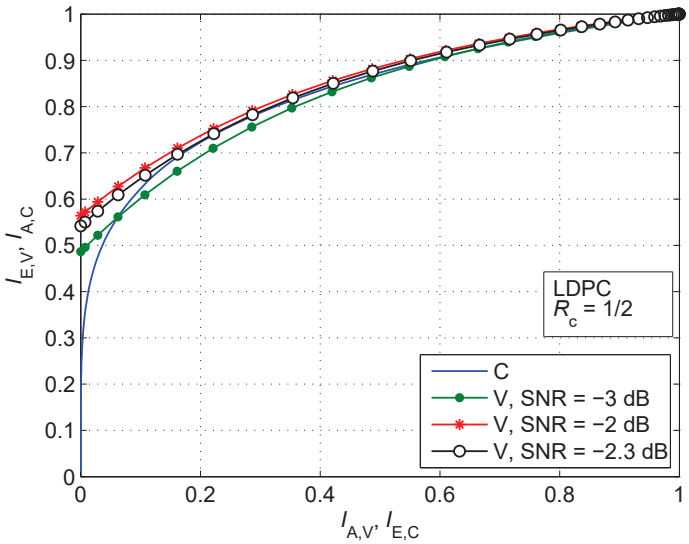

Figure 5.21: EXIT chart for the LDPC code of rate 1/2.

the EXIT curves corresponding to an LDPC code with variable nodes of constant degree d_v, weighted by the fraction of edges incident to variable nodes of degree d_v [14]. Similarly, the check node EXIT curve is the average, over all occurring degrees d_c, of the EXIT curves corresponding to an LDPC code with check nodes of constant degree d_c, weighted by the fraction of edges incident to check nodes of degree d_c.

In Figure 5.21, we try to fit the EXIT curves for the variable nodes and the check nodes, by adjusting the SNR value. For SNR $= -3$ dB, the variable node EXIT curve crosses the check node EXIT curve at an unwanted point different from (1,1); decoding will get stuck after a few iterations and will never converge to a mutual information equal to 1. However, for SNR $= -2$ dB the tunnel is obviously open with a clear gap between the two transfer curves. There exist smaller SNRs for which the tunnel is still open, but for which the curves fit closer together. This is the case for SNR $= -2.3$ dB, for which $I_{\text{avg}} = 0.54$. This implies that for an unlimited number of decoding iterations and an infinite codeword length, an arbitrary low BER can be achieved for SNR $= -2.3$ dB.

Similarly, Figure 5.22 gives the EXIT chart for the LDPC code of rate 5/6. The tunnel is closed for SNR $= 1$ dB, corresponding to $I_{\text{E,V}}(0) = 0.80$. The tunnel opens at SNR $= 2$ dB, yielding $I_{\text{avg}} = 0.86$.

Table 5.3 gives for all considered LDPC code rates the minimum I_{avg} and the corresponding SNR for BPSK at which the tunnel is open. The EXIT chart for a certain LDPC code rate is valid for both the shorter block length with

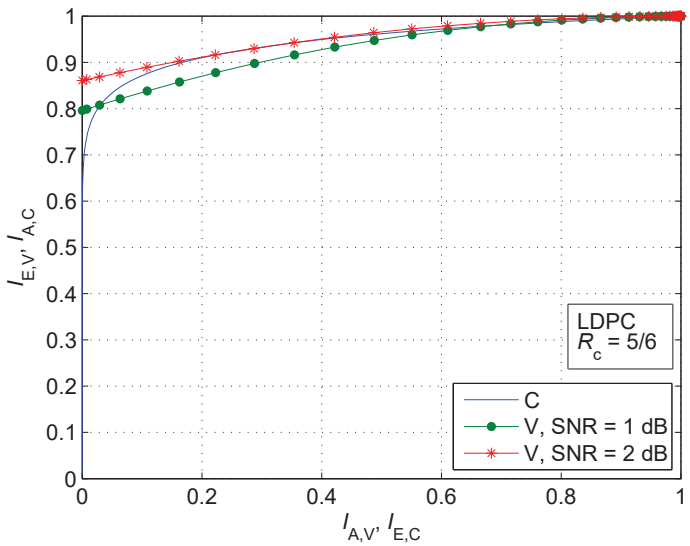

Figure 5.22: EXIT chart for the LDPC code of rate 5/6.

Table 5.3: SNR and mutual information I_{avg} for which the tunnel is open in the EXIT chart for all LDPC code rates.

R_c	1/2	2/3	5/6	16/18	20/21
SNR (for BPSK)	-2.3 dB	-0.1 dB	2 dB	3 dB	5 dB
I_{avg}	0.54	0.71	0.86	0.91	0.97

$K = 960$ and the longer block length with $K = 4320$ because for a given rate the longer LDPC codes happen to have the same degree distributions as the shorter LDPC codes at both the variable nodes and the check nodes.

5.3.5 Analysis of Finite-Length LDPC Codes

In Section 5.3.4 we have pointed out that the value of the average mutual information I_{avg} determines whether the tunnel in the EXIT chart is open, which indicates whether an infinitely long LDPC code, characterized by its variable node and check node degree distributions, can achieve an arbitrarily low error probability. By means of simulations, it has been shown in [52] that I_{avg} also characterizes the bit error rate (BER) and the word error rate (WER) of finite-length LDPC codes:

- Consider an LDPC code, with all coded bits mapped to a constellation

$\mathcal{A}_1$ and all resulting constellation symbols operating at SNR = SNR_1, which yields $I_{\text{avg}} = I_{\text{avg},1}$. Consider the same LDPC code, with all coded bits mapped to a constellation $\mathcal{A}_2$ and all resulting constellation symbols operating at SNR = SNR_2, which yields $I_{\text{avg}} = I_{\text{avg},2}$. When SNR_2 is adjusted such that $I_{\text{avg},1} = I_{\text{avg},2}$, than it turns out that the configurations $(\mathcal{A}_1, \text{SNR}_1)$ and $(\mathcal{A}_2, \text{SNR}_2)$ yield the same BER and the same WER.

- The above result can be generalized to the case where subsets of coded bits are mapped to different constellations, each operating at different SNRs: configurations yielding the same value of I_{avg} give rise to the same BER and the same WER.

This powerful result from [52] allows the following analysis method.

- **Step 1**: We consider the case where all bits from the LDPC code are mapped to a same selected QAM constellation and all resulting constellation symbols operate at the same SNR. By means of computer simulations, we determine the BER and WER as a function of SNR.

- **Step 2**: We make use of the relation between I_{avg} and SNR for the QAM constellation selected in step 1 to determine the BER and WER from step 1 as a function of I_{avg}.

- **Step 3**: Making use of the relations between I_{avg} and SNR for all 2^b-QAM constellations ($b = 1, ..., 12$), it is possible to compute I_{avg} according to Section 5.3.3 for (i) the configuration where the coded bits are mapped to a constellation different from the one selected in step 1 and the resulting constellations symbols operate at a same SNR; or (ii) even for the more general configuration where subsets of coded symbols are mapped to different constellations, which operate at different SNRs. Having obtained the value of I_{avg} for the considered configuration, the corresponding BER or WER follows from the relation expressing the BER or WER as a function of I_{avg}, derived in step 2.

Hence, in order to determine the error performance for a very wide range of configurations mentioned in step 3, we only need for each of the considered LDPC codes a table containing the BER and WER as a function of SNR for a *single* QAM constellation, and for each considered QAM constellation we need a table containing I_{avg} as a function of SNR. The advantage of this analysis method is that only a limited number of (time-consuming) LDPC decoder simulations is needed.

5.3.6 Rule of Thumb

Unlike the cases of uncoded transmission or TCM, for LDPC codes we have no expression which relates the BER to the minimum distance between the non-

normalized constellation symbols $\sqrt{\mathrm{SNR}^{(k)}}a^{(k)}$. Instead, the BER is related to the mutual information I_{avg}.

Nevertheless, for a given QAM constellation, we can compare from simulation results the SNRs for LDPC coding and for uncoded transmission which give rise to a target value of the BER. Suppose that LDPC coding with rate $R = K/N$ and mapping to 2^b-QAM yields an SNR gain (in dB) of G_{dB}, compared to uncoded 2^b-QAM. For the considered LDPC coding, the number of information bits per QAM symbol equals bR. For a fair comparison, we should consider the coding gain of LDPC with mapping to 2^b-QAM, compared to uncoded 2^{bR}-QAM, so that both systems have the same information bitrate. Following the same reasoning as in Section 5.2.4, this coding gain equals $G_{\mathrm{dB}} - 3b(1-R)$ dB.

5.3.7 Numerical Results

To analyze the error performance of the proposed LDPC codes, for each rate and code length, and combined with both LP and NLP, exhaustive simulations are required. In Figure 5.23, the BER performance is shown for the LDPC code of rate 5/6 and with $K = 960$ and LP for 2^b-QAM with $b = 1, \ldots, 12$, under the assumption of constant constellation size and SNR on all tones and application of SPA decoding. The LDPC decoder is limited to maximum 50 decoding iterations.

From the results depicted in Figures 5.23 and 5.18 (i.e., BER versus SNR and I_{avg} versus SNR), we obtain the BER versus I_{avg}, for each of the considered constellations. These results are shown in Figure 5.24. Apparently, the BER versus I_{avg} is approximately independent of the constellation size 2^b, which confirms the results from [52], mentioned in Section 5.3.5. We have verified that this result holds for all considered LDPC code rates and block lengths. Hence, the BER for a certain LDPC code is nearly completely determined by the average mutual information I_{avg}. This result can be extended to the WER as shown in Figure 5.25; the WER versus I_{avg} is also approximately independent of the constellation size 2^b.

Obtaining the BER of the LDPC decoder as a function of SNR, for BER values down to a target BER of 10^{-7}, requires very long simulations. In order to limit the simulation time, we make use of the property that the BER for a certain value of I_{avg} is approximately constant for all constellation sizes. As a reference, we will use the BER versus SNR curve for 2^6-QAM, which is a constellation with "average" constellation size because b is limited to 12 bit maximum. Figure 5.26 shows the BER versus I_{avg} for 2^6-QAM, for both the shorter and longer block lengths with respectively $K = 960$ and $K = 4320$ and $R_{\mathrm{c}} \in \{1/2, 2/3, 5/6, 16/18, 20/21\}$. For a certain rate R_{c}, the two curves for $K = 960$ and $K = 4320$ first coincide in the region of large BER, but then diverge from each other. The curve corresponding to the code with $K =$

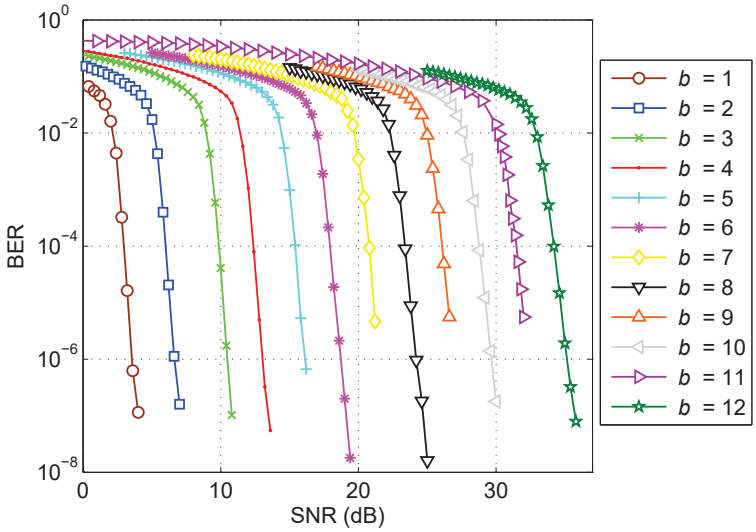

Figure 5.23: BER performance versus SNR of the LDPC code with $R_c = 5/6$, $K = 960$ and LP for constant constellation size, simulations results with SPA decoding.

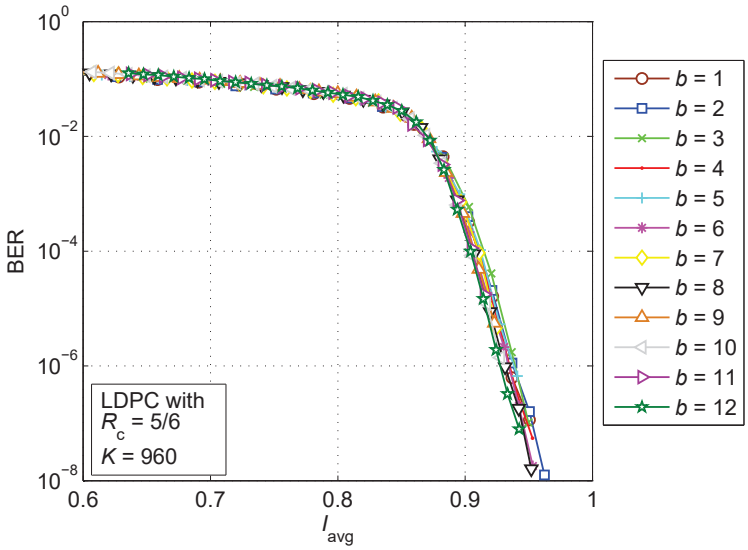

Figure 5.24: BER performance versus I_{avg} of the LDPC code with $R_c = 5/6$, $K = 960$ and LP for constant constellation size, simulations results with SPA decoding.

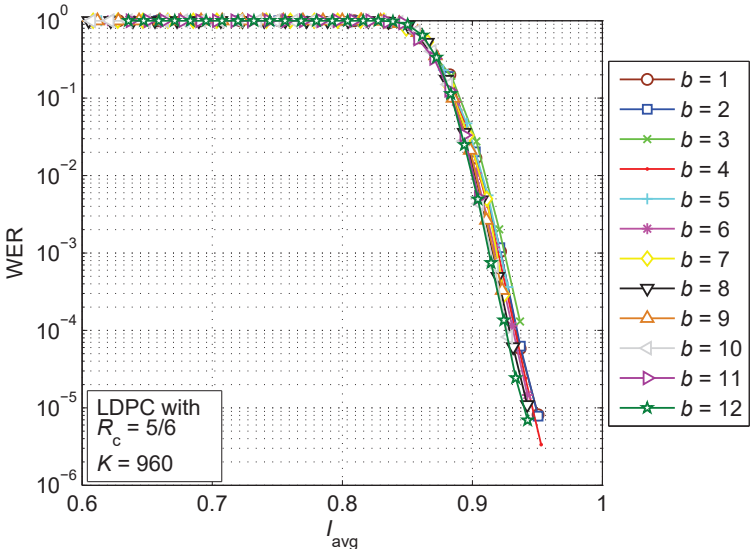

Figure 5.25: WER performance versus I_{avg} of the LDPC code with $R_{\text{c}} = 5/6$, $K = 960$ and LP for constant constellation size, simulations results with SPA decoding.

4320 decays more rapidly than the shorter code and has therefore a better performance in the region of low BER.

It is interesting to compare Figure 5.26 with Table 5.3, which contains the values of I_{avg} for which the tunnel in the EXIT chart opens. We observe that the values of I_{avg} from Table 5.3 are close to the abscissa of the points where the curves in Figure 5.26 for the short codes and the long codes start to diverge. Hence, for a code length growing to infinity, we expect the BER curve to descend very steeply from a point which is close to the point of divergence observed in Figure 5.26, which is in agreement with this very long code achieving an arbitrary small BER when I_{avg} exceeds the value indicated in Table 5.3.

Similar to the system with LP, the BER for the system with NLP can be obtained from simulations with SPA decoding. In Figure 5.27, we show results versus SNR for a limited number of constellation sizes ($b = 1, \ldots, 6$), under the assumption of constant constellation 2^b−QAM and SNR on all tones. As expected, there is a degradation as compared to LP, which decreases for increasing constellation sizes.

We now combine Figures 5.27 and 5.18, showing respectively the BER versus SNR and I_{avg} versus SNR, to obtain the BER versus I_{avg} for the considered constellations, which is presented in Figure 5.28. Similar as for LP, we can conclude that the BER versus I_{avg} for NLP is essentially independent of the constellation

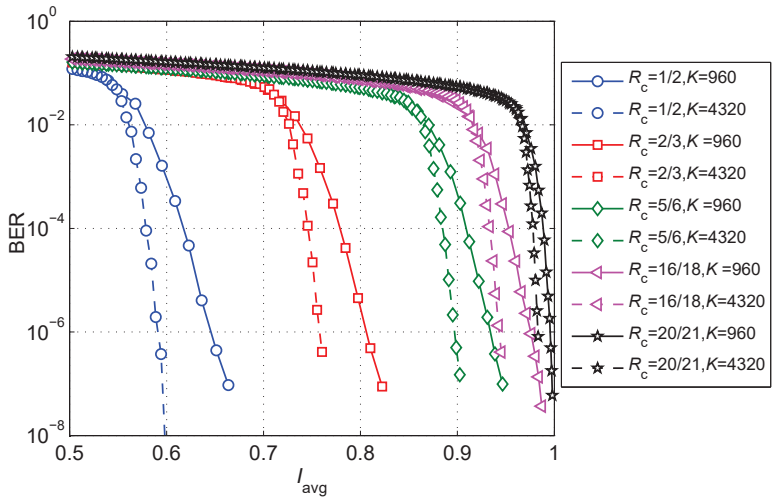

Figure 5.26: BER performance versus I_{avg} of all LDPC codes with $K = 960$ or $K = 4320$, $R_{\text{c}} \in \{1/2, 2/3, 5/6, 16/18, 20/21\}$, LP and 2^6-QAM, SPA decoding.

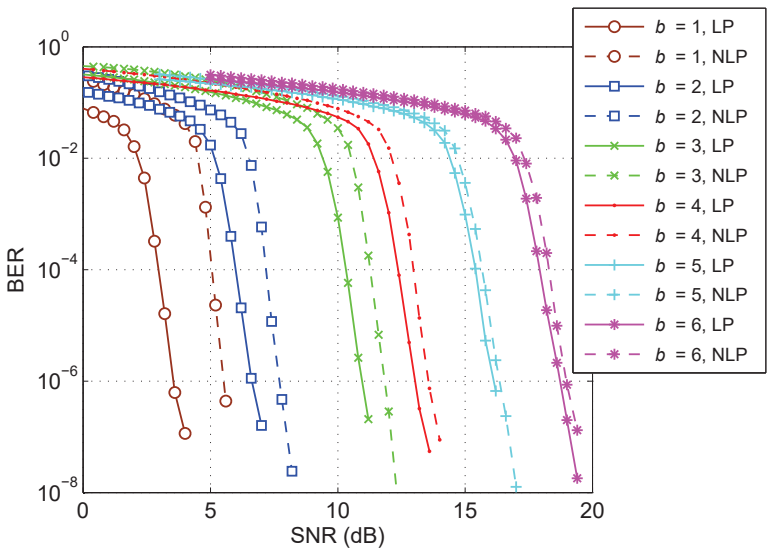

Figure 5.27: Comparison of the BER performance versus SNR for LP and NLP of the LDPC code with $R_{\text{c}} = 5/6$ and $K = 960$, simulations results with SPA decoding.

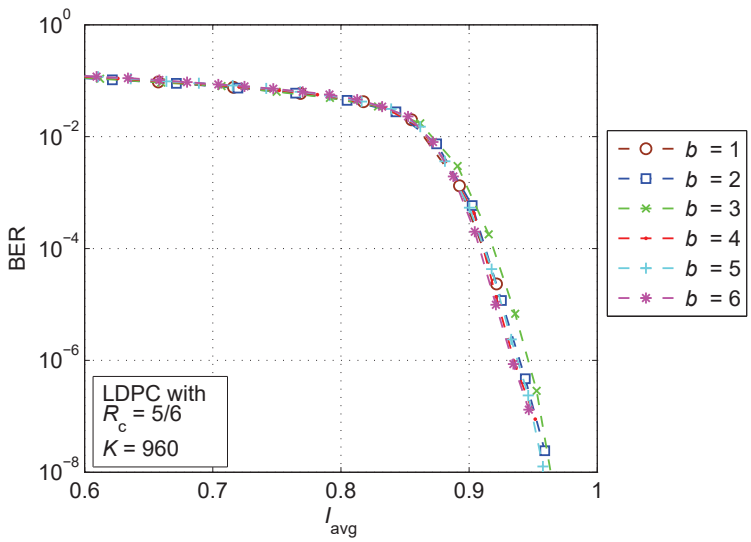

Figure 5.28: BER performance versus I_{avg} of the LDPC code with $R_{\text{c}} = 5/6$, $K = 960$ and NLP, for constant constellation size, simulations results with SPA decoding.

size 2^b. Moreover, the curves for LP and NLP virtually coincide. The BER performance of SPA is thus completely determined by I_{avg}, irrespective of LP or NLP.

From the simulation results performed this far, we derive the value of I_{avg} which is required to achieve BER $= 10^{-7}$ for both LP and NLP, assuming the same constellation size $M = 2^b$ and the same SNR on all tones. Results are given in Table 5.4 for $K = 960$ and $K = 4320$. To obtain the SNR values at which the required I_{avg} is achieved, Figures 5.17 and 5.18 should be consulted which give the relation between SNR and I_{avg} for respectively LP and NLP. For the sake of illustration, Table 5.5 shows the SNR values required to achieve BER $= 10^{-7}$ as a function of the constellation size, for the rate 1/2 LDPC code with $K = 4320$ and SMSA decoding, using NLP. For this setting, an increase of the constellation by 1 bit requires an increase of the SNR by about 2 dB (for large constellations) to maintain the same BER. This in is contrast with the rule of thumb for uncoded transmission and TCM, stating that an increase of the constellation by 1 bit requires a 3 dB higher SNR. Hence, in general *this rule of thumb does not apply to LDPC codes*. This can be understood when assuming that the LDPC code operates close to the Shannon capacity: in this case, for a rate-R_{c} code and a 2^b-QAM constellation, we have $R_{\text{c}} b \approx \log_2(1 + \text{SNR})$ or, equivalently, SNR $\approx 2^{R_{\text{c}} b} - 1$, so that increasing b by 1 bit requires SNR to increase by a factor of about $2^{R_{\text{c}}}$, which corresponds to $3R_{\text{c}}$ dB; for $R_{\text{c}} = 1/2$,

Table 5.4: Required I_{avg} to achieve the target BER of 10^{-7} for all proposed LDPC codes with $K = 960$ and $K = 4320$ and SPA, with LP and NLP.

R_{c}	1/2	2/3	5/6	16/18	20/21
$K = 960$	0.663	0.822	0.946	0.985	0.988
$K = 4320$	0.598	0.764	0.904	0.948	0.985

Table 5.5: Required SNR_{avg} (in dB) to achieve the target BER_{avg} of 10^{-7} for the rate 1/2 LDPC code with $K = 4320$ and SMSA, and NLP.

b	1	2	3	4	5	6
NLP	2.4	4.0	7.2	8.5	10.5	12.4

b	7	8	9	10	11	12
NLP	14.4	16.3	18.5	20.3	22.5	24.3

the resulting SNR increment would be 1.5 dB.

5.4 The Effect of IN

Here we consider the effect of IN on the error performance of the considered uncoded and coded systems with LP and NLP. The IN model and its associated parameters have been introduced in Section 4.5.

5.4.1 Uncoded Transmission and TCM

5.4.1.1 Detection Algorithm

The detection rules for uncoded transmission ((5.3) for LP and (5.8) for NLP) and TCM ((5.10) for LP and (5.18) for NLP), which have been derived for the case where IN is absent, are still valid in the presence of IN, provided that $\text{SNR}^{(k)}$ is replaced by the instantaneous SNR, denoted $\text{SNR}^{(k)}_{\text{act}}$. This seems to suggest that the receiver needs to know the instantaneous SNR on each tone, in order to perform ML detection; however, we will point out that this is not the case when the instantaneous noise variance $N^{(k)}_{0,\text{act}}$ is the same for all tones.

Let us concentrate on the detection of TCM in the case of LP. We have pointed out in Section 4.3 that, for the user considered (user index dropped for notational convenience), $z^{(k)}$ is a scaled and rotated version of the observation $y^{(k)}$, i.e., $z^{(k)} = y^{(k)} \exp(-j \arg(L^{(k)}))/\sqrt{N^{(k)}_{0,\text{act}}}$, where $L^{(k)}$ and $N^{(k)}_{0,\text{act}}$ are

the diagonal element of $\mathbf{L}^{(k)}$ and the instantaneous noise variance, both associated with the considered user. The function $D_{\mathrm{LP}}(\tilde{\mathbf{a}})$ from (5.10), with $\mathrm{SNR}^{(k)}$ replaced by $\mathrm{SNR}_{\mathrm{act}}^{(k)}$, can be transformed into

$$D_{\mathrm{LP}}(\tilde{\mathbf{a}}) = \sum_{k=1}^{N_{\mathrm{t}}} \frac{|L^{(k)}|^2 E^{(k)}}{N_{0,\mathrm{act}}^{(k)}} \left| \frac{y^{(k)}}{L^{(k)}\sqrt{E^{(k)}}} - \tilde{a}^{(k)} \right|^2 \tag{5.31}$$

where $N_{0,\mathrm{act}}^{(k)} = N_0^{(k)}$ or $N_{0,\mathrm{act}}^{(k)} = N_0^{(k)} + N_{0,\mathrm{imp}}$ when the DMT symbol interval is not hit or hit by IN, respectively. Adopting the common assumption that the stationary Gaussian noise $n(t)$ on the channel for the considered user is white, $N_0^{(k)}$ does not depend on the tone index k, and neither does $N_{0,\mathrm{act}}^{(k)}$; hence we set $N_0^{(k)} = N_0$ and $N_{0,\mathrm{act}}^{(k)} = N_{0,\mathrm{act}}$ for $k = 1, ..., N_t$. In this case, *removing* $N_{0,\mathrm{act}}$ from (5.31) does not affect the ML detection, so we can redefine $D_{\mathrm{LP}}(\tilde{\mathbf{a}})$ as

$$D_{\mathrm{LP}}(\tilde{\mathbf{a}}) = \sum_{k=1}^{N_{\mathrm{t}}} |L^{(k)}|^2 E^{(k)} \left| \frac{y^{(k)}}{L^{(k)}\sqrt{E^{(k)}}} - \tilde{a}^{(k)} \right|^2 \tag{5.32}$$

Hence, it is sufficient that the receiver knows $L^{(k)}\sqrt{E^{(k)}}$, and no knowledge about the instantaneous total noise variance $N_{0,\mathrm{act}}$ is needed.

The same reasoning applies to TCM with NLP, and to uncoded transmission[1] (both with LP and NLP). The modifications to be made to the corresponding decision rules in order to remove $N_{0,\mathrm{act}}$ are similar to those for TCM with LP.

5.4.1.2 Error Performance

Let us consider the case where on each tone a 2^b-QAM constellation is used, and all tones have the same value of $\mathrm{SNR}_{\mathrm{act}}^{(k)}$, i.e., $\mathrm{SNR}_{\mathrm{act}}^{(k)} = \mathrm{SNR}_{\mathrm{act}}$. Taking into account that $N_{0,\mathrm{act}}^{(k)}$ does not depend on the tone index, this implies that $|L^{(k)}|^2 E^{(k)}$ does not depend on the tone index either, which occurs (for instance) when the channel transfer function is flat and $E^{(k)}$ does not depend on the tone index. We set $\mathrm{SNR}_{\mathrm{act}} = \mathrm{SNR}$ when the DMT symbol is not hit by IN, and $\mathrm{SNR}_{\mathrm{act}} = \mathrm{SNR}/(1+\kappa)$ when the DMT symbol is hit by IN, with $\kappa = N_{0,\mathrm{imp}}/N_0$. Hence, the average SNR (i.e., signal power over long-term average noise power) on each tone is given by $\mathrm{SNR}_{\mathrm{avg}} = \mathrm{SNR}/(1+p_1\kappa)$, with p_1 denoting the steady-state probability that the considered DMT symbol is hit by IN. The resulting average BER in the presence of IN is given by

$$\mathrm{BER}_{\mathrm{avg}} = p_0 \mathrm{BER}(s=0) + p_1 \mathrm{BER}\,(s=1) \tag{5.33}$$

where $p_0 = 1 - p_1$, while $\mathrm{BER}(s=0)$ and $\mathrm{BER}(s=1)$ denote the BER for the considered system (uncoded transmission or TCM, combined with LP or

[1]For uncoded transmission, it is not needed that $N_0^{(k)}$ is independent of k

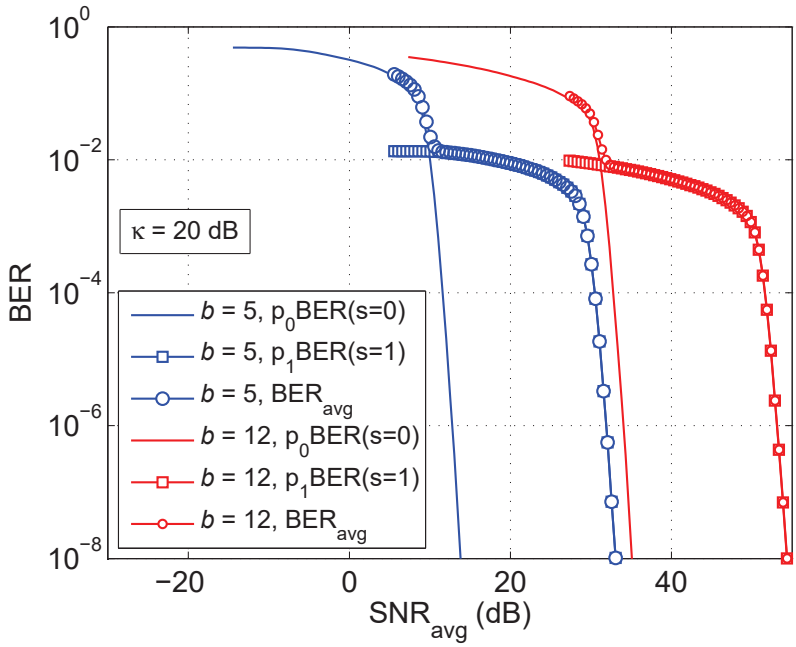

Figure 5.29: BER performance for TCM with NLP, $\kappa = 20$ dB, 32-QAM and 4096-QAM.

NLP) when the DMT symbol is not hit ($s = 0$) and hit ($s = 1$), respectively. For given $\mathrm{SNR}_{\mathrm{avg}}$, the corresponding instantaneous SNRs are $\mathrm{SNR}_{\mathrm{act},s=0} = (1 + p_1\kappa)\mathrm{SNR}_{\mathrm{avg}}$ and $\mathrm{SNR}_{\mathrm{act},s=1} = \frac{1+p_1\kappa}{1+\kappa}\mathrm{SNR}_{\mathrm{avg}}$. Hence, $\mathrm{BER}_{\mathrm{avg}}$ is a function of $\mathrm{SNR}_{\mathrm{avg}}$, κ and p_1. The relations between the instantaneous BERs ($\mathrm{BER}_{s=0}$ and $\mathrm{BER}_{s=1}$) and their corresponding SNRs ($\mathrm{SNR}_{\mathrm{act},s=0}$ and $\mathrm{SNR}_{\mathrm{act},s=1}$) are the same as between BER and SNR in the absence of IN; the latter relation has been investigated in Sections 5.1.4 (for uncoded transmission) and 5.2.5 (for TCM).

5.4.1.3 Numerical Results

Considering TCM with NLP, for 2^5-QAM and 2^{12}-QAM and assuming $p_1 = 0.026$ and $\kappa = 20$ dB, in Figure 5.29 shows $\mathrm{BER}_{\mathrm{avg}}$ from (5.33) as a function of $\mathrm{SNR}_{\mathrm{avg}}$, along with the contributing terms $p_0\mathrm{BER}(s = 0)$ and $p_1\mathrm{BER}\,(s = 1)$. For small BER (i.e., $\mathrm{BER}_{\mathrm{avg}} \ll p_1$), we notice that $\mathrm{BER}_{\mathrm{avg}} \approx p_1\mathrm{BER}\,(s = 1)$, which indicates that the error performance is dominated by the errors that occur in the DMTs which are hit by IN; we have verified that this observation also holds for other system parameter values (other constellation sizes, TCM or uncoded transmission, LP or NLP, $\kappa = 10$ dB or $\kappa = 20$ dB). Tables 5.6-5.9

Table 5.6: Required $\mathrm{SNR}_{\mathrm{avg}}$ (in dB) to achieve the target $\mathrm{BER}_{\mathrm{avg}}$ of 10^{-7} for uncoded transmission with LP and NLP in the presence of IN with $p_1 = 0.026$, $\kappa = 10$ dB.

b	1	2	3	4	5	6
LP	19.4	22.4	26.4	29.2	32.3	35.4
NLP	20.0	22.6	26.7	29.4	32.3	35.5

b	7	8	9	10	11	12
LP	38.3	41.4	44.3	47.3	50.2	53.3
NLP	38.4	41.5	44.4	47.3	50.2	53.3

Table 5.7: Required $\mathrm{SNR}_{\mathrm{avg}}$ (in dB) to achieve the target $\mathrm{BER}_{\mathrm{avg}}$ of 10^{-7} for uncoded transmission with LP and NLP in the presence of IN with $p_1 = 0.026$, $\kappa = 20$ dB.

b	1	2	3	4	5	6
LP	24.4	27.4	31.4	34.2	37.2	40.4
NLP	25.0	27.6	31.7	34.4	37.3	40.4

b	7	8	9	10	11	12
LP	43.3	46.4	49.3	52.3	55.2	58.3
NLP	43.4	46.5	49.3	52.3	55.2	58.3

show, for uncoded transmission and TCM (both with LP and NLP), the value of $\mathrm{SNR}_{\mathrm{avg}}$ (in dB) for which a 2^b-QAM constellation (with $b = 1, ..., 12$) achieves $\mathrm{BER}_{\mathrm{avg}} = 10^{-7}$ in the presence of IN with $p_1 = 0.026$ and $\kappa \in \{10\,\mathrm{dB}, 20\,\mathrm{dB}\}$. Because of the presence of IN, the required SNR values are considerably larger than those from Tables 5.1 (uncoded transmission) and 5.2 (TCM), which refer to the case where IN is absent. The required SNR is smaller for LP than for NLP, but the difference gets smaller with increasing constellation size. In agreement with the rule of thumb, for large constellations the addition of 1 bit (i.e., doubling the number of constellation points) needs an increase of $\mathrm{SNR}_{\mathrm{avg}}$ by about 3 dB in order to maintain $\mathrm{BER}_{\mathrm{avg}} = 10^{-7}$.

5.4.2 LDPC Codes

We have pointed out in Section 5.3.2 that LDPC decoding according to the SPA with the exact LLR computation requires knowledge of the SNR. Assuming that in a practical scenario the presence or absence of IN cannot be detected

Table 5.8: Required $\mathrm{SNR}_{\mathrm{avg}}$ (in dB) to achieve the target $\mathrm{BER}_{\mathrm{avg}}$ of 10^{-7} for TCM with LP and NLP in the presence of IN with $p_1 = 0.026$, $\kappa = 10$ dB.

b	1	2	3	4	5	6
LP	13.8	16.8	21.0	24.2	27.4	30.6
NLP	15.5	17.7	21.7	24.7	27.5	30.8

b	7	8	9	10	11	12
LP	33.5	36.7	39.6	42.8	45.6	48.8
NLP	33.7	36.9	39.7	42.8	45.6	48.8

Table 5.9: Required $\mathrm{SNR}_{\mathrm{avg}}$ (in dB) to achieve the target $\mathrm{BER}_{\mathrm{avg}}$ of 10^{-7} for TCM with LP and NLP in the presence of IN with $p_1 = 0.026$, $\kappa = 20$ dB.

b	1	2	3	4	5	6
LP	18.8	21.8	26.0	29.2	32.3	35.6
NLP	20.5	22.7	26.7	29.6	32.5	35.8

b	7	8	9	10	11	12
LP	38.5	41.7	44.6	47.7	50.6	53.7
NLP	38.7	41.8	44.7	47.8	50.6	53.8

for each individual DMT symbol, the instantaneous SNR is not known to the receiver. One could still perform LDPC decoding using the average SNR instead of the instantaneous SNR; however, in this case the decoder is a "mismatched" (i.e., operating at an SNR which is different from the assumed SNR), yielding an error performance degradation. Therefore we consider here LDPC decoding according to the (S)MSA (see Sections 2.3.1.2 and 2.3.1.3) with the approximate LLR computation, which does not require knowledge of the instantaneous SNR at the receiver.

5.4.2.1 Approximate LLRs

In the case of LP, the approximate LLR messages (2.7) in the MSA and SMSA decoding algorithm from respectively Sections 2.3.1.2 and 2.3.1.3 are computed as

$$L_{\mathrm{ch}\to n} = \min_{\tilde{a}^{(k)}\in\mathcal{A}^{(k)}_{c_n=1}} \left| y^{(k)} - L^{(k)}\sqrt{\mathrm{E}^{(k)}}\tilde{a}^{(k)} \right|^2 - \min_{\tilde{a}^{(k)}\in\mathcal{A}^{(k)}_{c_n=0}} \left| y^{(k)} - L^{(k)}\sqrt{\mathrm{E}^{(k)}}\tilde{a}^{(k)} \right|^2 \tag{5.34}$$

with $y^{(k)}$ from (4.21). In the case of NLP, $y^{(k)}$ is given by (4.30), so that $\mathcal{A}^{(k)}_{c_n=0}$ and $\mathcal{A}^{(k)}_{c_n=1}$ in (5.34) must be replaced by their extended constellations $\mathcal{A}^{(k)}_{\mathrm{ext},c_n=0}$ and $\mathcal{A}^{(k)}_{\mathrm{ext},c_n=1}$.

The approximate LLRs serve as input to the decoding algorithms from Sections 2.3.1.2 and 2.3.1.3 for respectively MSA and SMSA.

5.4.2.2 Error Performance

For the error performance analysis of the LDPC codes in the presence of IN we follow the same reasoning as for uncoded transmission and TCM, outlined in 5.4.1.2.

Considering the case where on each tone a 2^b-QAM constellation is used, and all tones have the same value of $\mathrm{SNR}^{(k)}_{\mathrm{act}}$, i.e., $\mathrm{SNR}^{(k)}_{\mathrm{act}} = \mathrm{SNR}_{\mathrm{act}}$, the expression (5.33) also holds in the case of LDPC codes. The relations between the instantaneous BERs ($\mathrm{BER}_{s=0}$ and $\mathrm{BER}_{s=1}$) and their corresponding SNRs ($\mathrm{SNR}_{\mathrm{act},s=0}$ and $\mathrm{SNR}_{\mathrm{act},s=1}$) are the same as between BER and SNR in the absence of IN. The latter relation has been investigated in Section 5.3.7, but only for decoding according to the SPA with exact LLRs. In Section 5.4.2.3 we will verify that the performance analysis method outlined in Section 5.3.5 still applies for decoding by means of the (S)MSA with approximated LLRs.

5.4.2.3 Numerical Results

First, we investigate for the approximate LLRs the relation between the average mutual information I_{avg} and the instantaneous SNR, for various constellations. For the approximate LLRs, the mutual information $I(c^{(k)}_j, L^{(k)}_j)$ from (5.30)

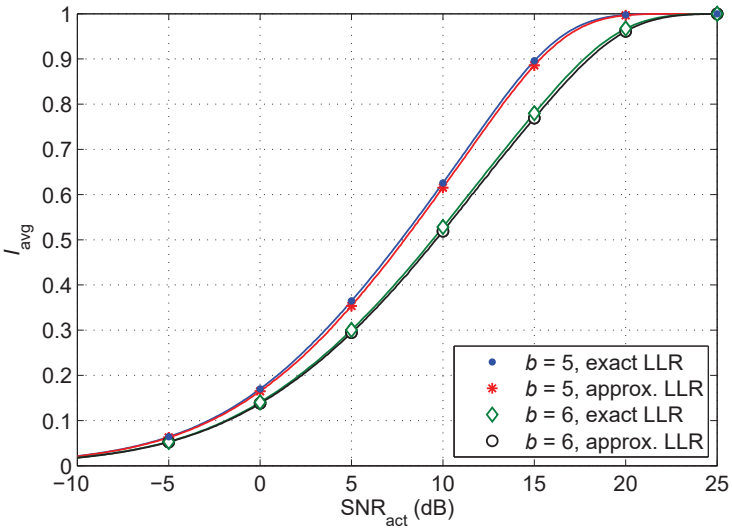

Figure 5.30: Comparison of I_{avg} versus SNR_{act} for exact and approximate LLRs (2^b-QAM ($b = 5, 6$), LP).

is no longer equal to $I(c_j^{(k)}, z^{(k)})$, because the approximate LLRs are not a sufficient statistic. Figure 5.30 compares I_{avg} for the exact and the approximate LLRs, with LP and for both 2^5-QAM and 2^6-QAM. The curves corresponding to a same constellation are very close, indicating that the loss in I_{avg} for the approximate LLRs compared to the exact LLRs is very small.

Now, we will compare the decoding performances of basic MSA and scaled MSA, which have both been introduced in Section 2.3.1. We consider the case where IN is absent; on all tones we have the same constellation size and the same SNR, and LP is used. Figure 5.31 compares the BER versus SNR for the LDPC code of rate $R_c = 5/6$ and $K = 960$ with respectively the SPA decoder, the MSA decoder and the SMSA decoder with parameter $S = 13$. We considered $S \in \{7, 10, 13\}$ based on [13], and found the best BER performance results (with only a minor difference) for $S = 13$. Notice that there is only a very small difference between the two BER curves for respectively SPA and SMSA for a same constellation size. For very small BER, SMSA even slightly outperforms SPA. MSA performs worst and is therefore no longer considered in the following. To check whether the error performance is still completely determined by I_{avg} in the case of a SMSA decoder which uses approximate LLRs, we derive for various constellation sizes the BER as a function of I_{avg}, making use of the BER versus SNR (on all tones we have the same constellation size and the same SNR) and I_{avg} versus SNR. The results are shown in Figure 5.32, for $b \in \{1, 4, 5, 6, 12\}$,

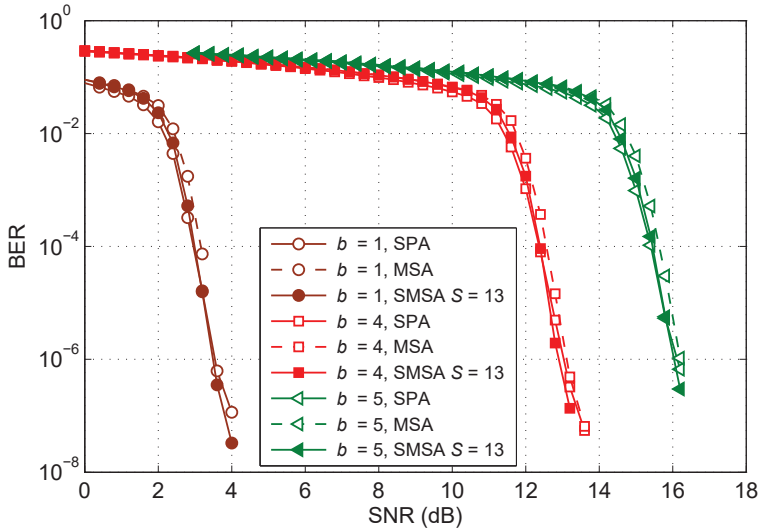

Figure 5.31: Comparison of the BER performance versus SNR for the LDPC code of rate $R_c = 5/6$, $K = 960$ and LP with respectively SPA decoder, MSA decoder and SMSA decoder, simulations results.

assuming a rate 5/6 LDPC code with $K = 960$ and using LP. We observe that the BER versus I_{avg} is still approximately independent of the constellation size 2^b, indicating the validity of the analysis method from Section 5.3.5 also for LDPC decoding with SMSA and approximate LLRs.

Table 5.10 shows the value of I_{avg} which is required to achieve BER $= 10^{-7}$ for all proposed LDPC codes with $K = 960$ and $K = 4320$, and SMSA with $S = 13$ and approximate LLRs for both LP and NLP. Comparing with Table 5.4 which holds for SPA with exact LLRs, we see that SMSA for most code configurations requires a slightly lower I_{avg}; this confirms the better performance at BER $= 10^{-7}$ for SMSA compared to SPA, observed in Figure 5.31 for the rate 5/6 code with $K = 960$.

Having verified the validity of the analysis method from Section 5.3.5 for SMSA with approximate LLRs, we have used this method to determine BER_{avg} versus SNR_{avg} in the presence of IN (on all tones we have the same constellation size and the same instantaneous SNR), based on (5.33). As an illustration, Table 5.11 gives the required SNR_{avg} values for the rate 20/21 LDPC code with $K = 4320$ and SMSA decoding ($S = 13$) to achieve target BER_{avg} equal to 10^{-7}. We consider both LP and NLP. Furthermore, IN of levels $\kappa = -\infty$ (i.e., no IN), 10 dB and 20 dB are considered. We observe that LP outperforms NLP in terms of SNR_{avg} but the difference in performance becomes negligible for large constellations. The stronger the IN, the larger the value of SNR_{avg}

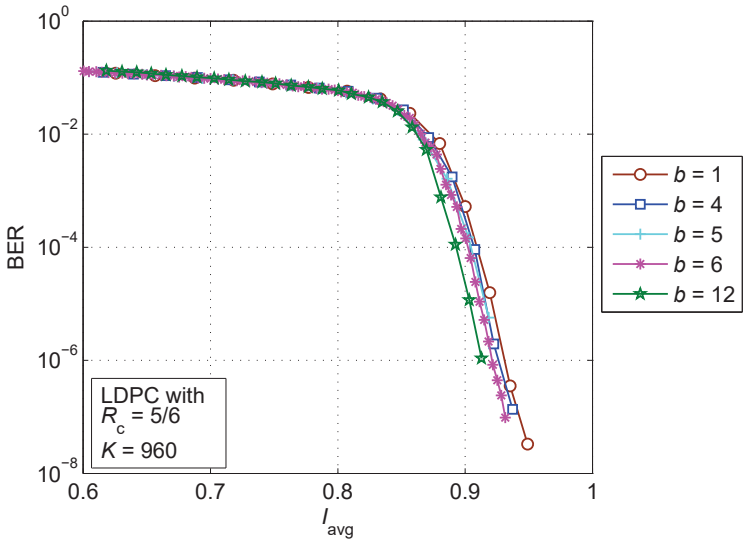

Figure 5.32: BER performance versus I_{avg} of the LDPC code with $R_{\text{c}} = 5/6$, $K = 960$ and LP for constant constellation size, simulations results with SMSA decoding.

needed to maintain $\text{BER}_{\text{avg}} = 10^{-7}$.

5.5 Interleaving against IN

The combination of interleaving with FEC is a means to counteract burst errors caused by IN.

In the arrangement shown in Figure 5.33, a number of codewords are collected at the output of the FEC encoder, and applied to the interleaver. The interleaver performs a random permutation on 'parts' (bits, bytes, constellation symbols) of the codeword, and the output of the interleaver is sent over the chan-

Table 5.10: Required I_{avg} to achieve the target BER of 10^{-7} for all proposed LDPC codes with $K = 960$ and $K = 4320$, and SMSA with $S = 13$, for both LP and NLP.

R_{c}	1/2	2/3	5/6	16/18	20/21
$K = 960$	0.649	0.805	0.931	0.975	0.996
$K = 4320$	0.596	0.759	0.893	0.939	0.982

Table 5.11: Required $\mathrm{SNR}_{\mathrm{avg}}$ to achieve the target BER of 10^{-7} for the LDPC codes with rate 20/21 and $K = 960$, with SMSA ($S = 13$), and LP and NLP. For following cases: absence of IN ($\kappa = -\infty$ dB), IN with $\kappa = 10$ dB and $\kappa = 20$ dB.

	LP			NLP		
b	$\kappa = -\infty$	$\kappa = 10$	$\kappa = 20$	$\kappa = -\infty$	$\kappa = 10$	$\kappa = 20$
1	5.4	14.5	19.5	6.7	15.9	20.9
2	8.4	17.5	22.5	9.1	18.3	23.3
3	12.5	21.7	26.7	13.1	22.3	27.3
4	15.1	24.3	29.3	15.4	24.5	29.5
5	18.3	27.4	32.4	18.3	27.4	32.4
6	21.2	30.3	35.3	21.1	30.3	35.3
7	24.2	33.3	38.3	24.2	33.4	38.3
8	27.0	36.1	41.1	27.0	36.2	41.2
9	29.9	39.0	44.0	29.9	39.0	44.0
10	32.8	41.9	46.9	32.8	41.9	46.9
11	35.6	44.7	49.7	35.6	44.7	49.7
12	38.6	47.7	52.7	38.6	47.7	52.7

nel. At the receiver, the de-interleaver applies the inverse permutation to the parts of the codewords to restore their original order, and the de-interleaver output is applied to the FEC decoder. Because of the interleaving/de-interleaving operation, a single noise impulse affects multiple codewords (instead of a single codeword), but in each codeword only a small number of parts is hit by IN, so that the FEC decoder might be able to correctly decode most of the codewords.

We will point out that the arrangement from Figure 5.33 does not bring much advantage when the interleaving is applied to TCM. Therefore, in the case of TCM, we will use the concatenated code arrangement from Figure 5.34 instead, where the TCM inner code and the RS outer code are separated by an interleaver. In this arrangement, a byte interleaver is applied to the RS codewords, and the bits at the interleaver output are treated as information bits by the TCM encoder. Because of the error correcting capability of the RS decoder, the arrangement from Figure 5.34 also improves the error performance when IN is absent, compared to using TCM only.

Instead of using the frequency division duplexing (FDD) of the downstream and the upstream as adopted in VDSL2, the upstream and downstream communication in G.fast are separated by means of time division duplexing (TDD). In FDD, different frequency bands are used for upstream and downstream transmission, whereas in TDD different time slots are used for upstream and downstream transmission, as illustrated in Figure 5.35. The ratio of time assigned to

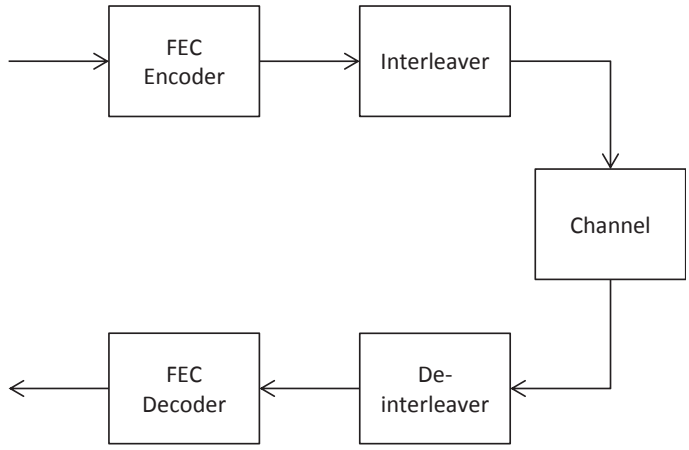

Figure 5.33: Block diagram for the system with FEC and interleaving.

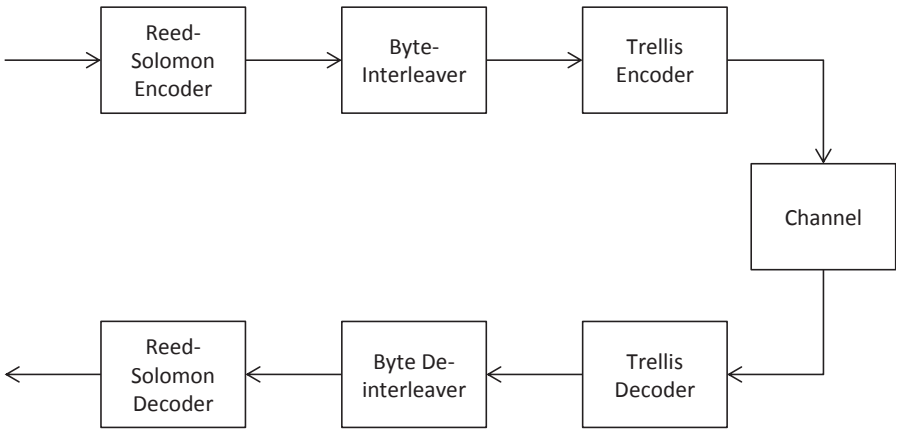

Figure 5.34: Block diagram for the system with RS outer coding and TCM inner coding with byte-interleaver in between.

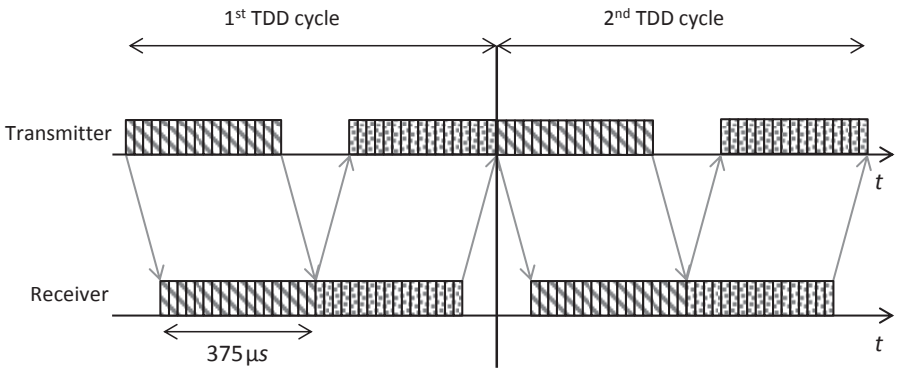

Figure 5.35: TDD cycle with 50% upstream and 50% downstream traffic.

the upstream and the downstream in G.fast is flexible, any ratio between 90% down/10% up and 30% down/70% up can be configured [53]. In our study, the TDD frame (including upstream and downstream DMT symbols) has a duration $T_{\text{TDD}} = 750$ μs and an equal ratio (50%) of the TDD frame is assigned to upstream and downstream traffic. With a DMT symbol rate of 48000 symbols/s, 36 DMT symbols can be transmitted during a TDD frame of 750 μs. One DMT symbol period is reserved as gap [42] and not used to transmit a DMT symbol. 18 DMTs are transmitted during a downstream time slot of 375 μs followed by 17 DMTs which are transmitted during the next upstream time slot.

A major drawback of interleaving is that it adds latency, as the receiver has to wait until all interleaved symbols are received before deinterleaving and further processing can start. When interleaving occurs over J downstream slots, the latency amounts to J downstream slots plus $J-1$ upstream slots. In order to avoid the contribution of upstream slots to the latency, the interleaving should be limited to a single downstream slot.

Let us assume that as a result of interleaving, an interleaved codeword contains bits from L consecutive DMT symbols, with time indices $l, l+1, \ldots, l+L-1$. A DMT symbol with time index l is characterized by a state s_l which indicates whether the DMT is hit ($s_l = 1$) or not hit ($s_l = 0$) by IN with statistics as described in Section 4.5. Representing the conditional BER and WER after decoding the considered codeword as $\text{BER}(\mathbf{s}_l)$ and $\text{WER}(\mathbf{s}_l)$ with $\mathbf{s}_l = (s_l, s_{l+1}, \ldots, s_{l+L-1})$, the average BER and WER after decoding are given by $\text{BER}_{\text{avg}} = \mathbb{E}\left[\text{BER}(\mathbf{s}_l)\right]$ and $\text{WER}_{\text{avg}} = \mathbb{E}\left[\text{WER}(\mathbf{s}_l)\right]$, where the expection is over the state vector $\mathbf{s}_l$; the probability of $\mathbf{s}_l$ follows easily from the 2-state Markov IN model.

5.5.1 TCM + tone-interleaving

We consider the case where the QAM symbols of a TCM codeword are interleaved over L DMT symbols, and LP is used. Let us assume that, as a result of the interleaving/de-interleaving operation, n of the N_t symbols of the codeword originate from DMT symbols which are hit by IN, while the remaining $N_t - n$ symbols originate from DMT symbols which are not hit by IN. The most likely decoding errors involve codewords at minimum distance $2d$ from the correct codeword, which differ from the correct codeword only at QAM symbol positions which are hit by IN. In Section 5.2.2 we have shown that these codewords at distance $2d$ differ from the correct codeword in only one trellis section (i.e., they represent parallel transitions). As it is more likely that only one QAM symbol from a trellis section is hit by IN, we will ignore the probability that both QAM symbols from a same trellis section are affected by IN. Considering decoding errors involving only one QAM symbol error, it follows from Section 5.2.2 that there exist 4 parallel transitions at distance $2d$ for a given QAM symbol position. Assuming that on average half of the $\log_2(M) - 1/2$ information bits associated with the erroneous QAM symbol are in error, the resulting approximation of the average BER is

$$\begin{aligned}\mathrm{BER}_{\mathrm{avg}} &\approx \frac{2\mathbb{E}[n]}{N_t} Q\left(\sqrt{2d^2 \mathrm{SNR}_{\mathrm{avg}} \frac{1+p_1\kappa}{1+\kappa}}\right) \\ &= 2p_1 Q\left(\sqrt{2d^2 \mathrm{SNR}_{\mathrm{avg}} \frac{1+p_1\kappa}{1+\kappa}}\right) \end{aligned} \tag{5.35}$$

where we have taken into account that $\mathbb{E}[n] = p_1 N_t$.

When no interleaving is used, the BER follows from (5.33). Keeping only the second term, we obtain

$$\mathrm{BER}_{\mathrm{avg}} \approx 12 p_1 Q\left(\sqrt{2d^2 \mathrm{SNR}_{\mathrm{avg}} \frac{1+p_1\kappa}{1+\kappa}}\right) \tag{5.36}$$

Comparing (5.35) and (5.36), we conclude that tone-interleaving of the TCM codeword reduces the BER only by a factor of about 6, compared to no interleaving. This rather poor performance in the presence of interleaving is due to the existence of codewords at minimum distance involving only one QAM symbol error.

For tone-interleaved TCM with NLP, similar conclusions can be drawn: the use of interleaving does not improve the argument of the function $Q(.)$ in the BER expression, but only somewhat reduces the factor in front of the function $Q(.)$.

5.5.2 RS + byte-interleaver + TCM

Considering the very limited error performance gain achieved by applying tone-interleaving to the TCM codewords, DSL transmission typically makes use of the arrangement from Figure 5.34, involving the concatenation of a RS outer code and TCM as inner code, with byte-interleaving (transmitter side) and byte-de-interleaving (receiver side) in between the two encoders and decoders, respectively. At the receiver, first Viterbi decoding of the TCM is performed, next the resulting bytes are de-interleaved and supplied to the RS decoder for possible correction of remaining errors. The byte interleaver is introduced to counteract the burstiness of byte errors at the TCM decoder output. Section 5.5.2.1 presents the error performance analysis of this system and numerical results are presented in Section 5.5.2.2.

5.5.2.1 Performance Analysis

The RS code used in G.fast [42] is defined over the Galois field GF(2^8), where a RS code symbol consists of 1 byte ($q = 8$ bits). Per block of K information bytes, an additional $N - K$ parity bytes are transmitted. The value of N can vary from 32 to 255, and $N - K$ must be one of the values from the set $\{2, 4, 6, 8, 10, 12, 14, 16\}$. This RS($N$,$K$) code is systematic and can correct up to

$$t = \left\lfloor \frac{N-K}{2} \right\rfloor \tag{5.37}$$

byte errors.

Taking into account the considerations about the interleaver depth in Section 5.5, we briefly discuss the case where RS codewords are interleaved over 18 DMT symbols (i.e., within one downstream slot of 375 μs). Assuming the use of the traditional RS(255,239) code (which can correct up to 8 byte errors) and distributing the bytes of the RS codewords evenly over the 18 DMT symbols, each DMT contains either $\left\lfloor \frac{255}{18} \right\rfloor = 14$ or $\left\lfloor \frac{255}{18} \right\rfloor + 1 = 15$ bytes from a given RS codeword. When only one of the 18 DMT symbols is hit by IN, then in each of the RS codewords contained within the downstream slot there are 14 or 15 bytes hit by IN. When the IN is strong, these bytes are likely to be erroneous. As the RS code can correct only up to 6 byte errors, none of these RS codewords can be correctly decoded. Hence, interleaving over only 18 DMT symbols does not give much protecion against IN. As a matter of fact, the situation with interleaving is even worse than without interleaving, because in the former case the number of undecodable RS codewords is 18 times as large as in the latter (assuming only one of the 18 DMT symbols is hit by IN). From this brief analysis it follows that interleaving within a single downstream slot is not sufficient to provide adequate protection against IN.

As explained in the beginning of Section 5.5, interleaving over multiple downstream slots would considerably increase the latency for each codeword.

Therefore, in this dissertation we will instead focus on a different technique to counteract IN, i.e., the use of retransmission protocols (see Section 5.6). As far as interleaving (without ARQ) is concerned, we will restrict our attention to the traditional concatenation of a RS outer code and TCM inner code; we will consider the limiting case of infinite interleaving, which provides a lower bound on the error probability, to which the actual error probability will converge when the interleaving depth increases (at the expense of growing latency).

The input of the RS decoder is the byte de-interleaved output of the TCM Viterbi decoder. We assume an infinitely large interleaver, such that the byte errors in the RS codeword can be considered to occur independently with a byte error probability $P_{\text{byte,avg}}$, where $P_{\text{byte,avg}}$ denotes the average byte error rate at the TCM decoder output. Similar to (5.33), $P_{\text{byte,avg}}$ is the average of the byte error rates corresponding to DMT symbol intervals which are hit and not hit by IN, respectively. When all tones of the TCM codeword carry the same constellation and have the same SNR, an approximation of the byte error rate at the TCM decoder output in the absence of IN can be obtained in a similar way as the BER approximation (5.16); we obtain

$$\text{ByteER} \approx 24Q\left(\sqrt{2d^2\text{SNR}}\right) \tag{5.38}$$

where the factor in front of $Q(.)$ results from the simplifying assumption that (i) a trellis section contains an integer number of information bytes; and (ii) all bytes associated with an erroneous trellis section are in error. In the presence of IN, we must replace in (5.38) SNR by $\text{SNR}_{\text{avg}}\frac{1+p_1\kappa}{1+\kappa}$ or $\text{SNR}_{\text{avg}}(1+p_1\kappa)$, when the considered DMT symbol is hit by IN or not hit, respectively.

The error performance of the RS decoder solely depends on the RS code parameters t and N, and on the average byte error probability $P_{\text{byte,avg}}$ after TCM Viterbi decoding. The corresponding RS decoding error probability $P_{\text{e,RS}}$ and the byte error rate $\text{ByteER}_{\text{RS}}$ at the RS decoder output are given by (2.11) and (2.12). The BER after RS decoding is then approximated as

$$\text{BER}_{\text{RS}} \approx \frac{1}{2}\text{ByteER}_{\text{RS}} \tag{5.39}$$

assuming that on average half the bits of an erroneous byte are in error.

5.5.2.2 Numerical Results

We consider two RS codes: the traditional RS(255,239) code and the shortened RS(136,120) code with an information block length equal to the information block length of the shorter LDPC codes (120 bytes = 960 bits). For both RS codes, 16 parity bytes are added which implies that up to $t = 8$ erroneous bytes can be corrected.

First we consider the byte error rate at the TCM decoder output, in the absence of IN. Figure 5.36 shows ByteER for LP, constant SNR and the same

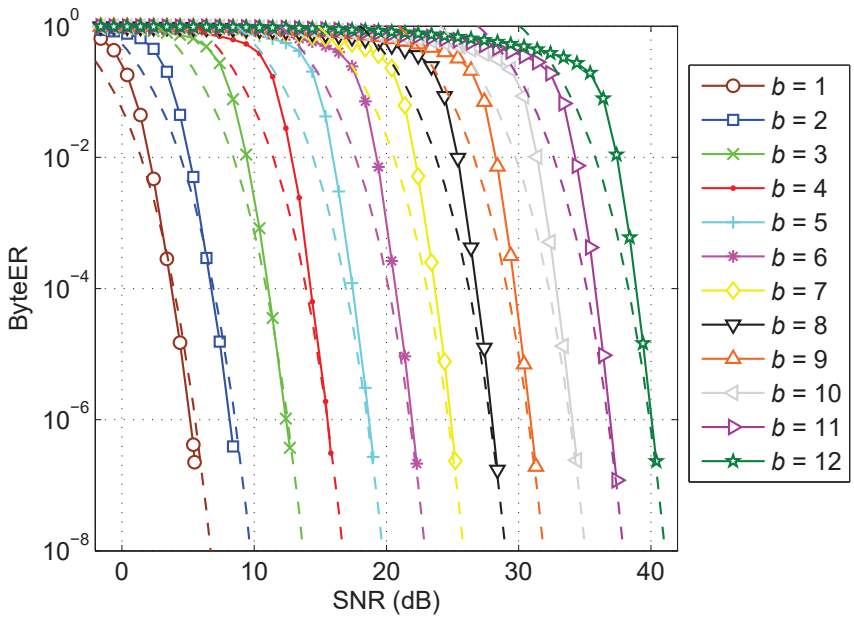

Figure 5.36: Simulated (solid line with markers) and approximate (dashed line) ByteER at TCM decoder output for LP with b-bit QAM constellations ($b = 1, \ldots, 12$).

QAM constellations on all tones with $b = 1, \ldots, 12$. The solid lines and dashed lines correspond to the simulations and the approximation (5.38) respectively. The analytical approximation is less accurate for small constellation sizes, where multiple tone pairs are needed to transmit a single byte of information bits. Therefore, in the sequel we take ByteER equal to the simulation result.

Figure 5.37 compares for 2^6-QAM and LP several error rates versus SNR (same SNR on all tones) in the absence of IN: the BER for uncoded transmission, the BER and ByteER for TCM, and the BER for the concatenation of the RS(255,239) and RS(136,120) outer code and the TCM inner code, with infinite byte-interleaving. The considered coding schemes bring a substantial performance improvement w.r.t. uncoded transmission. Note, however, that the number of information bits per QAM symbol is different for the considered schemes: for uncoded transmission, TCM only, RS(255,239)+TCM and RS(136,120)+TCM, the number of information bits per QAM symbol equals 6, 5.5, $5.5 \cdot 239/255 = 5.15$ and $5.5 \cdot 120/136 = 4.85$, respectively. In order to have 6 information bits per QAM symbol also for the coded systems, their constellation sizes should be increased to (on average) 6.5, 6.9 and 7.3 bits (i.e., info bits+parity bits) for TCM only, RS(255,239)+TCM and RS(136,120)+TCM, in

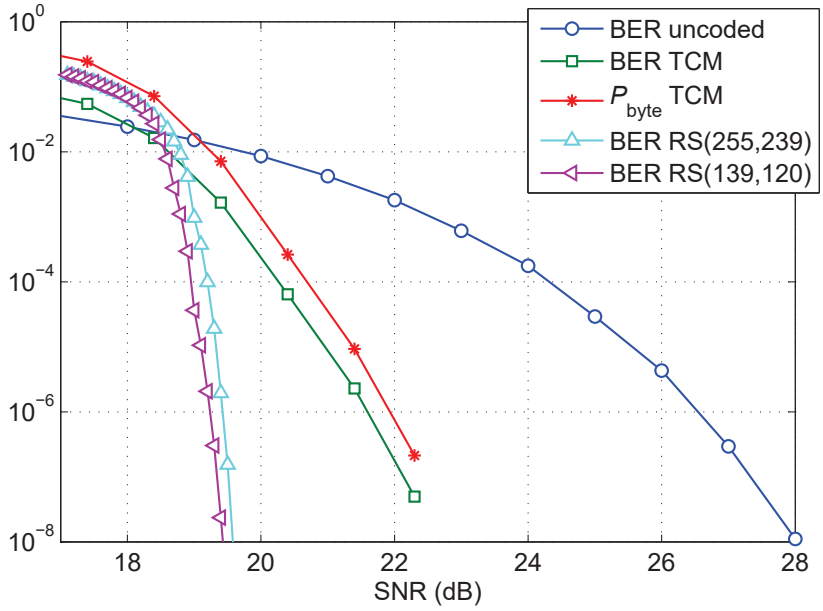

Figure 5.37: Comparison of error performance versus SNR for LP, 2^6-QAM and uncoded transmission, TCM, RS(255,239) and RS(136,120).

which case the coding gains observed from Figure 5.37 should be decreased by about 1.5 dB, 2.7 dB and 3.9 dB, according to the rule of thumb from Section 5.2.4.

In Figure 5.38, the BER performance in the absence of IN is shown for both RS(255,239) + byte-interleaver + TCM and RS(136,120) + byte-interleaver + TCM versus SNR with LP; all tones have the same SNR and the same 2^b-QAM constellation, with $b = 1, \ldots, 12$. For each constellation size the figures shows two curves: the solid line gives the BER for the concatenated code with RS(255,239), the dashed line corresponds to the concatenated code with RS(136,120); only a minor difference (in favor of the shorter RS code) is visible on this scale between each two curves corresponding to the same constellation size. Similar figures can be obtained for RS + byte-interleaver + TCM with NLP, but are not shown here for conciseness.

We have investigated the BER performance of the concatenation of RS and TCM with infinite byte-interleaving, operating on (5.1) for LP and on (5.2) for NLP, assuming that the constellation size and the SNR are the same for all tones. Table 5.12 shows, for 2^b-QAM, the SNR values that are required to achieve BER $= 10^{-7}$ in the absence of IN, for the two considered concatenated codes with RS(255,239) and RS(136,120). Tables 5.13 and 5.14 show similar

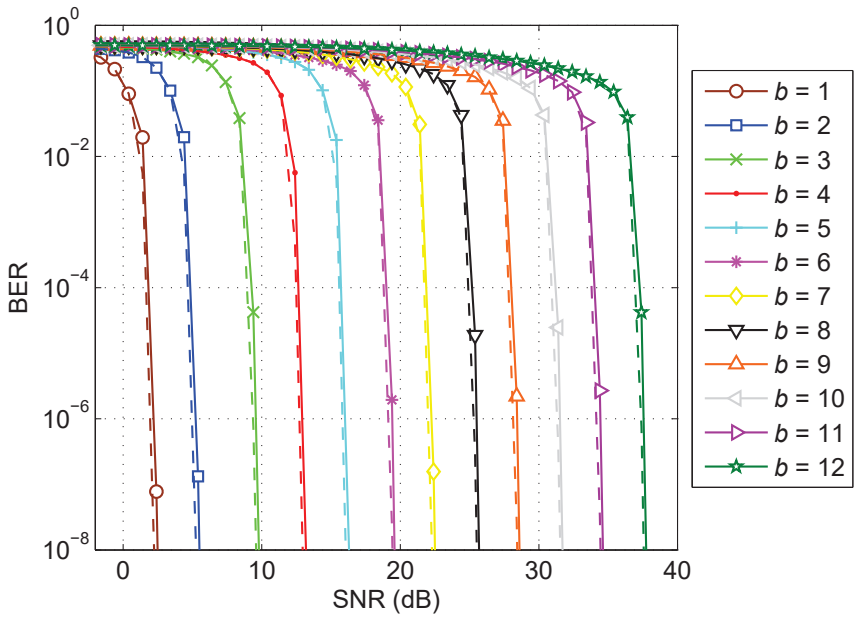

Figure 5.38: BER performance of RS(255,239)+TCM (solid lines with markers) and RS(136,120)+TCM (dashed lines), both with LP and for constant constellation size.

Table 5.12: Required SNR (in dB) to achieve the target BER of 10^{-7} for RS + byte-interleaver + TCM with LP and NLP.

	b	1	2	3	4	5	6
RS(255,239)	LP	2.4	5.4	9.7	13.1	16.2	19.5
	NLP	4.7	6.7	10.7	13.6	16.5	19.8
RS(136,120)	LP	2.2	5.2	9.5	12.9	16.1	19.4
	NLP	4.6	6.6	10.5	13.5	16.3	19.6

	b	7	8	9	10	11	12
RS(255,239)	LP	22.5	25.6	28.6	31.7	34.5	37.7
	NLP	22.6	25.8	28.6	31.8	34.5	37.8
RS(136,120)	LP	22.3	25.5	28.4	31.6	34.4	37.5
	NLP	22.4	25.6	28.4	31.6	34.4	37.6

results pertaining to $\mathrm{SNR}_{\mathrm{avg}}$ for RS + byte-interleaver + TCM, now in the presence of IN with $\kappa = 10$ dB and $\kappa = 20$ dB, respectively. From these tables we observe that the SNR difference between LP and NLP for the same b is larger as compared to the system with only TCM (see Tables 5.8 and 5.9), especially for small b; this can be explained by noticing that in the concatenated arrangement the trellis decoder operates at a much smaller SNR (higher error rate) than in the case where only TCM is used, and that the degradation, in terms of SNR, of the trellis decoder performance for NLP compared to LP increases with increasing error rate. The concatenation involving the shorter RS code slightly outperforms the concatenation involving the longer RS code; both codes have the same error correcting capability, but for the longer code there are more error patterns containing a given number of byte errors. For large constellations, doubling the constellation size (i.e., increasing b by one bit) requires an increase of the SNR by about 3 dB to maintain the BER value at 10^{-7}, which is in agreement with the rule of thumb.

5.6 Use of ARQ against IN

In the presence of IN, our system can benefit from the addition of ARQ, as depicted in the block diagram in Figure 5.39. The receiver is able to request the retransmission of a packet or data transfer unit (DTU) that has been erroneously received by sending a negative acknowledgment (NAK) message to the transmitter. The occurrence of an occasional large noise impulse will typically affect several consecutive DTUs and result in a burst of DTU errors. This can be handled by ARQ; if the cyclic redundancy check (CRC) in the header of the DTU fails, the receiver will ask for a retransmission and the transmitter

Table 5.13: Required $\mathrm{SNR}_{\mathrm{avg}}$ (in dB) to achieve the target BER of 10^{-7} for RS + byte-interleaver + TCM with LP and NLP in the presence of IN with $p_1 = 0.026$, $\kappa = 10$ dB.

	b	1	2	3	4	5	6
RS(255,239)	LP	11.6	14.6	18.9	22.3	25.5	28.8
	NLP	14.0	16.0	19.9	22.9	25.7	29.0
RS(136,120)	LP	11.4	14.4	18.7	22.1	25.3	28.6
	NLP	13.8	15.8	19.8	22.7	25.5	28.8

	b	7	8	9	10	11	12
RS(255,239)	LP	31.7	34.9	37.8	41.0	43.8	46.9
	NLP	31.9	35.0	37.8	41.0	43.8	47.0
RS(136,120)	LP	31.5	34.7	37.5	40.7	43.6	46.7
	NLP	31.6	34.8	37.6	40.8	43.6	46.8

Table 5.14: Required $\mathrm{SNR}_{\mathrm{avg}}$ (in dB) to achieve the target BER of 10^{-7} for RS + byte-interleaver + TCM with LP and NLP in the presence of IN with $p_1 = 0.026$, $\kappa = 20$ dB.

	b	1	2	3	4	5	6
RS(255,239)	LP	16.6	19.6	23.9	27.3	30.5	33.8
	NLP	19.0	21.0	24.9	27.9	30.7	34.0
RS(136,120)	LP	16.3	19.3	23.7	27.1	30.3	33.6
	NLP	18.8	20.8	24.8	27.7	30.5	33.8

	b	7	8	9	10	11	12
RS(255,239)	LP	36.7	39.9	42.8	46.0	48.8	51.9
	NLP	36.9	40.0	42.8	46.0	48.8	52.0
RS(136,120)	LP	36.5	39.7	42.5	45.7	48.6	51.7
	NLP	36.6	39.8	42.6	45.8	48.6	51.7

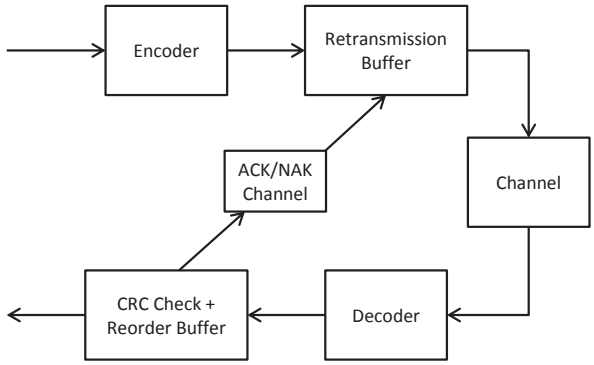

Figure 5.39: General block diagram for the system with ARQ and interleaver.

issues the retransmission of a copy of the specific DTU. A maximum number of retransmissions N_{retr} per DTU is allowed. The use of ARQ implies an increase in latency, which is maximum when the maximum number of retransmissions N_{retr} is necessary to recover a DTU. It is important to note that the latency caused by retransmissions only applies to packets which have not been correctly received during the first transmission; hence, the latency of a packet is a random variable, depending on how many retransmissions are needed to receive the packet correctly. This is in contrast with the case without ARQ, where the latency caused by interleaving (which has to be large in order to be effective against IN) applies to each packet. Subsection 5.6.1 provides a derivation of the resulting latency in the system with TDD and interleaving. The theoretical error performance analysis is presented in Subsection 5.6.2.

5.6.1 Latency Constraint

Here, we want to determine the latency increase caused by ARQ. The use of TDD in G.fast strictly separates in time the downstream and upstream traffic. A retransmission request for an erased DTU in a downstream time slot will be retransmitted at the earliest in the next upstream time slot. We assume that a TDD frame has a duration $T_{\text{TDD}} = 750$ μs [42]. At a DMT symbol rate of 48000 symbols/second, 36 DMT symbols can be transmitted during a TDD frame, where 1 DMT symbol period is reserved as gap and not used for transmission of a DMT symbol. We assume that an equal ratio of time is assigned to upstream and downstream traffic. This implies that in 1 TDD frame 18 DMTs are transmitted during a downstream slot followed by 17 DMTs transmitted during an upstream slot. Assuming interleaving over L consecutive DMTs, the RTT depends on the interleaver size L. We investigate the following two illustrative cases:

$\boldsymbol{L = 9}$ The system with ARQ and interleaving over $L = 9$ DMTs is visualized

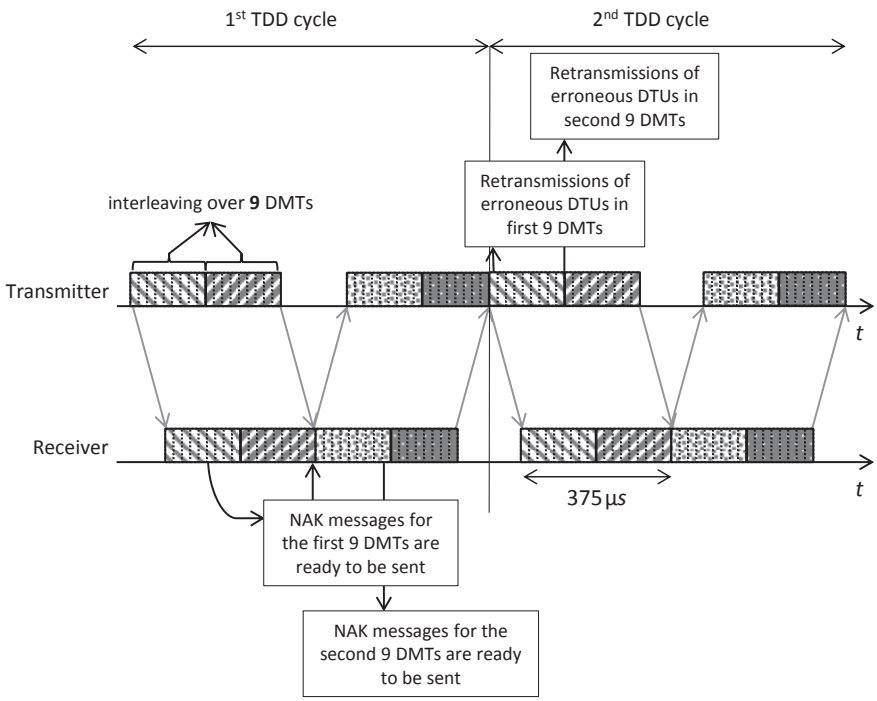

Figure 5.40: Illustration of the system with ARQ and interleaving over $L = 9$ DMTs.

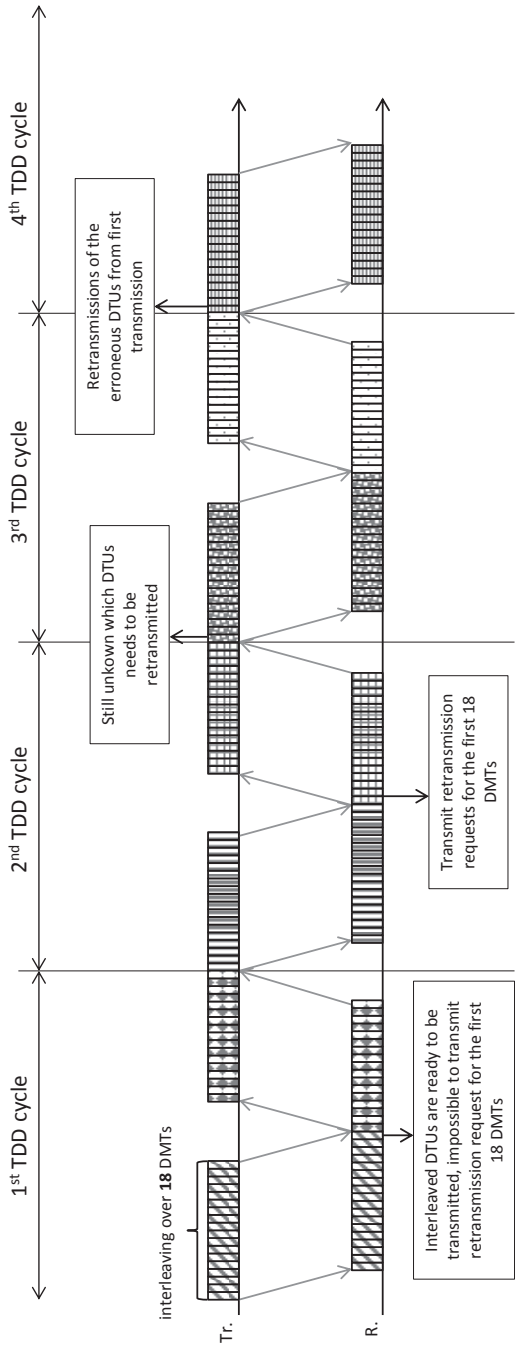

Figure 5.41: Illustration of the system with ARQ and interleaving over $L = 12$ DMTs.

in Figure 5.40. The retransmission request for an erased DTU will be transmitted in the next upstream time slot and the copy of the DTU will be retransmitted in the next downstream time slot. This implies that each retransmitted copie of a DTU adds one TDD cycle to the latency; the RTT here is equal to T_{TDD}. (In the upstream time slot consisting of 17 DMT symbols interleaving is done respectively over 9 and 8 symbols.) Similar results hold for smaller L, involving more than two sets of interleaved DTUs per downstream (or upstream) time slot.

$L = 18$ Here the interleaving occurs over a whole downstream/upstream time slot as shown in Figure 5.41. The retransmission request can not be sent in the next directly following upstream time slot because the receiver needs to perform some post processing of the received DMTs (interleaving and CRC check) before a retransmissions request can be issued. The data to be transmitted in the upstream time slot is also interleaved over 18 DMTs and is ready to be sent in the transmit buffer. The earliest possibility to send the NAK message is in the subsequent upstream time slot. Similarly, the copy of the DTU can not be transmitted in the next downstream time slot but only in the subsequent downstream slot. This implies that each retransmitted copie of a DTU adds 3 TDD cycles to the latency; the RTT here is equal to $3\,T_{\mathrm{TDD}}$. (In the upstream time slot consisting of 17 DMT symbols interleaving is done over all 17 symbols.)

The latency increase caused by adding ARQ is largest when interleaving over $L = 18$ DMT symbols; in this case the maximum number of retransmissions N_{retr} will be 3 times smaller as compared to interleaving over $L = 9$ (or less) DMTs.

As we have argued in Section 5.5.2.1 that interleaving over multiple DMT symbols within a downstream slot does not bring much advantage against IN, we will in this dissertation consider ARQ without interleaving over multiple DMT symbols, and make the comparison with the traditional configuration (without ARQ) consisting of the RS(255,239) outer code and the TCM inner code, with (infinite) interleaving in between.

5.6.2 Theoretical Performance Analysis of ARQ

A typical value for the maximum allowed latency in G.fast [42] is given by $T_{\mathrm{lat}} = 10$ ms. In this time duration, 13 TDD frames can be transmitted for a TDD frame of duration $T_{\mathrm{TDD}} = 750$ μs. The retransmission of a DTU occurs at least 1 TDD frame later than the DMT where it was first transmitted as we have illustrated in Section 5.6.1. If the number of DTUs that needs to be retransmitted is large (due to the occurrence of noise impulses), this number might exceed the capacity of 1 TDD frame, in which case a part of those DTUs will need to be retransmitted in the second next TDD frame. Therefore, the

maximum number of retransmissions $N_{\text{retr,max}}$ is limited to 6, in order not to exceed the maximum latency of $T_{\text{lat}} = 10$ ms.

Furthermore, in Section 4.5 we showed that after about 8 DMT symbol intervals steady-state values are reached for the probabilities $\Pr[s_{l+i}|s_l]$, with s_l denoting the state which indicates wheter the lth DMT symbol is hit by IN. Considering that the RTT exceeds 8 DMT symbols, the original DTU and its retransmitted copies are hit independently by IN. This implies that the probability of an erroneous DTU and the BER for ARQ with N_{retr} retransmissions can be computed by respectively (2.16) and (2.17), where the error probabilities must be understood as the averages of the error probabilities with and without the occurrence of IN, similar to the expression (5.33).

In order to enable the detection of DTU transmission errors, CRC bits are added to each K-bit information word in case of uncoded transmission and TCM; in the case of LDPC coding or RS coding a retransmission request might be issued when the decoder does not provide a valid codeword, in which case no CRC needs to be added. Adding the CRC bits reduces the information bitrate; denoting by R_b' and R_b the information bitrates for a fixed-bandwidth system with and without CRC, we now have

$$R_b' = R_b \cdot \frac{K}{K + K_{\text{CRC}}} \tag{5.40}$$

where K_{CRC} denotes the number of CRC bits that protect the K information bits.

Figure 5.42 shows $\text{BER}^{(N_{\text{retr}})}$, the BER after N_{retr} retransmissions, for $N_{\text{retr}} = 0, 1, \ldots, 4$, in the case of uncoded transmission over an AWGN channel. Also, the probability of an erroneous DTU, $P_{\text{e,DTU}}$ is shown. A DTU consists of 960 bits and is considered to be erroneous if at least one bit in the DTU is erroneous. We observe that the BER curve drops faster when N_{retr} increases, which illustrates the beneficial effect of ARQ on the error performance.

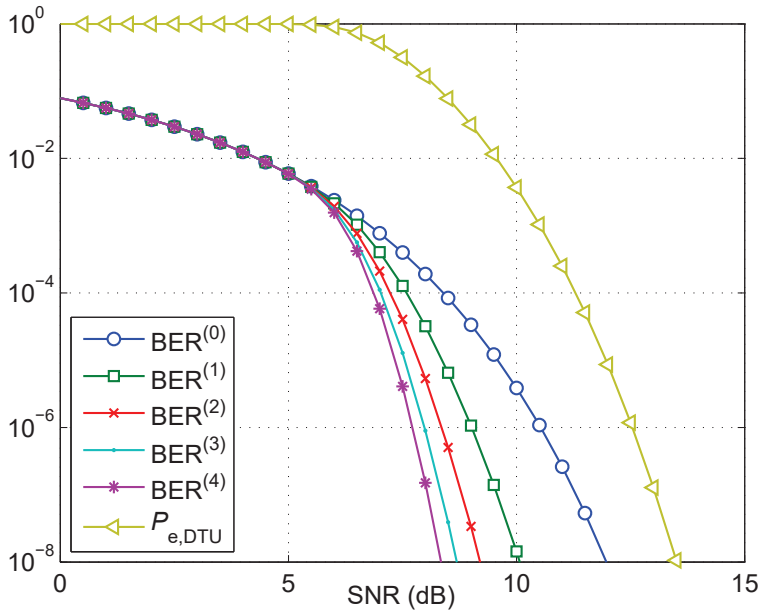

Figure 5.42: BER$^{(N_{\mathrm{retr}})}$ versus SNR for ARQ with N_{retr} retransmissions in the case of uncoded transmission over AWGN channel.

6 Bitloading Algorithms

A bitloading algorithm determines which constellation is to be used on a tone, depending on the channel conditions and on some restrictions (power, power spectral density) on the transmit signal. In this chapter, limits on the bitloading are presented in Section 6.1, which are based on the concepts of mutual information and capacity from the field of information theory. Next, we present two bitloading algorithms for LP and NLP which achieve a given target BER under simultaneous transmit power and transmit PSD constraints. In Section 6.4, we propose a "greedy" algorithm that can be viewed as an extension of the Zanatta-Filho algorithm from [54]. Section 6.5 gives an adjusted version of the column norm scaling algorithm from [55], such that the resulting bitloading table is in compliance with our ATP and PSD constraints. The complexity of both bitloading algorithms is discussed in Section 6.6.

6.1 Shannon's Channel Coding Theorem

The observations at the receiver on tone k and for user i for LP and NLP are respectively given by (4.23) and (4.31). Dropping the indices i and k for conciseness, we set $a' = a_i^{(k)}$ for LP and $a' = a_i^{(k)} + l_i^{(k)} A_i^{(k)}$ for NLP. The

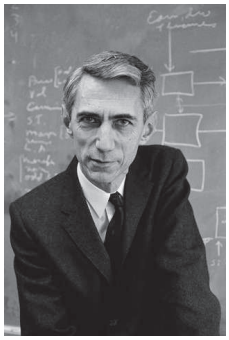

Figure 6.1: Claude Shannon (April 30, 1916, Massachusetts – February 24, 2001).

observation becomes

$$z = \sqrt{\mathrm{SNR}}\, a' + w \tag{6.1}$$

with SNR given by (4.24) and w is zero-mean complex-valued Gaussian noise with $\mathbb{E}[|w|^2] = 1$. Consider a and z as the realizations of two random variables A and Z.

As stated by Shannon in the noisy channel coding theorem [8], the maximum number of information bits per channel use that can be transmitted with arbitrary small decoding error probability is given by the mutual information $I(A;Z)$ between the transmitted symbol A and the observation Z, defined as

$$\begin{aligned} I\left(A;Z\right) &= \int p_{A,Z}(a,z)\log_2\left(\frac{p_{A,Z}(a,z)}{p_A(a)p_Z(z)}\right)\mathrm{d}a\mathrm{d}z \\ &= \int p_{Z|A}(z|a)p_A(a)\log_2\left(\frac{p_{Z|A}(z|a)}{p_Z(z)}\right)\mathrm{d}a\mathrm{d}z \end{aligned} \tag{6.2}$$

The pdf $p_Z(z)$ of the channel output z is computed as

$$p_Z(z) = \int p_{Z|A}(z|a)p_A(a)\mathrm{d}a \tag{6.3}$$

For a given channel transition pdf $p_{Z|A}(z|a)$, the mutual information $I(A;Z)$ depends only on the a priori pdf $p_A(a)$ of the constellation symbol a. The noisy channel coding theorem states that codes exist for which an arbitrarily small error rate can be obtained if R_{info}, the number of information bits per channel use, satisfies $R_{\mathrm{info}} < I(A;Z)$. If $R_{\mathrm{info}} > I(A;Z)$, it is not possible for any code to achieve an arbitrarily low bit error rate.

In the case of LP we have $a' = a$, in which case the channel transition pdf $p_{Z|A}(z|a)$ resulting from (6.1) is given by

$$p_{Z|A}(z|a) = \frac{1}{\pi}\exp\left(-|z - \sqrt{\mathrm{SNR}}\, a|^2\right) \tag{6.4}$$

because $\mathbb{E}[(\Re[w])^2] = \mathbb{E}[(\Im[w])^2] = 1/2$. To obtain the corresponding channel capacity C [20], the mutual information $I(A', Z)$ is maximized over the a priori input distribution $p_A(a)$. For a channel transition distribution pdf given by (6.4), the input pdf $p_{A'}(a')$ that maximizes $I(A', Z)$ under the restriction $\mathbb{E}[|a'|^2] = 1$ is the distribution of a zero-mean Gaussian complex-valued random variable, i.e.,

$$p_A(a) = \frac{1}{\pi} \exp\left(|a|^2\right) \tag{6.5}$$

The resulting capacity (expressed in information bits per channel use) is given by

$$C = \log_2 (1 + \mathrm{SNR}) \tag{6.6}$$

and serves as upper bound on the maximum number of information bits that can be transmitted per channel use for a certain SNR.

Figure 6.2 shows the capacity (6.6), along with the mutual information (6.2) for the 2^b-QAM ($b = 1, \ldots, 12$) constellations described in Section 4.2, assuming LP. Obviously, the capacity curve is above any of the mutual information curves. The curves for 2^b-QAM converge for increasing SNR to a mutual information equal to b.

A similar reasoning can be applied in the case of NLP (i.e, $a' = a + lA$), for which $u = [z]_{A\sqrt{\mathrm{SNR}}}$ is a sufficient statistic. Hence, $I(A; Z) = I(A; U)$, where $I(A; U)$ is obtained by replacing in (6.2) $p_{Z|A}(z|a)$ by $p_{U|A}(u|a)$, with

$$p_{U|A}(u|a) = \begin{cases} \frac{1}{\pi} \sum_{l=-\infty}^{+\infty} \exp\left(-|u - \sqrt{\mathrm{SNR}}\,(a + lA)|^2\right) & |u| < \sqrt{\mathrm{SNR}}a \\ 0 & \text{else} \end{cases} \tag{6.7}$$

The results for the mutual information $I(A; Z)$ versus SNR and for the capacity C versus SNR will differ from those for LP, but are qualitatively similar.

In Section 6.1.1, we will apply information-theoretic bounds on the number of information bits per tone, for a given SNR.

6.1.1 Information-Theoretic Bounds on Bitloading for Given SNR

Let us consider DMT transmission over DSL, where $\mathrm{SNR}^{(k)}$ is the SNR value on the kth tone. We denote by $I_{2^b-\mathrm{QAM}}(\mathrm{SNR})$ the mutual information $I(A; Z)$ related to the 2^b-QAM constellation, considered as a function of SNR. As Figure 6.2 indicates that for given SNR the mutual information $I_{2^b-\mathrm{QAM}}(\mathrm{SNR})$ increases with b, the highest information bitrate for which decoding with arbitrarily small error probability is possible, is obtained by selecting on each tone the largest allowed constellation (i.e., $b^{(k)} = b_{\max}$), and using on the kth tone (with $k = 1, ..., N_t$) a (very powerful) code with rate $R_{\mathrm{c}}^{(k)} = I_{2^{b_{\max}}-\mathrm{QAM}}(\mathrm{SNR}^{(k)})/b_{\max}$. As a result, the information bitrate on each tone equals its maximum possible value (i.e., $I_{2^{b_{\max}}-\mathrm{QAM}}(\mathrm{SNR}^{(k)})$ for the kth tone).

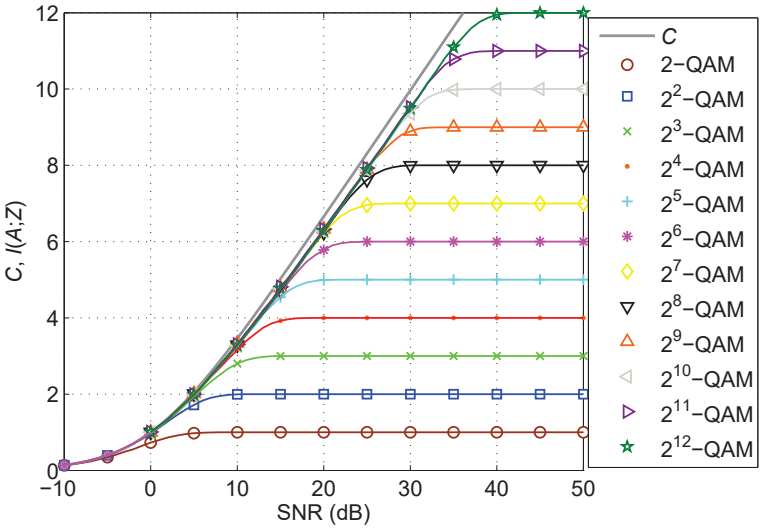

Figure 6.2: Capacity and mutual information for 2^b-QAM ($b = 1, \ldots, 12$) for LP.

Although conceptually simple, the above strategy has practical disadvantages. First, each tone requires its specific code; hence, the constellation symbols from a codeword on the kth tone belong to different DMTs, which introduces a huge latency. Secondly, the code rates to be used are a function of the SNR profile; the SNR profile changes with line length, noise environment,..., and is therefore not known in advance.

Instead, practical DMT transmission over DSL makes use of coding across tones, i.e., the constellation symbols corresponding to a given codeword are on the different tones of a same DMT symbol, rather than on a given tone in different DMT symbols. Therefore, we consider in the information-theoretic analysis the same code rate for all tones. For a code with rate R_{c} and a 2^b-QAM constellation, the number of information bits per channel use equals $R_{\text{info}} = R_{\text{c}} b$. An arbitrarily small error rate can be achieved only when $R_{\text{info}} < I_{2^b-\text{QAM}}(\text{SNR})$. Hence, solving for SNR the equation $R_{\text{c}} b = I_{2^b-\text{QAM}}$ yields the *minimum* SNR for which it is possible to achieve an arbitrarily small error rate with 2^b-QAM and code rate R_{c}. This minimum SNR, which we denote by $\text{SNR}_{\min}(b)$, is the abscissa of the point on the mutual information curve for 2^b-QAM with ordinate $R_{\text{c}} b$. The quantities $\text{SNR}_{\min}(b)$ for $b = 1, ..., b_{\max}$ increase with increasing b. Depending on the actual SNR value, the following cases can be considered.

- When $\text{SNR} < \text{SNR}_{\min}(1)$, no constellation size can give rise to arbitrarily small error rate.

- When $\mathrm{SNR}_{\min}(b) \leq \mathrm{SNR} < \mathrm{SNR}_{\min}(b+1)$ and $b = 1, ..., b_{\max} - 1$, 2^b-QAM is the largest constellation allowing an arbitrarily small error rate; the corresponding information rate is $R_{\mathrm{info}} = R_{\mathrm{c}} b$ bits per channel use.

- When $\mathrm{SNR}_{\min}(b_{\max}) \leq \mathrm{SNR}$, $2^{b_{\max}}$-QAM is the largest constellation allowing an arbitrarily small error rate; the corresponding information rate is $R_{\mathrm{info}} = R_{\mathrm{c}} b_{\max}$ bits per channel use.

As a result, R_{info} changes with SNR according to a *staircase* function, starting at level zero, increasing by R_{c} with each step, and ending at level $R_{\mathrm{c}} b_{\max}$; at $\mathrm{SNR} = \mathrm{SNR}_{\min}(b)$, the level jumps from $R_{\mathrm{c}}(b-1)$ to $R_{\mathrm{c}} b$. Note that the above reasoning applies to coded transmission (i.e., $R_{\mathrm{c}} < 1$) only; for uncoded transmission (i.e., $R_{\mathrm{c}} = 1$) and any constellation, we need an infinitely large SNR in order to achieve an arbitrarily small error probability.

We have determined R_{info} versus SNR for the following code rates:

- $R_{\mathrm{c}} \in \{\frac{1}{2}, \frac{2}{3}, \frac{5}{6}, \frac{16}{18}, \frac{20}{21}\}$, which are the rates of the LDPC codes from Chapter 5.

- $R_{\mathrm{c}} = 1 - \frac{1}{2b}$, $R_{\mathrm{c}} = \frac{239}{255}\left(1 - \frac{1}{2b}\right)$ and $R_{\mathrm{c}} = \frac{120}{136}\left(1 - \frac{1}{2b}\right)$, which are the rates of the TCM from Chapter 5, and of the concatenation of TCM as inner code with RS(255,239) and RS(136,120) as outer code, respectively[1]. Note that for TCM (with 1/2 parity bit per constellation symbol) and its concatenated arrangements the code rates are not constant, but increasing with the constellation size.

In Figures 6.3 and 6.4, we have superposed the curves showing R_{info} versus SNR, for $R_c \in \{\frac{1}{2}, \frac{2}{3}, \frac{5}{6}, \frac{16}{18}, \frac{20}{21}, 1 - \frac{1}{2b}\}$ and $R_c \in \{1 - \frac{1}{2b}, \frac{239}{255}\left(1 - \frac{1}{2b}\right), \frac{120}{136}\left(1 - \frac{1}{2b}\right)\}$, respectively. For $R_c = \gamma\left(1 - \frac{1}{2b}\right)$ (here we consider $\gamma = 1, \frac{239}{255}, \frac{120}{136}$), $\mathrm{SNR}_{\min}(b)$ is the SNR value yielding $\gamma\left(b - \frac{1}{2}\right) = I_{2^b-\mathrm{QAM}}$; for $\mathrm{SNR}_{\min}(b) \leq \mathrm{SNR} < \mathrm{SNR}_{\min}(b+1)$ and $b = 1, ..., b_{\max} - 1$, we have $R_{\mathrm{info}} = \gamma\left(b - \frac{1}{2}\right)$, while for $\mathrm{SNR}_{\min}(b_{\max}) \leq \mathrm{SNR}$ the maximum information rate $R_{\mathrm{info}} = \gamma\left(b_{\max} - \frac{1}{2}\right)$ is achieved. We observe that curves corresponding to different code rates cross each other multiple times, indicating that there is no single code rate R_c which yields the largest R_{info} for all SNR. At low SNR (high SNR) the lower-rate (higher-rate) codes tend to yield the larger R_{info}; the behavior at large SNR is impacted by the restriction imposed on the constellation size (i.e., $b \leq b_{\max}$), causing the information rate R_{info} to reach at high SNR a limit which increases with the code rate.

[1] For $b = 1$, the code rate for TCM equals 3/4 (i.e., the same as for $b = 2$), because of the pairing of two 2-QAM symbols to make a 4-QAM symbol. A similar remark holds for the code rate of concatenated codes involving TCM.

6.2 Practical Bitloading as a Function of SNR

The above information-theoretical considerations provide, for a given code rate, the maximum constellation size (and the corresponding maximum information rate) versus SNR, which allows to achieve an arbitrarily small decoding error probability; however, a vanishingly small error probability requires powerful codes with a length approaching infinity. In a practical setting, *finite*-length codes are used, which for a given code rate require a larger SNR to achieve a small error probability (say, BER = 10^{-7}), compared to the SNR resulting from information theory. In the following, we consider for this practical setting two algorithms for determining the maximum constellation size versus SNR for a given code. The first algorithm (which we will adopt in this dissertation) is based on the relation between the BER and the SNR, obtained in chapter 5 for the considered codes. The second algorithm is a commonly used algorithm based on the "SNR gap".

6.2.1 Bitloading Based on BER versus SNR Curves

Based on the tables from Chapter 5, which show the values of SNR corresponding to BER = 10^{-7} for various codes and constellation sizes, we can obtain for a specific code the relation between the maximum information rate R_{info} and the SNR, under the restriction that the BER does not exceed 10^{-7}. The resulting curve is a staircase function similar to the information-theoretic curve from Figures 6.3 or 6.4 for the corresponding code rate; the jumps now occur at the SNR values from the table from Chapter 5 for the considered code, which are larger than those from the curve derived from information theory.

Figure 6.5 shows R_{info} versus SNR, for the information-theoretic curve related to the code rate $1 - \frac{1}{2b}$, and for TCM with target BER of 10^{-7} for LP and NLP. The degradation of the curves for TCM compared to the information-theoretic curve illustrate the limited error correcting capability of TCM compared to the infinite-length powerful code assumed for the former curve. Similar curves can be obtained for the other practical codes considered in Chapter 5; when superposed, they show multiple crossings, as was the case for the information-theory based curves. Hence, also when using practical codes, there is no code yielding the largest information rate for all SNR.

6.2.2 Bitloading Based on SNR Gap Γ

A commonly used algorithm [56] determines for given SNR the number b of bits in the constellation according to the following closed-form expression

$$b = \log_2 \left(1 + \frac{\text{SNR}}{\Gamma}\right) \tag{6.8}$$

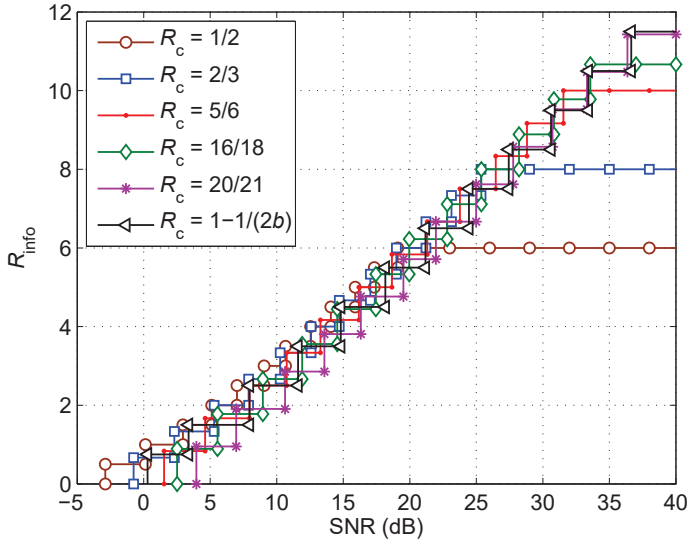

Figure 6.3: R_{info} versus SNR for 2^b-QAM ($b = 1, \ldots, 12$).

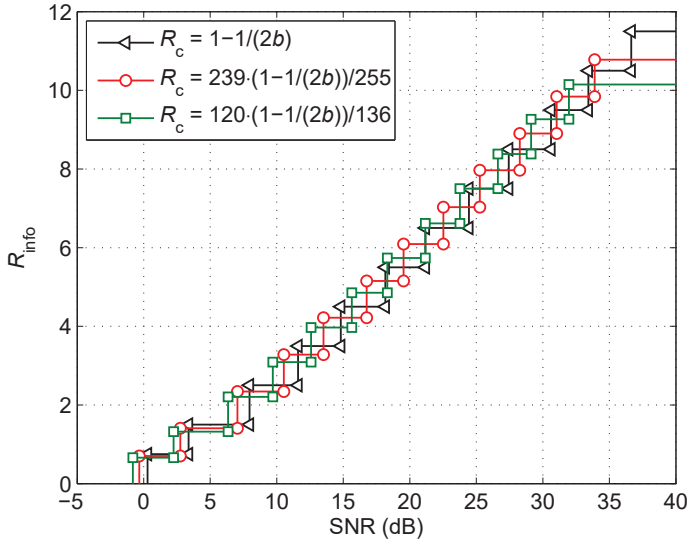

Figure 6.4: R_{info} versus SNR for 2^b-QAM ($b = 1, \ldots, 12$).

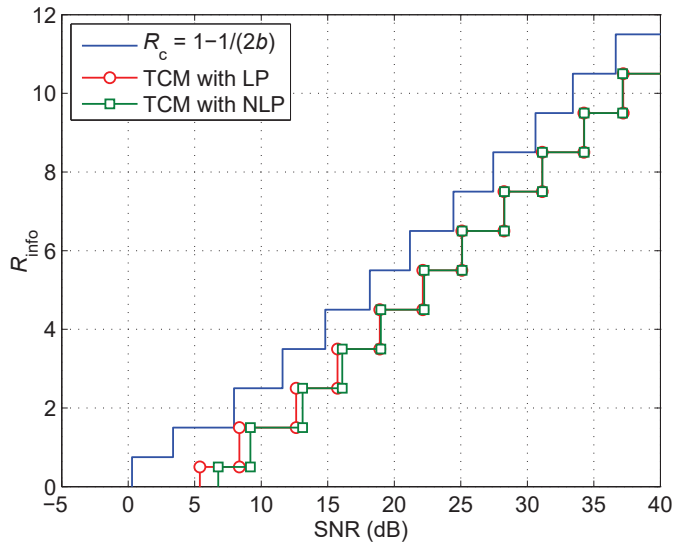

Figure 6.5: R_{info} versus SNR for 2^b-QAM ($b = 1, \ldots, 12$): information-theoretical result for code rate $1 - \frac{1}{2b}$, practical result (BER $= 10^{-7}$) for TCM with LP and NLP.

where Γ is a design parameter referred to as the 'SNR gap'. The corresponding constellation size is given by

$$M = 2^b = 1 + \frac{\text{SNR}}{\Gamma} \tag{6.9}$$

According to (6.8) when SNR doubles (i.e., a 3 dB increase) the number of bits can be increased by 1 (assuming SNR $\gg \Gamma$). Note that the value of b resulting from (6.8) is not necessarily integer; hence, for practical use some kind of rounding to an integer is required.

For $\Gamma = 1$, the right-hand side of (6.8) equals the capacity C of an AWGN channel with given SNR (see (6.6)). For $\Gamma > 1$, the curve showing b versus SNR (in dB) is obtained by shifting to the right by $\Gamma_{\text{dB}} = 10 \log_{10}(\Gamma)$ the curve of C versus SNR (in dB); therefore, Γ is referred to as the *SNR gap* to the capacity curve.

However, the relation of (6.8) with capacity is rather confuse. Suppose we use a very long and powerful code which operates close to capacity. For such code, the right-hand side of (6.8), with $\Gamma = 1$, denotes the information rate (per channel use) R_{info} corresponding to the given SNR, whereas the left-hand side is the number of bits b in the constellation; obviously both sides of the equation refer to different quantities (because $R_{\text{info}} = R_c b$). A similar remark can be made for practical codes operating at an SNR gap Γ from capacity. Moreover,

the capacity formula assumes a Gaussian a priori distribution of the transmitted symbols, which obviously does not correspond to a QAM constellation.

To some extent, the use of (6.8) can be a justified for uncoded transmission and for TCM. Following the discussions in Sections 5.1.1.2, 5.1.2.2, 5.2.2.2 and 5.2.3.2, it follows that in both cases, the BER (for both LP and NLP) can be approximated as

$$\mathrm{BER} \approx f(M) Q\left(\sqrt{\alpha d^2(M) \mathrm{SNR}}\right) \tag{6.10}$$

where $d(M)$ is the minimum distance in the normalized M-QAM constellation, and $f(M)$ is a coefficient depending on the M-QAM constellation, on the precoding (LP or NLP) and on the code (uncoded transmission or TCM); we have $\alpha = 1/2$ for uncoded transmission and $\alpha = 2$ for TCM. Expressions for $d(M)$ have been provided in Section 4.2; for a square M-QAM constellation we have

$$d^2(M) = \frac{6}{M-1} \tag{6.11}$$

Substituting (6.11) into (6.10), and replacing M by (6.9), we obtain BER $\approx f(M) Q\left(\sqrt{6\alpha\Gamma}\right)$, which indicates that the bitloading algorithm (6.8) gives rise to a BER which depends on the constellation size only through the coefficient $f(M)$. It turns out that this coefficient varies only slightly with M; for instance, for uncoded transmission and LP, $f(M)$ decreases monotonically with M, from $f(2^2) = 1$ to $f(2^{12}) \approx 0.33$. Hence, the BER resulting from the bitloading algorithm (6.8) depends mainly on Γ, and is quasi-independent of the constellation size; the value of Γ can be selected to achieve some target BER. However, it should be noted that the target BER is achieved only approximately, not only because of the slight dependence of the coefficient $f(M)$ on the constellation used, but also because (6.11) does not hold for the constellations with an odd number of bits (for instance, for 8-QAM-VDSL and 8-QAM-G.fast, the expression (6.11) with $M = 8$ overestimates the actual minimum distance by about 1 dB and 0.3 dB, respectively).

For LDPC codes, there is no BER approximation similar to (6.10); actually, the BER for LDPC codes is mainly determined by the mutual information I_{avg}, as outlined in Section 5.3.3. Actually, we have pointed out in Section 5.3.7, that for LDPC codes the increase of SNR required to support an additional bit in the constellation is in general not equal to 3 dB (for the rate 1/2 code, the increase is only about 2 dB). Hence, when applying the bitloading algorithm (6.8) in the case of LDPC codes, the BER would still depend on the constellation size. Hence, the SNR gap algorithm is not suited when using LDPC codes. This is illustrated in Figure 6.6, which shows the BER versus Γ for 2^b-QAM, $b = 1, \ldots, 12$ of the rate 1/2 LDPC code, with $K = 4320$ and NLP, with SMSA decoding.

Based on the above considerations, we decide to use the bitloading algorithm from Section 6.2.1, which does not involve any of the approximations and limitations associated with the SNR gap algorithm.

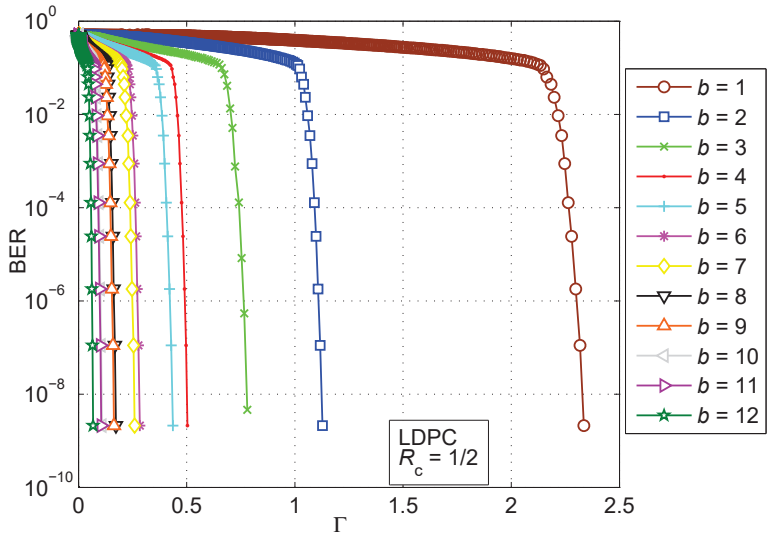

Figure 6.6: BER performance versus Γ of the LDPC code with $R_c = 1/2$, $K = 4320$ and NLP for constant constellation size, with SMSA decoding.

6.3 Bitloading and Energy Allocation

In the case of DMT transmission over the DSL channel, different tones have different SNR values. For a given SNR profile, the algorithm from Section 6.2.1 can be used to select for each tone the constellation yielding the largest information rate under the restriction that the BER does not exceed 10^{-7}; the corresponding maximum total information rate equals the sum of the information rates on the individual tones. Next, the best code from the set of considered codes can be determined, which yields the largest maximum total information rate.

However, as explained in Sections 4.3 and 4.4, for LP and NLP, the SNR on the kth tone associated with the ith user (denoted $\text{SNR}_i^{(k)}$) depends on the corresponding constellation energy $E_i^{(k)}$; the constellation energies in turn determine the transmit energies $\mathbb{E}\left[|x_i^{(k)}|^2\right]$, which are subjected to constraints (spectral density constraint per tone and per user (see (4.10)), and total power constraint per user (see (4.11)). So we are faced with an *energy allocation problem*, i.e., we have to determine the constellation energies $E_i^{(k)}$ such that (i), the constraints on the transmit energies $\mathbb{E}\left[|x_i^{(k)}|^2\right]$ are satisfied; and (ii) the corresponding $\text{SNR}_i^{(k)}$ are such that the maximum total information rate (under the restriction that the BER does not exceed 10^{-7}) is the highest possible for the considered code.

The energy allocation in vectored systems is much more complicated than in a single-user context, because the signal transmitted on a specific tone and line is a function of the constellation symbols of all users on that tone. Moreover, the number of constraints becomes large, especially when a transmit PSD constraint is imposed. For LP, a greedy bitloading algorithm has been introduced in [54] (for a transmit power constraint per line, but no PSD constraint) and an efficient algorithm has been presented in [55] (for a transmit PSD contraint per line, but no transmit power constraint). With regard to NLP, the waterfilling bitloading algorithm from [36] (for a transmit power constraint per line, but no PSD constraint) strongly relies on a diagonally dominant channel matrix, which is a valid assumption for VDSL2 but not for G.fast. In contrast with [36], the proposed algorithms for NLP take account of the effect of the modulo operation on the error performance of trellis-coded modulation. In the following, we will extend the algorithms from [54] and [55], so that they can be applied for both LP and NLP. In order to achieve a target BER for the multicarrier system, we adjust for a given bitloading the power on each tone such that the SNR for each tone equals the SNR for which a single-carrier system with the same constellation yields the target BER. This approach will be validated by computer simulations in Chapters 7 and 8, and can be justified by noting that (i) for uncoded transmission and TCM, the minimal distance between received constellation points is kept (nearly) the same for all tones; and (ii) for LDPC codes, the average mutual information per bit is kept the same for all tones.

6.4 Extended Zanatta-Filho Algorithm

The first algorithm, which will be referred to as EZF, is an extension of the Zanatta-Filho bitloading algorithm [54] which in its original form is limited to LP under only a ATP constraint per line. The EZF is a greedy decremental bitloading algorithm consisting of two parts. In the first part, power and bit allocation is performed for each tone individually, under the PSD mask constraint only. In the second part, it is verified whether the solution from the first part satisfies the ATP constraint; if not, the bitloading is altered in order to also satisfy the ATP contraint. The two parts are detailed in Tables 6.1 and 6.2.

In the first part of EZF, for a given bitloading $\mathbf{b}^{(k)} = (b_1^{(k)}, b_2^{(k)}, ..., b_{N_\mathrm{u}}^{(k)})$ for the kth tone on all lines, the constellation energies $\mathbf{E}^{(k)}$ are determined that yield the corresponding SNRs from the tables in Chapter 5. From these constellation energies, $\mathbb{E}\left[|x_i^{(k)}|^2\right]$ is computed for $i = 1, 2, ..., N_\mathrm{u}$ according to (4.25) for LP or (4.33) for NLP. Assume that the largest $\mathbb{E}\left[|x_i^{(k)}|^2\right]$ occurs for $i = i_{\max}$. If $\mathbb{E}\left[|x_{i_{\max}}^{(k)}|^2\right]$ does not satisfy the PSD constraint, one bit is subtracted from the constellation on line i^* (and $E_{i^*}^{(k)}$ is reduced accordingly to yield the corresponding SNR from the tables in Chapter 5) that provides the

Table 6.1: The EZF bit allocation algorithm.

PART I: Repeat for all tones $k = 1, \ldots, N_{\mathrm{t}}$

1. Initialization: Set $b_i^{(k)} = b_{\max} \quad \forall$ lines $i : 1 \leq i \leq N_{\mathrm{u}}$;
2. While $\exists i : \mathbb{E}\big[|x_i^{(k)}|^2\big] > E_{\mathrm{PSD}}^{(k)}$:
   (a) $i_{\max} = \arg\max_i \big(\mathbb{E}\big[|x_i^{(k)}|^2\big]\big)$
   (b) For $i = 1, \ldots, N_{\mathrm{u}}$: evaluate $\mathbb{E}\big[|\tilde{x}_{i_{\max},i}^{(k)}|^2\big]$, the energy transmitted on line $i_{\max}$ associated with decreasing 1 bit on tone k from line i ;
      i. Let $\tilde{b}_i^{(k)} = b_i^{(k)} - 1$;
      ii. Evaluate the symbol energy $\mathbb{E}\big[|\tilde{u}_i^{(k)}|^2\big]$ on line i associated with $\tilde{b}_i^{(k)}$;
      iii. Evaluate the transmitted energy $\mathbb{E}\big[|\tilde{x}_{i_{\max},i}^{(k)}|^2\big]$ at line $i_{\max}$, required for this new allocation on line i;
   (c) Find i^* such that $i^* = \arg\min_i \mathbb{E}\big[|\tilde{x}_{i_{\max},i}^{(k)}|^2\big]$
   (d) Substract one bit at the optimum position: $b_{i^*}^{(k)} = b_{i^*}^{(k)} - 1$;
   (e) Compute $\mathbb{E}\big[|u_i^{(k)}|^2\big]$ and $\mathbb{E}\big[|x_i^{(k)}|^2\big]$ (for all i) for this new bit allocation;
3. End.

Table 6.2: The EZF bit allocation algorithm.

PART II: While $\exists i : \sum_{k=1}^{N_\mathrm{t}} \mathbb{E}\left[|x_i^{(k)}|^2\right] > E_\mathrm{ATP}$:

1. $i'_\mathrm{max} = \arg\max_i \sum_{k=1}^{N_\mathrm{t}} \left(\mathbb{E}\left[|x_i^{(k)}|^2\right]\right)$
2. For $i = 1, \ldots, N_\mathrm{u}$ and $k = 1, \ldots, N_\mathrm{t}$: evaluate $\mathbb{E}\left[|\tilde{x}_{i'_\mathrm{max},i}^{(k)}|^2\right]$, the energy transmitted on line i'_max associated with decreasing 1 bit on tone k from line i ;
   (a) Let $\tilde{b}_i^{(k)} = b_i^{(k)} - 1$;
   (b) Evaluate the symbol energy $E_i^{(k)}$ on line i associated with $\tilde{b}_i^k$;
   (c) Evaluate the transmitted energy at line i'_max, $\mathbb{E}\left[|\tilde{x}_{i'_\mathrm{max},i}^{(k)}|^2\right]$, required for this new allocation on line i and tone k;
3. Find (i^*, k^*), such that $(i^*, k^*) = \arg\max_{i,k} \mathbb{E}\left[|x_{i'_\mathrm{max}}^{(k)}|^2\right] - \mathbb{E}\left[|\tilde{x}_{i'_\mathrm{max},i}^{(k)}|^2\right]$
4. Subtract one bit at the optimum position: $b_{i^*}^{(k^*)} = b_{i^*}^{(k^*)} - 1$;
5. Compute $E_i^{(k)}$ and $\mathbb{E}\left[|x_i^{(k)}|^2\right]$ (for k^*, all i) for this new bit allocation;

End.

largest reduction of $\mathbb{E}\left[|x_{i_{\max}}^{(k)}|^2\right]$. The algorithm is initialized with the maximum bitloading, and is repeated until all PSD constraints are satisfied. For each one-bit reduction, the selected line is optimum, in the sense that the energy $\mathbb{E}\left[|x_i^{(k)}|^2\right]$ that exceeds E_{ATP} by the largest amount is maximally reduced.

The second part of EZF is initialized with the bitloading that results from the first part. Assume that the largest $\sum_{k=1}^{N_\text{t}} \mathbb{E}\left[|x_i^{(k)}|^2\right]$ occurs for $i = i'_{\max}$. If $\sum_{k=1}^{N_\text{t}} \mathbb{E}\left[|x_{i'_{\max}}^{(k)}|^2\right]$ does not satisfy the ATP constraint, one bit is subtracted from the constellation on line i^* and tone k^* (and $E_{i^*}^{(k^*)}$ is reduced accordingly to yield the corresponding SNR from the tables in Chapter 5) that provides the largest reduction of $\sum_{k=1}^{N_\text{t}} \mathbb{E}\left[|x_{i'_{\max}}^{(k)}|^2\right]$. The algorithm is repeated until all ATP constraints are satisfied. For each one-bit reduction, the selected line and tone are optimum, in the sense that the energy $\sum_{k=1}^{N_\text{t}} \mathbb{E}\left[|x_i^{(k)}|^2\right]$ that exceeds $E_{\text{ATP}}^{(k)}$ by the largest amount is maximally reduced.

6.5 Column Norm Scaling Algorithm

The column norm scaling (CNS) algorithm has originally been presented in the context of LP under a PSD constraint [55]. Here it is extended to a joint PSD and ATP constraint, for both LP and NLP.

In a first part, CNS for LP provides in a simple way symbol energies $E_i^{(k)}$ that satisfy the PSD constraints. As in [55], for given k the algorithm initially takes $E_i^{(k)} = \alpha^{(k)} \left(\sum_{j=1}^{N_\text{u}} |P_{j,i}^{(k)}|^2\right)^{-1}$ where $\alpha^{(k)}$ is the largest factor for which the PSD constraints are satisfied, and computes the corresponding $\text{SNR}_i^{(k)}$. The bitloading $b_i^{(k)}$ is uniquely determined by the inequality $\text{SNR}_{\text{LP}}(b_i^{(k)}) \leq \text{SNR}_i^{(k)} < \text{SNR}_{\text{LP}}(b_i^{(k)} + 1)$, and $E_i^{(k)}$ is reduced such that the corresponding $\text{SNR}_i^{(k)}$ equals $\text{SNR}_{\text{LP}}(b_i^{(k)})$.

In a second part, CNS for LP takes the ATP constraints into account. Assume that the largest $\sum_{k=1}^{N_\text{t}} \mathbb{E}\left[|x_i^{(k)}|^2\right]$ occurs for $i = i'_{\max}$ and the largest $\mathbb{E}\left[|x_{i'_{\max}}^{(k)}|^2\right]$ occurs for $k = k^*$. If $\sum_{k=1}^{N_\text{t}} \mathbb{E}\left[|x_{i'_{\max}}^{(k)}|^2\right]$ does not satisfy the ATP constraint, one bit is subtracted from the constellations on tone k^* on all N_u lines. The algorithm is repeated until all ATP constraints are satisfied.

The extension of CNS to NLP is detailed in Table 6.3 (first part) and Table 6.4 (second part), and follows a same reasoning as for LP. The main difference is that in the first part (related to the PSD constraints) we start with $E_i^{(k)} = E_{\text{PSD}}^{(k)}$ for all lines. Taking into account that $\sum_{j=1}^{N_\text{t}} |Q_{i,j}^{(k)}|^2 = 1$, these initial $E_i^{(k)}$ would result from applying the CNS algorithm (with $\mathbf{P}^{(k)}$ replaced by $\mathbf{Q}^{(k)}$) when assuming $\rho_i^{(k)} = 1$ (see (4.33) and (4.34)); however these ini-

Table 6.3: The CNS bit allocation algorithm for NLP.

PART I: Repeat for all tones $k = 1, \ldots, N_{\mathrm{t}}$ and all lines $i = 1, \ldots, N_{\mathrm{u}}$

1. Initialization: Set $E_i^{(k)} = E_{\mathrm{PSD}}^{(k)}$;
2. Compute $\mathrm{SNR}_i^{(k)}$ from (4.24)
3. Determine the value of m for which:

$$\mathrm{SNR}_{\mathrm{NLP}}(m) < \mathrm{SNR}_i^{(k)} \leq \mathrm{SNR}_{\mathrm{NLP}}(m+1)$$

4. Take $b_i^{(k)} = m$ and $\mathrm{SNR}_i^{(k)} = \mathrm{SNR}_{\mathrm{NLP}}(m)$;
5. Compute $E_i^{(k)}$ and $\mathbb{E}\big[|x'^{(k)}_i|^2\big]$ for this new bit allocation;
6. Check whether $\mathbb{E}\big[|x'^{(k)}_i|^2\big] \leq E_{\mathrm{PSD}}^{(k)}$
   $\longrightarrow$ if not: $b_i^{(k)} = b_i^{(k)} - 1$; update $E_i^{(k)}$

End.
Compute $E\big[|x_i^{(k)}|^2\big]$ (for all i, k) for this new bit allocation.

tial $E_i^{(k)}$ cause $\mathbb{E}\left[|x_i^{(k)}|^2\right]$ to (slightly) exceed the PSD constraints for all i, because $\rho_i^{(k)} \geq 1$. Then, $b_i^{(k)}$ is determined by the inequality $\mathrm{SNR}_{\mathrm{NLP}}(b_i^{(k)}) \leq \mathrm{SNR}_i^{(k)} < \mathrm{SNR}_{\mathrm{NLP}}(b_i^{(k)} + 1)$, with $\mathrm{SNR}_i^{(k)}$ given by (4.24), and $E_i^{(k)}$ is reduced such that $\mathrm{SNR}_i^{(k)} = \mathrm{SNR}_{\mathrm{NLP}}(b_i^{(k)})$. Finally, we verify for each i whether $\mathbb{E}\left[|x_i'^{(k)}|^2\right] = \rho_i^{(k)} E_i^{(k)}$ exceeds $E_{\mathrm{PSD}}^{(k)}$; if yes, we reduce $b_i^{(k)}$ by one bit, and reduce $E_i^{(k)}$ accordingly. This conservative approach yields $\mathbb{E}\left[|x_i'^{(k)}|^2\right] \leq E_{\mathrm{PSD}}^{(k)}$ for all lines; hence all PSD constraints $\mathbb{E}\left[|x_i^{(k)}|^2\right] \leq E_{\mathrm{PSD}}^{(k)}$ are satisfied because the PSD constraints are identical for all lines and $\sum_{j=1}^{N_{\mathrm{t}}} |Q_{i,j}^{(k)}|^2 = 1$. The second part (related to the ATP constraint) is the same as for LP.

Table 6.4: The CNS algorithm bit allocation algorithm for NLP.

PART II: While $\exists i : \sum_{k=1}^{N_\mathrm{t}} \mathbb{E}\big[|x_i^{(k)}|^2\big] > E_\mathrm{ATP}$:

1. $i'_\mathrm{max} = \arg\max_i \sum_{k=1}^{N_\mathrm{t}} \big(\mathbb{E}\big[|x_i^{(k)}|^2\big]\big)$
2. $k_\mathrm{max} = \arg\max_k \mathbb{E}\big[|x_{i'_\mathrm{max}}^{(k)}|^2\big]$
3. For all lines, decrease the bitloading on tone k_max by 1 bit: $b_i^{(k_\mathrm{max})} = b_i^{(k_\mathrm{max})} - 1$ for $i = 1, 2, ..., N_\mathrm{u}$;
4. Compute $E_i^{(k)}$ and $E\big[|x_i^{(k)}|^2\big]$ (for k_max, all i) for this new bit allocation;

End.

6.6 EZF and CNS Complexity

In the first part of EZF, we start from the maximum bitloading, and reduce the bitloading by one bit at a time, until all PSD constraints are met. The first part requires, per tone and per one-bit reduction, a search over N_u transmit energies $\mathbb{E}\left[|x_i^{(k)}|^2\right]$ to determine i_max, followed by the computation of and a search over N_u transmit energies $\mathbb{E}\left[|x_{i_\mathrm{max}}^{(k)}|^2\right]$ to determine i^*. In the second part of EZF, we again reduce the bitloading by one bit at a time, until all ATP constraints are satisfied. The second part requires, per one-bit reduction, a search over N_u transmit energies $\sum_{k=1}^{N_\mathrm{t}} \mathbb{E}\left[|x_i^{(k)}|^2\right]$ to determine i'_max, followed by the computation of and a search over $N_\mathrm{u}N_\mathrm{t}$ transmit energies $\sum_{k=1}^{N_\mathrm{t}} \mathbb{E}\left[|x_{i'_\mathrm{max}}^{(k)}|^2\right]$ to determine (i^*, k^*).

In the first part of CNS, the constellation energies $E_i^{(k)}$ and the corresponding bitloadings $b_i^{(k)}$ yielding transmit energies $\mathbb{E}\left[|x_i^{(k)}|^2\right]$ that satisfy all PSD constraints are computed directly, without requiring one-bit reduction nor any search. The second part of CNS involves reducing by one bit the bitloadings on all lines for a particular tone; this requires, per such reduction, a search over N_u transmit energies $\sum_{k=1}^{N_\mathrm{t}} \mathbb{E}\left[|x_i^{(k)}|^2\right]$ to determine i'_max, followed by a search over N_t transmit energies $\mathbb{E}\left[|x_{i'_\mathrm{max}}^{(k)}|^2\right]$ to determine k_max.

The above considerations indicate that CNS has a much smaller computational complexity, compared to EZF.

7 Information Rates in the Absence of Impulsive Noise

In this chapter, we apply the bit allocation algorithms from Chapter 6 to the DSL communication system described in Chapter 4. We compare the resulting information bitrates, goodput and computation times for the different precoding schemes in a G.fast downlink scenario without impulsive noise. We consider DSL loops of 100 m, 150 m, 200 m and 250 m and apply the different protection techniques that are described in Chapter 2. A binder consists of 8 parallel twisted pair lines; the corresponding size-8x8 channel matrices for each tone have been obtained from measurements. The cable contains foam-skin polyethylene insulation and the copper diameter is 0.6 mm. Each line is assumed to be affected by stationary white noise with PSD given by N_0 = -140 dBm/Hz. The G.fast-based DMT modulation uses frequencies up to 212 MHz, a tone spacing of 51.75 kHz and a DMT symbol rate of 48000 symbols/s. The transmission power P_{ATP} on each line is limited to 4 dBm. The maximum allowed BER is set to 10^{-7}.

In Sections 7.1, 7.2, 7.3, 7.4 and 7.6 we consider respectively uncoded trans-

mission, TCM, LDPC codes, the concatenation of RS outer code + byte-interleaver + TCM inner code, and ARQ.

7.1 Uncoded Transmission

In the case of uncoded transmission, the bit allocation algorithms make use of Table 5.1 which contains the SNR levels for which the target BER of 10^{-7} is achieved for constellation sizes up to 2^{12}-QAM. After applying the bit allocation algorithms on the different DSL loops, following results are obtained. For an arbitrary line ($i = 4$) in the 200 m DSL loop, the PSD mask, the transmit energy $\mathbb{E}[|x_i^{(k)}|^2]$, the constellation energy $E_i^{(k)}$ and the corresponding bitloading are shown as a function of the tone frequency, in Figures 7.1, 7.2, 7.3 and 7.4, for several combinatons of precoding (LP or NLP) and bitloading (CNS of EZF): LP-CNS, NLP-CNS, LP-EZF and NLP-EZF, respectively. The results from these figures can be explained as follows:

- For tones at low frequencies (say, below 40 MHz), crosstalk is negligible and the channel matrix can be considered diagonal. In this case, the transmit energy and the constellation energy are essentially the same. The PSD mask would allow constellation sizes exceeding 12 bit, but the number of bits per constellation symbol is limited to 12: the transmit energy is adjusted such that BER = 10^{-7} for 2^{12}-QAM, and is substantially below the PSD mask.

- For tones in the medium frequency range (say, 40-120 MHz), the channel matrix has significant off-diagonal elements, indicating the presence of strong crosstalk. For LP, the off-diagonal elements have a large impact on the constellation energies resulting from the bitloading, yielding strong fluctuations in constellation energy compared to the PSD mask. For NLP, this impact is much less, because of the modulo operation at the transmitter; in this case the constellation energy is much closer to the PSD mask.

- At high frequencies (say, above 120 MHz), several tones with zero bitloading occur, because of the very large signal attenuation. The corresponding constellation energy $E_i^{(k)}$ is obviously zero, but the transmit energy $\mathbb{E}[|x_i^{(k)}|^2]$ on these tones might be nonzero. This is because the transmitted component $x_i^{(k)}$ on line i depends on the constellation symbol energies on *all* lines, according to the type of precoding used.

Table 7.1 compares the EZF and CNS bitloading algorithms in terms of the resulting average information bitrate per line, for LP and NLP and for the different loop lengths. We observe that NLP outperforms LP by roughly

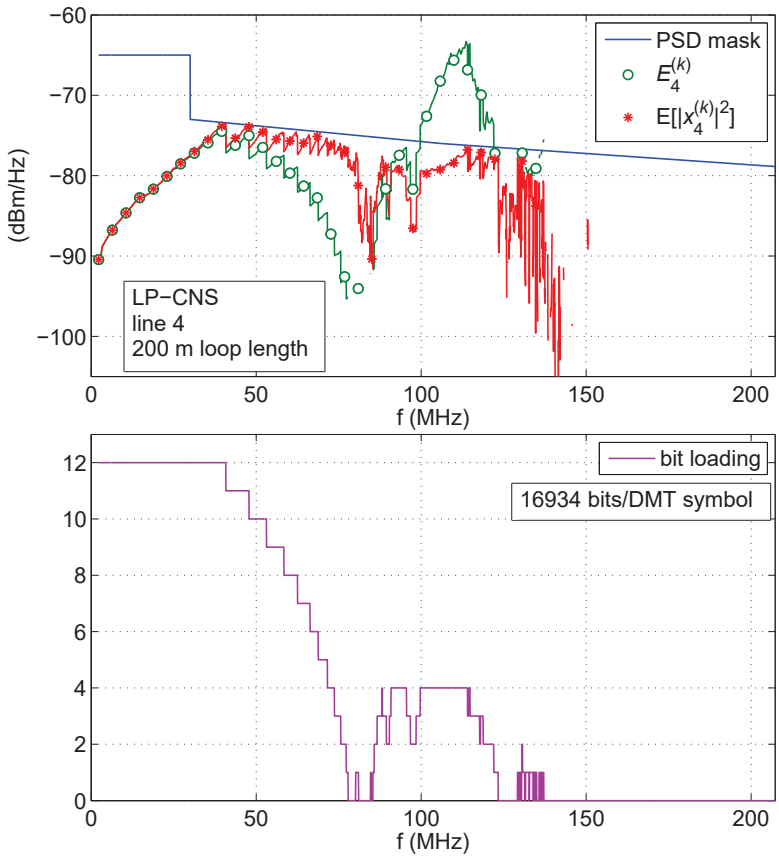

Figure 7.1: PSD values and bitloading for line 4 of the 200 m loop with uncoded transmission and LP-CNS.

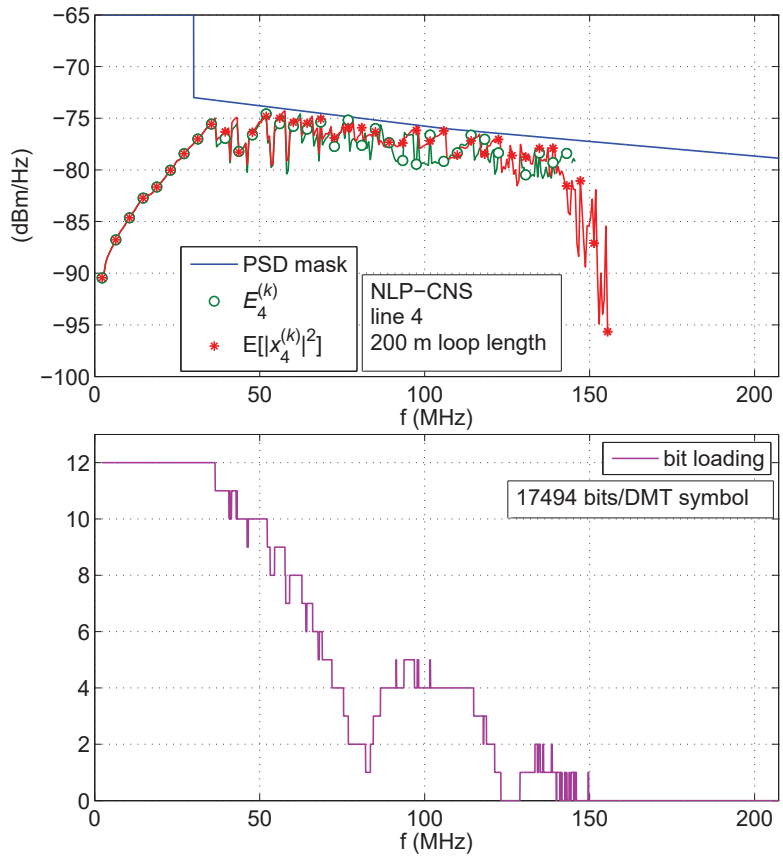

Figure 7.2: PSD values and bitloading for line 4 of the 200 m loop with uncoded transmission and NLP-CNS.

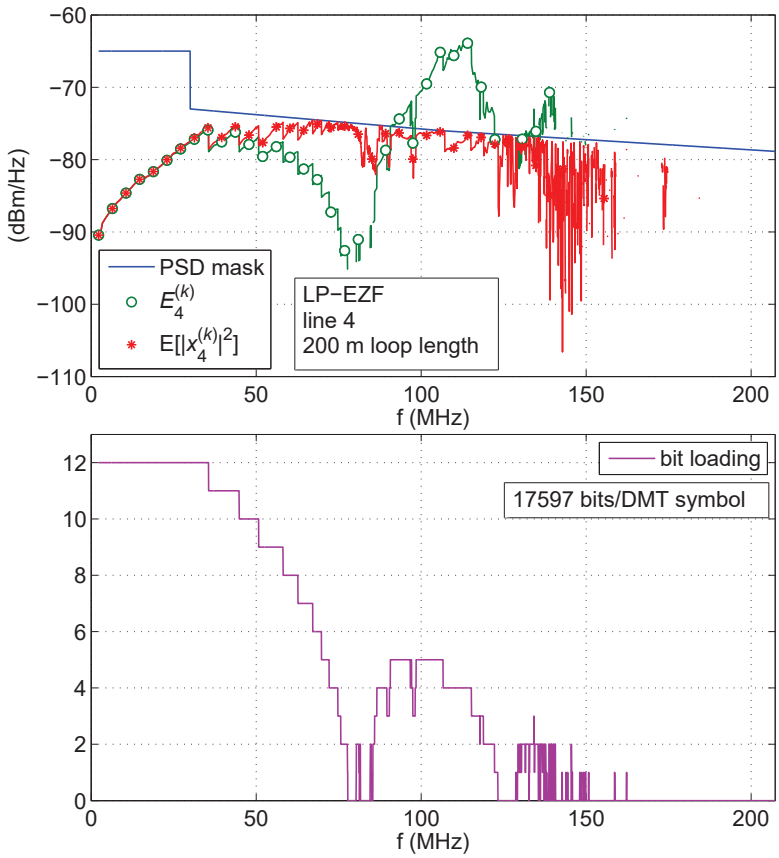

Figure 7.3: PSD values and bitloading for line 4 of the 200 m loop with uncoded transmission and LP-EZF.

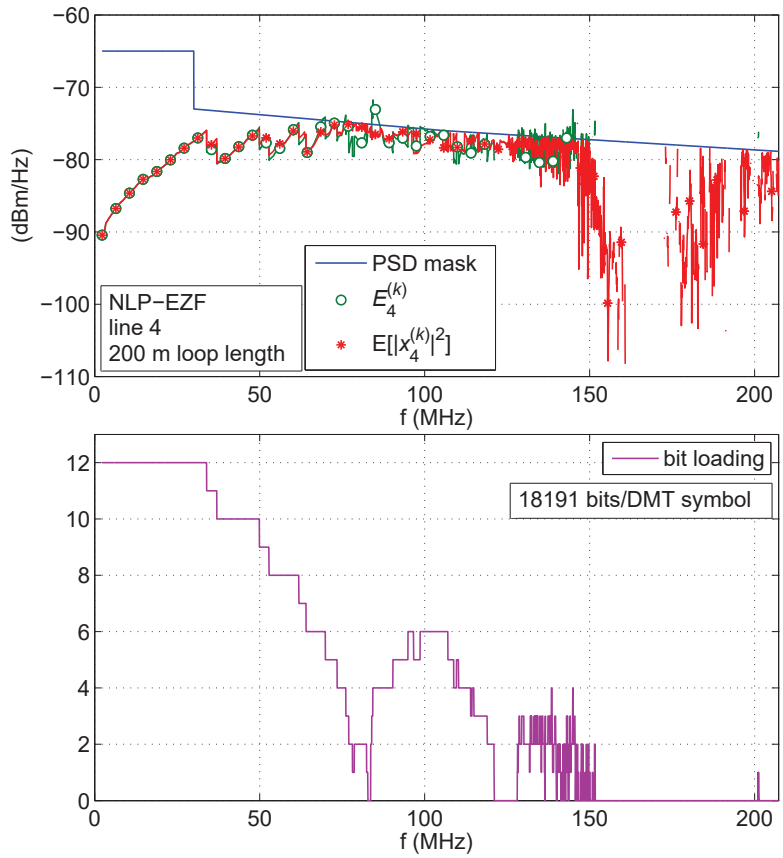

Figure 7.4: PSD values and bitloading for line 4 of the 200 m loop with uncoded transmission and NLP-EZF.

Table 7.1: Average information bitrates (Gbps) per line for uncoded transmission.

	100m	150m	200m	250m
LP-EZF	1.624	1.236	0.842	0.638
LP-CNS	1.575	1.213	0.817	0.615
NLP-EZF	1.764	1.339	0.930	0.695
NLP-CNS	1.741	1.325	0.919	0.688

Table 7.2: Computation time to determine the bitloading for uncoded transmission and all lines on all tones.

	100m	150m	200m	250m
LP-EZF	3.06 s $\otimes$	16.74 s $\otimes$	13.52 s $\otimes$	8.24 s $\otimes$
LP-CNS	0.07 s	0.06 s	0.08 s	0.06 s
NLP-EZF	3.43 s $\otimes$	22.75 s $\otimes$	24.99 s $\otimes$	16.01 s $\otimes$
NLP-CNS	0.16 s	0.21 s $\otimes$	0.19 s $\otimes$	0.16 s

10% in terms of information rate. The differences between EZF and CNS are considerably less than 10%; EZF outperforms CNS by about 3% and 1.1% for LP and NLP, respectively.

Table 7.2 shows the computation times for the various combinations of precoding technique and bitloading algorithm, providing an indication of the relative complexity of the algorithms; the symbol $\otimes$ indicates that part II of the algorithm (related to the constraint on the transmission power per line) had to be executed. We observe that CNS is at least a factor of 20 faster than EZF. Hence, the slightly higher information bitrate resulting from EZF compared to CNS comes at the cost of a considerably higher computational complexity.

7.2 TCM

In Chapter 5, Table 5.2 gives SNR values that are required for transmission with TCM to achieve exactly the target BER equal to 10^{-7}. These SNR values are determined specifically for TCM codewords with a constant constellation size on all tones in the DMT. To show that these table values are also valid for

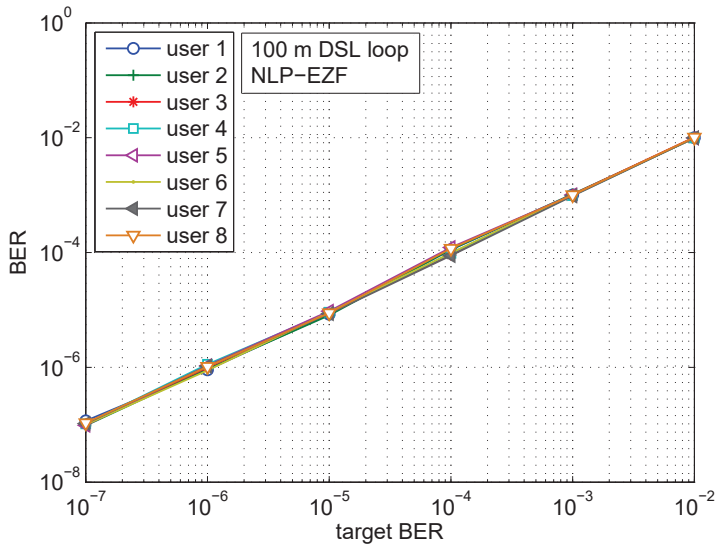

Figure 7.5: Resulting BER versus target BER for TCM and the 100 m DSL loop with 8 users.

TCM codewords with different constellation sizes on different tones in the DMT, simulations are performed on a real DSL loop, where we applied bitloading for target BER values ranging from 10^{-2} to 10^{-7}; the SNR tables for target BER values different from 10^{-7} were obtained in the same way as those for BER $= 10^{-7}$. The actual BER resulting from the bitloading is shown in Figure 7.5 versus the target BER, for the 100 m DSL loop with 8 lines and NLP-EZF. It is clear that there is a very good agreement between resulting BER and target BER.

After applying the bit allocation algorithms from Chapter 6 on different DSL loops, following numerical results are obtained. For an arbitrary line ($i = 4$) in the 200 m DSL loop, the PSD mask, the transmit energy $\mathbb{E}[|x_i^{(k)}|^2]$, the constellation energy $E_i^{(k)}$ and the corresponding bitloading are shown as a function of the tone frequency, in Figures 7.6, 7.7, 7.8 and 7.9, for LP-CNS, NLP-CNS, LP-EZF and NLP-ZF, respectively. A similar behaviour as with uncoded transmission is observed.

Table 7.3 compares the EZF and CNS bitloading algorithms in terms of the resulting average information bitrate per line, for LP and NLP and for the different loop lengths. We observe that NLP outperforms LP by roughly 10% in terms of information rate. The differences between EZF and CNS are considerably less; EZF outperforms CNS by about 3.7% and 1.3% for LP and

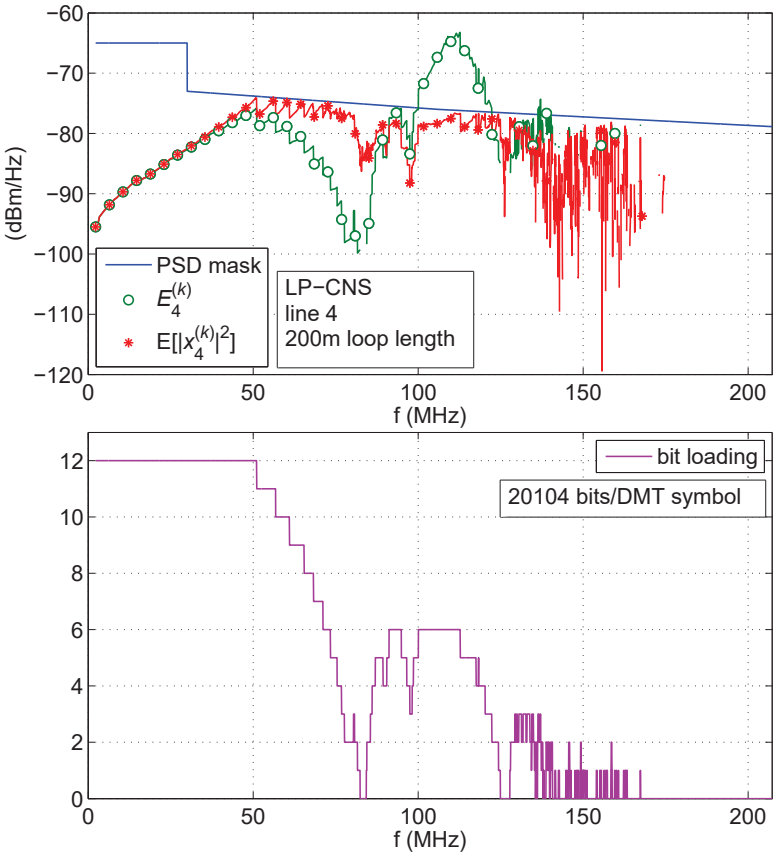

Figure 7.6: PSD values and bitloading for line 4 of the 200 m loop with TCM and LP-CNS.

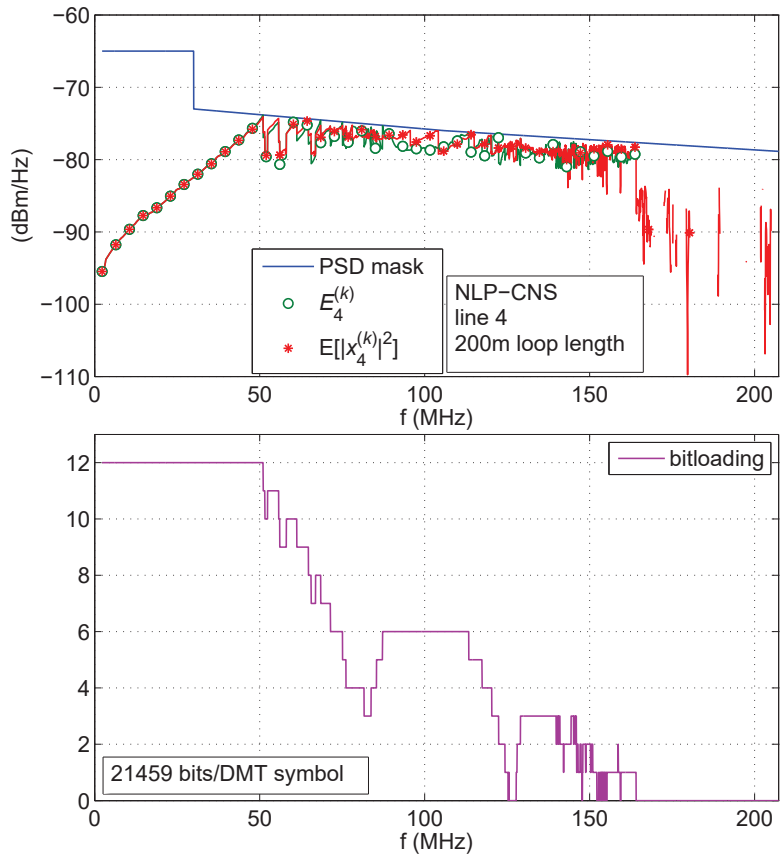

Figure 7.7: PSD values and bitloading for line 4 of the 200 m loop with TCM and NLP-CNS.

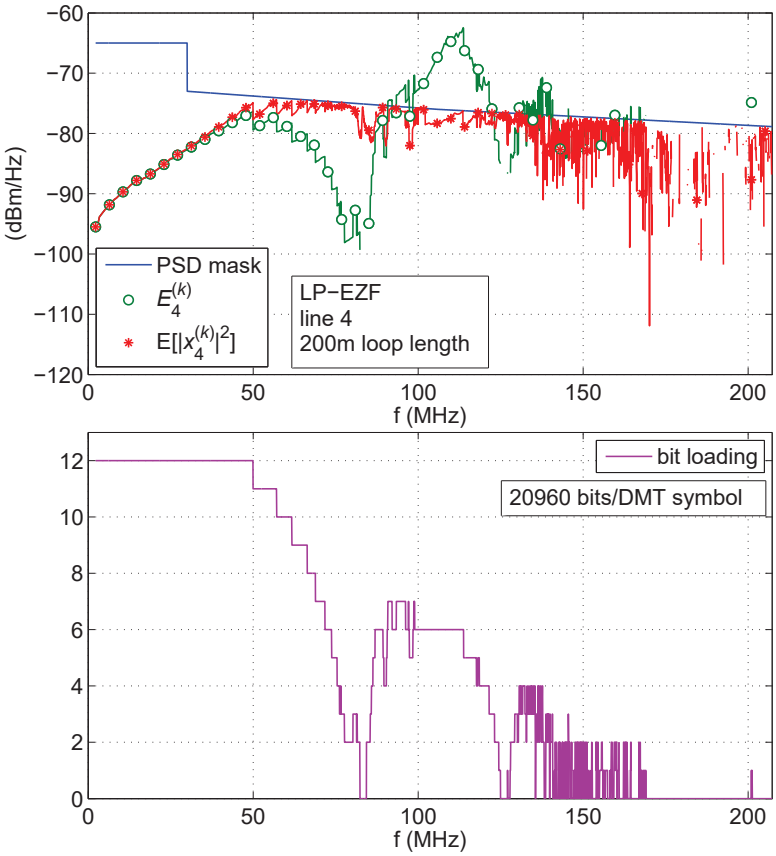

Figure 7.8: PSD values and bitloading for line 4 of the 200 m loop with TCM and LP-EZF.

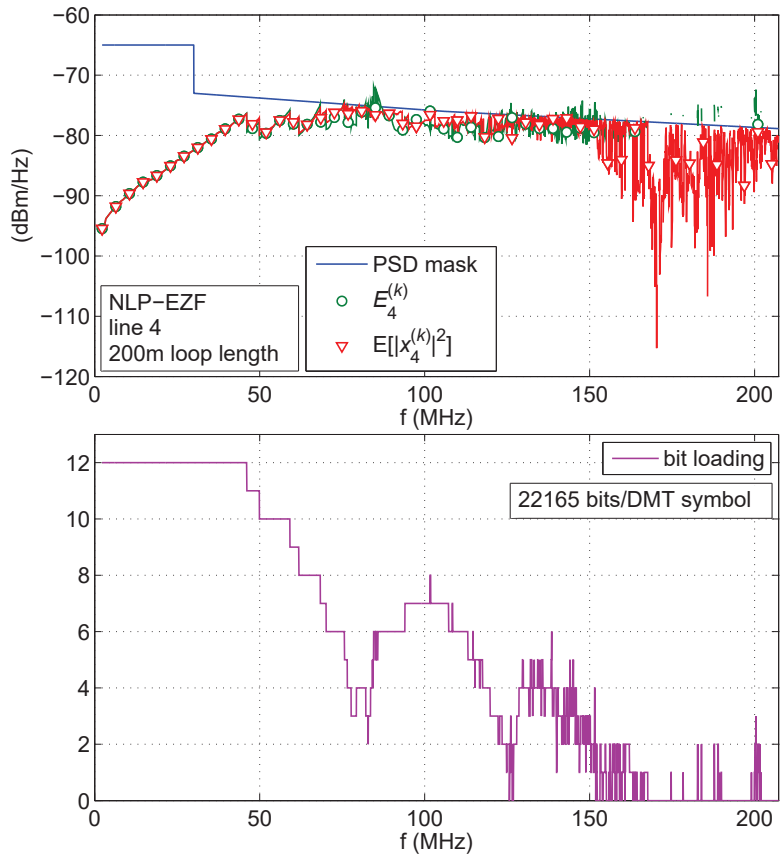

Figure 7.9: PSD values and bitloading for line 4 of the 200 m loop with TCM and NLP-EZF.

Table 7.3: Average information bitrates (Gbps) per line for TCM.

	100m	150m	200m	250m
LP-EZF	1.712	1.369	0.941	0.720
LP-CNS	1.665	1.324	0.904	0.690
NLP-EZF	1.846	1.482	1.055	0.786
NLP-CNS	1.825	1.460	1.040	0.775

Table 7.4: Computation time to determine the bitloading for TCM and all lines on all tones.

	100m	150m	200m	250m
LP-EZF	2.28 s	6.62 s $\otimes$	8.88 s $\otimes$	7.78 s $\otimes$
LP-CNS	0.14 s	0.11 s	0.15 s	0.14 s
NLP-EZF	2.55 s	8.59 s $\otimes$	20.91 s $\otimes$	16.11 s $\otimes$
NLP-CNS	0.21 s	0.22 s	0.36 s $\otimes$	0.19 s

NLP, respectively.

Table 7.4 shows the computation times for the various combinations of precoding technique and bitloading algorithm; the symbol $\otimes$ again indicates that part II of the algorithm had to be executed. We observe that CNS is at least a factor of 10 faster than EZF. Similar as for uncoded transmission, the slightly higher information rate of TCM resulting from EZF compared to CNS comes at the cost of a considerably higher computational complexity.

7.3 LDPC Codes

In the case of LDPC coding, the bit allocation algorithms make use of tables, containing the SNR values for which the target BER is achieved for all QAM constellations up to 2^{12}-QAM, for each code rate R_c and block length K; these tables are similar to Table 5.11, which was specifically obtained for the LDPC code with $R_c = 20/21$ and $K = 4320$. The required SNR values can be computed from the I_{avg} values given in Table 5.10. Although the LDPC codewords have different constellation sizes on the different tones in the DMT symbol when transmitting over the frequency-selective DSL channel, we have verified (in a

Table 7.5: Average information bitrates (Gbps) per line for all proposed LDPC codes and loop length equal to 100 m.

100m	R_c	1/2	2/3	5/6	16/18	20/21
$K = 960$	LP-EZF	1.080	1.361	1.604	1.653	**1.698**
	LP-CNS	1.064	1.334	1.566	1.613	**1.654**
	NLP-EZF	1.110	1.430	1.708	1.768	**1.830**
	NLP-CNS	1.104	1.412	1.690	1.751	**1.806**
$K = 4320$	LP-EZF	1.096	1.387	1.637	1.702	**1.757**
	LP-CNS	1.082	1.361	1.600	1.660	**1.714**
	NLP-EZF	1.118	1.449	1.736	1.814	**1.882**
	NLP-CNS	1.114	1.432	1.716	1.794	**1.864**

similar way as for TCM) that these SNR values remain valid to achieve the BER constraint; this is no surprise, because the error performance is essentially determined by the I_{avg} value.

In this section, we show results pertaining to the SMSA decoder only, as it achieves the target BER at lower SNR values than the SPA decoder and furthermore, it has a lower computation complexity.

Table 7.5 compares the EZF and CNS bitloading algorithm in terms of the average resulting information bitrate, for LP and NLP and a loop length of 100 m; note that for a given code rate the longer codes yield the larger information bitrate, because their better error performance allows to use larger constellations. In Table 7.6 we show results for CNS-NLP and the LDPC code for all rates and for $K = 4320$ only. We consider loop lengths of respectively 150 m, 200 m and 250 m. We observe from Tables 7.5 and 7.6 that the more powerful lower-rate codes are outperformed in terms of information bitrate by the less powerful higher-rate codes. Compared to higher-rate codes, the lower-rate codes allow to use larger constellations for given SNR, but these larger constellations contain a smaller fraction of information bits; the latter effect turns out to be dominant so that the information rate increases with the code rate. This can be explained by noting that many tones (at low and medium frequencies) operate at high SNR, in which case that the higher code rates are more advantageous in terms of information bitrate (see discussion regarding Figure 6.3).

Table 7.6: Average information bitrates (Gbps) per line for all proposed LDPC codes with $K = 4320$ and loop lengths equal to respectively 150 m, 200 m and 250 m.

	R_c	1/2	2/3	5/6	16/18	20/21
150 m	LP-EZF	0.945	1.166	1.354	1.393	**1.418**
	LP-CNS	0.920	1.130	1.307	1.346	**1.371**
	NLP-EZF	0.992	1.242	1.457	1.507	**1.538**
	NLP-CNS	0.982	1.226	1.430	1.481	**1.510**
200 m	LP-EZF	0.677	0.819	0.946	0.971	**0.984**
	LP-CNS	0.642	0.784	0.899	0.926	**0.937**
	NLP-EZF	0.778	0.945	1.084	1.109	**1.113**
	NLP-CNS	0.748	0.909	1.049	1.076	**1.088**
250 m	LP-EZF	0.528	0.637	0.728	0.747	**0.752**
	LP-CNS	0.506	0.605	0.694	0.713	**0.719**
	NLP-EZF	0.586	0.710	0.810	0.829	**0.829**
	NLP-CNS	0.562	0.684	0.784	0.805	**0.809**

Table 7.7: Average information bitrates (Gbps) per line for RS(255,236) + byte-interleaver + TCM.

	100m	150m	200m	250m
LP-EZF	1.681	1.372	0.958	0.735
LP-CNS	1.641	1.325	0.916	0.703
NLP-EZF	1.795	1.481	1.079	0.806
NLP-CNS	1.776	1.456	1.058	0.790

Table 7.8: Average information bitrates (Gbps) per line for RS(136,120) + byte-interleaver + TCM .

	100m	150m	200m	250m
LP-EZF	1.588	1.298	0.907	0.697
LP-CNS	1.550	1.253	0.867	0.666
NLP-EZF	1.695	1.400	1.022	0.763
NLP-CNS	1.676	1.337	1.003	0.748

7.4 RS + Byte-Interleaver + TCM

In Section 5.5.2, we analyzed the concatenation of RS and TCM assuming a constant constellation size and constant SNR on all tones, for two RS codes: RS(255,239) and RS(136,120). Here we apply the bitloading, using the resulting SNR Table 5.12, for the different DSL loops at our disposal. Figure 7.10 compares the PSD mask with the transmit energy $\mathbb{E}[|x_i^{(k)}|^2]$ and the constellation energy $E_i^{(k)}$ for RS(255,236) + byte-interleaver + TCM and LP with the EZF algorithm, along with the corresponding bitloading, for an arbitrary line ($i = 4$) in the 200 m DSL loop. As the concatenated coding scheme is more powerful than TCM, we can transmit larger constellation symbols, compared to Figure 7.8. This results in a large number of coded bits per DMT symbol.

The resulting average information bitrates per line are given in Tables 7.7 and 7.8 for the concatenated codes with respectively RS(255,239) and RS(136,120).

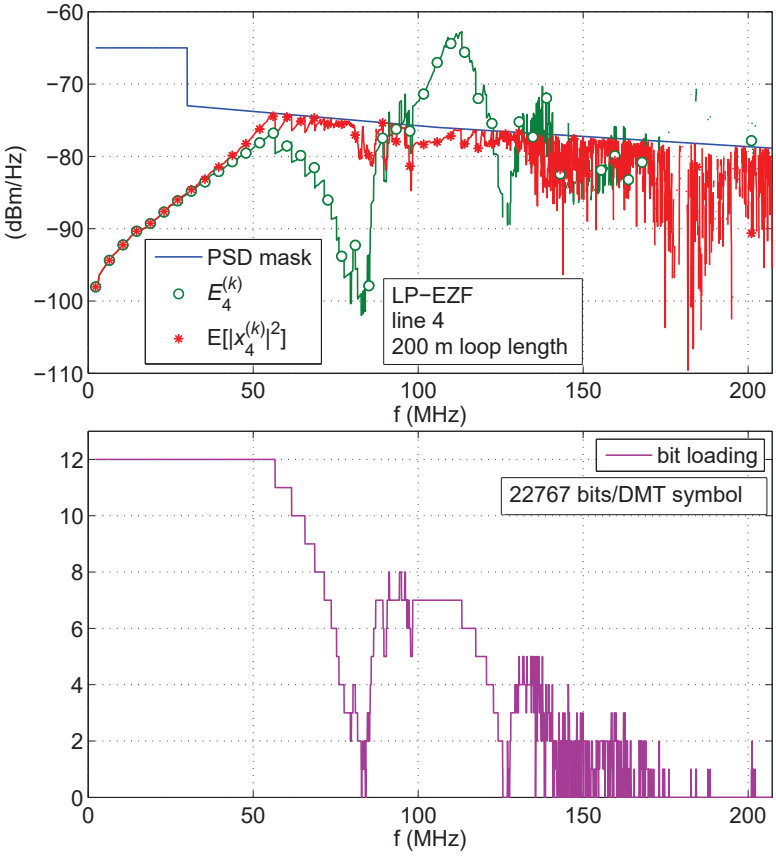

Figure 7.10: PSD values and bitloading for line 4 of the 200 m loop with RS(255,236) + byte-interleaver + TCM and LP-EZF.

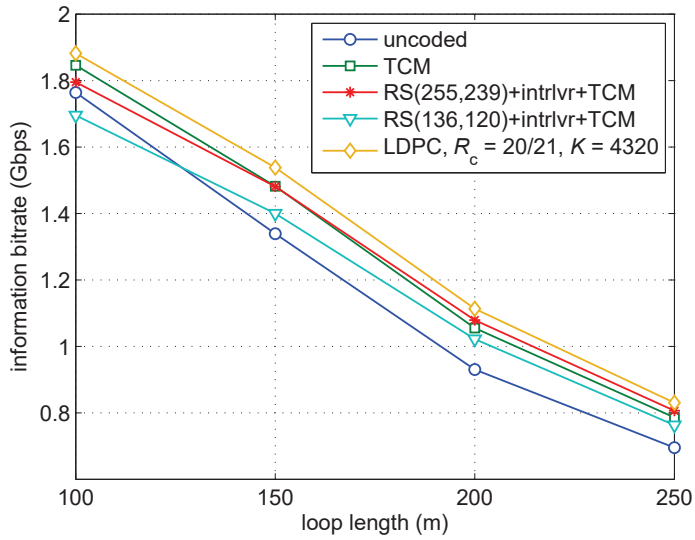

Figure 7.11: Comparison of the average information bitrate versus loop length for EZF-NLP and several protection strategies.

7.5 Comparison

Figure 7.11 shows the average information bitrates versus the loop length for all protection strategies considered so far in this chapter, assuming NLP-EZF. In order not to overload the figure, the results for the LDPC codes are limited to the LDPC code of rate 20/21 and $K = 4320$, as it achieves the largest information bitrates from Tables 7.5-7.6.

This comparison points out that the rate 20/21 LDPC code outperforms the other protection strategies for all loop lengths. Depending on the loop length, either TCM or RS(255,239) + byte-interleaver + TCM provide the second largest information bitrates; the former is better for the larger loop lengths, which operate at lower SNR. The RS(136,120) + byte-interleaver + TCM concatenation performs worse in terms of information bitrate due to the larger encoding overhead compared to RS(255,239) + byte-interleaver + TCM. Uncoded transmission achieves the lowest information bitrate for 150 m, 200 m and 250 m loop lengths, whereas for a 100 m loop the RS(136,120) + byte-interleaver + TCM concatenation yields the smallest information bitrate. These conclusions also hold for the other combinations of precoder and bitloading algorithms.

7.6 ARQ

In this section, we add ARQ on top of the protection strategies from the previous sections, except for the concatenation of RS and TCM, because the use of the RS outer code is already considered as an additional protection of TCM.

When using ARQ, the residual BER after N_{retr} retransmissions can be reduced by increasing N_{retr} (at the expense of increased latency). Hence, under a BER constraint, the number of information bits per DMT symbol increases with N_{retr}. However, as some of the DMT symbols also contain retransmitted data, we consider the goodput (see Section 2.8) as the proper performance indicator. Therefore, we will adopt the following approach in the case of ARQ:

1. We apply the bitloading using the SNR tables associated with a target value of the BER in the absence of ARQ, denoted $\text{BER}^{(0)}$. From the obtained bitloading, we determine for each line the corresponding WER in the absence of ARQ, along with the associated goodput, denoted respectively $\text{WER}_i^{(0)}$ and GP_i for line i; we have $\text{GP}_i = R_{\text{b},i}(1 - \text{WER}_i^{(0)})$, where $R_{\text{b},i}$ is the information bitrate on line i. We compute the average goodput over all lines, denoted $\overline{\text{GP}}$. This is repeated for several values of the BER target value, which yields $\overline{\text{GP}}$ as a function of $\text{BER}^{(0)}$.

2. We determine the value $\text{BER}_{\text{opt}}^{(0)}$ of $\text{BER}^{(0)}$, such that $\overline{\text{GP}}$ achieves its maximum value $\overline{\text{GP}}_{\text{opt}}$. The corresponding WER for line i is denoted $\text{WER}_{\text{opt},i}^{(0)}$.

3. Using the expression $\text{BER}_{\text{opt},i}^{(N_{\text{retr}})} = \text{BER}_{\text{opt}}^{(0)} \left(\text{WER}_{\text{opt},i}^{(0)}\right)^{N_{\text{retr}}}$ for the residual BER on line i when allowing N_{retr} retransmissions, we compute the average residual BER over all lines, denoted $\overline{\text{BER}}_{\text{opt}}^{(N_{\text{retr}})}$, as a function of N_{retr}. We determine the smallest value of N_{retr} (denoted $N_{\text{retr,opt}}$) which yields $\overline{\text{BER}}_{\text{opt}}^{(N_{\text{retr}})} \leq 10^{-7}$.

4. The system will be operated using the bitloading corresponding to $\text{BER}^{(0)} = \text{BER}_{\text{opt}}^{(0)}$, and using ARQ allowing $N_{\text{retr}} = N_{\text{retr,opt}}$ retransmissions. This way, the resulting average goodput equals the maximum value $\overline{\text{GP}}_{\text{opt}}$, and the residual BER after $N_{\text{retr,opt}}$ retransmissions does not exceed 10^{-7}. Of course, one can achieve $\overline{\text{GP}} = \overline{\text{GP}}_{\text{opt}}$ with $\overline{\text{BER}}_{\text{opt}}^{(N_{\text{retr}})} \leq 10^{-7}$ for any $N_{\text{retr}} \geq N_{\text{retr,opt}}$, but selecting $N_{\text{retr}} = N_{\text{retr,opt}}$ yields the maximum average output with minimum latency.

The word error rate $\text{WER}_i^{(0)}$ (or a close approximation thereof) has been determined from the symbol error probability in the case of uncoded transmission, from the trellis section error probability in the case of TCM, from the average mutual information per bit I_{avg} in the case of LDPC codes, and from the TCM byte error rate in the case of the concatenation of RS and TCM.

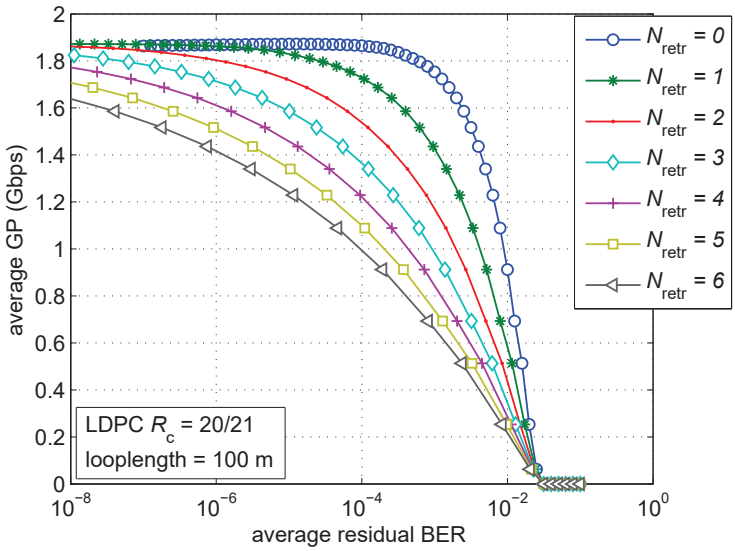

Figure 7.12: Average goodput $\overline{\mathrm{GP}}$ versus $\overline{\mathrm{BER}}^{(N_{\mathrm{retr}})}$ for the rate 20/21 LDPC code with $K = 4320$, NLP-CNS and various N_{retr} .

The system with ARQ will be compared to a system without ARQ, where the bitloading is applied using a target BER value $\mathrm{BER}^{(0)} = 10^{-7} < \mathrm{BER}^{(0)}_{\mathrm{opt}}$, yielding $\overline{\mathrm{GP}} < \overline{\mathrm{GP}}_{\mathrm{opt}}$. Hence, the system with ARQ has the larger constellation sizes.

In order to illustrate the above approach, we consider the system with the rate 20/21 LDPC code with $K = 4320$ and NLP-CNS. Figure 7.12 shows the average goodput, $\overline{\mathrm{GP}}$, as a function of the average residual BER after N_{retr} retransmissions, $\overline{\mathrm{BER}}^{(N_{\mathrm{retr}})}$; the latter is obtained by averaging $\mathrm{BER}_i^{(N_{\mathrm{retr}})} = \mathrm{BER}^{(0)} \left(\mathrm{WER}_i^{(0)}\right)^{N_{\mathrm{retr}}}$ over all lines. Let us concentrate on the curve for $N_{\mathrm{retr}} = 0$, i.e., $\overline{\mathrm{GP}}$ versus the value $\mathrm{BER}^{(0)}$ of the target BER. For a better understanding of the behavior of this curve, we also show $\mathrm{BER}^{(0)}$ and $\overline{\mathrm{WER}}^{(0)}$as a function of $\overline{R}_{\mathrm{b}}$ in Figure 7.13, and $\overline{\mathrm{GP}}$ as a function of $\overline{R}_{\mathrm{b}}$ in Figure 7.14; $\overline{\mathrm{WER}}^{(0)}$ and $\overline{R}_b$ denote the averages over all lines of $\mathrm{WER}_i^{(0)}$ and $R_{b,i}$, respectively.

- In the region of small $\mathrm{BER}^{(0)}$, we have $\mathrm{WER}_i^{(0)} \ll 1$, yielding $1 - \mathrm{WER}_i^{(0)} \approx 1$ for all lines. Hence, in this region of $\mathrm{BER}^{(0)}$, $\overline{\mathrm{GP}}$ is close to $\overline{R}_{\mathrm{b}}$, and increases with increasing $\mathrm{BER}^{(0)}$.

- In the region of large $\mathrm{BER}^{(0)}$, we have $\mathrm{WER}_i^{(0)} \approx 1$, yielding $1 - \mathrm{WER}_i^{(0)} \ll$

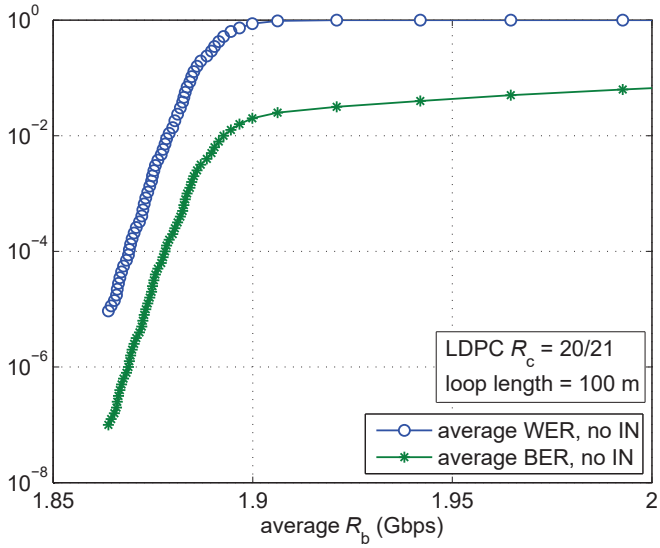

Figure 7.13: $\overline{\text{WER}}^{(0)}$ and $\text{BER}^{(0)}$ versus $\overline{R}_\text{b}$ for the rate 20/21 LDPC code with $K = 4320$, NLP-CNS and $N_\text{retr} = 0$.

1. Hence, $\overline{\text{GP}}$ is much smaller than $\overline{R}_\text{b}$. Increasing $\text{BER}^{(0)}$ yields an increase of $\overline{R}_\text{b}$ but a decrease of $1 - \text{WER}_i^{(0)}$. The latter effect is dominant in the considered region of $\text{BER}^{(0)}$, hence $\overline{\text{GP}}$ decreases with increasing $\text{BER}^{(0)}$.

- In between, $\overline{\text{GP}}$ achieves a maximum value $\overline{\text{GP}}_\text{opt}$ at $\text{BER}^{(0)} = \text{BER}_\text{opt}^{(0)}$.

For $N_\text{retr} > 1$, the maximum goodput value $\overline{\text{GP}}_\text{opt}$ occurs at $\overline{\text{BER}}^{(N_\text{retr})} = \overline{\text{BER}}_\text{opt}^{(N_\text{retr})}$, which implies that the abscissa of the maximum goodput value shifts to the left with increasing N_retr. For the case at hand, we have $\text{BER}_\text{opt}^{(0)} = 1.3 \cdot 10^{-5}$ and $\overline{\text{BER}}_\text{opt}^{(1)} = 1.3 \cdot 10^{-8}$, so that $N_\text{retr,opt} = 1$.

Tables 7.9-7.12 show for the different protection strategies the maximum average goodput under the error performance constraint $\text{BER} \leq 10^{-7}$ for the cases with ARQ and without ARQ. For uncoded transmission and TCM, the information word consists of 4320 bits and the LDPC codes have $K = 4320$ information bits; for the concatenation of the RS outer code, the infinite byte interleaver and TCM, only the traditional RS(255, 239) code is considered. In the case of ARQ, the value of $N_\text{retr,opt}$ is indicated between parentheses . The following observations can be made:

- In the case of ARQ, we achieve $\overline{\text{GP}} = \overline{\text{GP}}_\text{opt}$ for $N_\text{retr,opt} = 1$. The latency

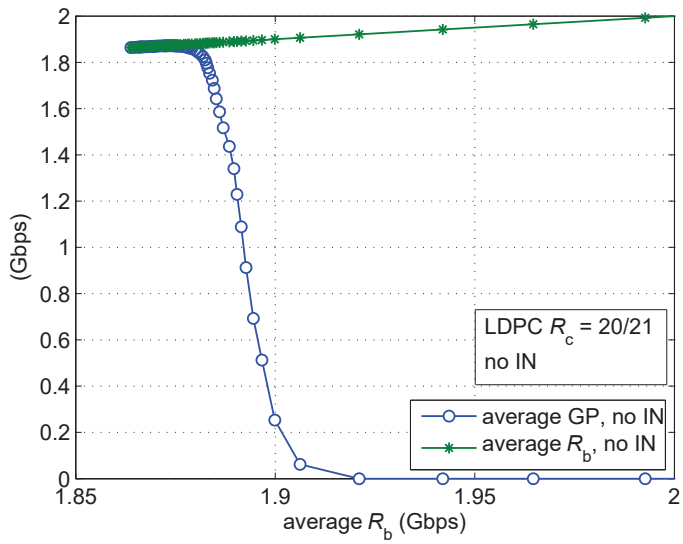

Figure 7.14: $\overline{R}_\mathrm{b}$ and $\overline{\mathrm{GP}}$ versus $\overline{R}_\mathrm{b}$ for the rate 20/21 LDPC code with $K = 4320$ and NLP-CNS.

constraint $N_\mathrm{retr} \leq 6$ from Section 5.6.2 does not limit the goodput. The use of ARQ provides only a small increase of the goodput (less than 1%), compared to the case where no ARQ is used.

- Among all considered protection strategies, the largest $\overline{\mathrm{GP}}$ is obtained for the rate 20/21 LDPC code with $N_\mathrm{retr} = 1$. A close second-best protection strategy is the rate 20/21 LDPC code without ARQ.
- As already discussed in detail in the previous sections, NLP-EZF outperforms the other bitloading and precoder combinations. NLP-CNS performs only slightly worse, but has the advantage of a much lower computational complexity.
- Note that the considered bitloading algorithms are suboptimal and that the comparison of the codes and precoding schemes is done for those two specific suboptimal bitloading algorithms. To find an optimal bitloading solution for all tones and users, an extensive search should be performed over all possible combinations of bitloadings. This is very computationally complex and therefore not considered in this dissertation.
- The increase in GP for NLP as compared to LP depends on the applied code, bitloading algorithm, loop length and whether ARQ is used or not. E.g., without ARQ, with EZF and loop length equal to 100 m, the increase

Table 7.9: Average goodput (Gbps) per line for BER $\leq 10^{-7}$ for 100 m loop length.

	LP-EZF	LP-CNS	NLP-EZF	NLP-CNS
uncoded, no ARQ	1.616	1.567	1.755	1.733
uncoded, ARQ	1.645 (1)	1.598 (1)	1.783 (1)	1.760 (1)
TCM, no ARQ	1.704	1.657	1.838	1.818
TCM, ARQ	1.731 (1)	1.684 (1)	1.860 (1)	1.841 (1)
LDPC 1/2, no ARQ	1.095	1.081	1.118	1.114
LDPC 1/2, ARQ	1.098 (1)	1.084 (1)	1.118 (1)	1.115 (1)
LDPC 2/3, no ARQ	1.387	1.361	1.449	1.432
LDPC 2/3, ARQ	1.393 (1)	1.367 (1)	1.452 (1)	1.435 (1)
LDPC 5/6, no ARQ	1.637	1.600	1.736	1.716
LDPC 5/6, ARQ	1.644 (1)	1.608 (1)	1.743 (1)	1.722 (1)
LDPC 16/18, no ARQ	1.702	1.660	1.814	1.794
LDPC 16/18, ARQ	1.710 (1)	1.670 (1)	1.820 (1)	1.802 (1)
LDPC 20/21, no ARQ	1.757	1.714	1.882	1.864
LDPC 20/21, ARQ	**1.766** (1)	**1.723** (1)	**1.889** (1)	**1.872** (1)
RS(255,239)+TCM, no ARQ	1.681	1.641	1.795	1.776

ranges from 2.1% for the LDPC code of rate 1/2 and $K = 4320$ up to 8.6% for uncoded transmission. The codes with the smaller code rate have the smaller advantage of using NLP over LP.

- The increase in GP for EZF as compared to CNS depends on the applied code, precoding scheme, loop length and whether ARQ is used or not. E.g., without ARQ, with LP and loop length equal to 100 m, the increase ranges from 1.3% for LDPC code of rate 1/2 and $K = 4320$ up to 3% for uncoded transmission. The codes with the smaller code rate have the smaller advantage of using EZF over CNS.

Table 7.10: Average goodput (Gbps) per line for BER $\leq 10^{-7}$ for 150 m loop length.

	LP-EZF	LP-CNS	NLP-EZF	NLP-CNS
uncoded, no ARQ	1.231	1.207	1.332	1.319
uncoded, ARQ	1.270 (1)	1.239 (1)	1.376 (1)	1.353 (1)
TCM, no ARQ	1.345	1.317	1.477	1.455
TCM, ARQ	1.377 (1)	1.350 (1)	1.511 (1)	1.486 (1)
LDPC 1/2, no ARQ	0.945	0.919	0.992	0.982
LDPC 1/2, ARQ	0.949 (1)	0.924 (1)	0.995 (1)	0.985 (1)
LDPC 2/3, no ARQ	1.166	1.130	1.249	1.226
LDPC 2/3, ARQ	1.273 (1)	1.137 (1)	1.248 (1)	1.232 (1)
LDPC 5/6, no ARQ	1.354	1.307	1.457	1.430
LDPC 5/6, ARQ	1.363 (1)	1.316 (1)	1.465 (1)	1.439 (1)
LDPC 16/18, no ARQ	1.393	1.346	1.507	1.481
LDPC 16/18, ARQ	1.402 (1)	1.355 (1)	1.515 (1)	1.488 (1)
LDPC 20/21, no ARQ	1.418	1.371	1.538	1.509
LDPC 20/21, ARQ	**1.430** (1)	**1.381** (1)	**1.549** (1)	**1.521** (1)
RS(255,239)+TCM, no ARQ	1.372	1.352	1.481	1.456

Table 7.11: Average goodput (Gbps) per line for BER $\leq 10^{-7}$ for 200 m loop length.

	LP-EZF	LP-CNS	NLP-EZF	NLP-CNS
uncoded, no ARQ	0.838	0.813	0.926	0.915
uncoded, ARQ	0.867 (1)	0.839 (1)	0.962 (1)	0.945 (1)
TCM, no ARQ	0.937	0.900	1.051	1.036
TCM, ARQ	0.968 (1)	0.924 (1)	1.087 (1)	1.068 (1)
LDPC 1/2, no ARQ	0.677	0.642	0.778	0.746
LDPC 1/2, ARQ	0.682 (1)	0.647 (1)	0.781 (1)	0.750 (1)
LDPC 2/3, no ARQ	0.819	0.784	0.945	0.909
LDPC 2/3, ARQ	0.825 (1)	0.780 (1)	0.951 (1)	0.916 (1)
LDPC 5/6, no ARQ	0.946	0.899	1.084	1.049
LDPC 5/6, ARQ	0.952 (1)	0.906 (1)	1.093 (1)	1.058 (1)
LDPC 16/18, no ARQ	0.971	0.926	1.109	1.076
LDPC 16/18, ARQ	0.979 (1)	0.932 (1)	1.118 (1)	1.084 (1)
LDPC 20/21, no ARQ	0.984	0.936	1.112	1.088
LDPC 20/21, ARQ	**0.993** (1)	**0.944** (1)	**1.124** (1)	**1.097** (1)
RS(255,239)+TCM, no ARQ	0.958	0.916	1.079	1.058

Table 7.12: Average goodput (Gbps) per line for BER $\leq 10^{-7}$ for 250 m loop length.

	LP-EZF	LP-CNS	NLP-EZF	NLP-CNS
uncoded, no ARQ	0.635	0.613	0.692	0.685
uncoded, ARQ	0.658 (1)	0.633 (1)	0.717 (1)	0.703 (1)
TCM, no ARQ	0.717	0.687	0.783	0.778
TCM, ARQ	0.740 (1)	0.710 (1)	0.810 (1)	0.798 (1)
LDPC 1/2, no ARQ	0.528	0.503	0.586	0.562
LDPC 1/2, ARQ	0.531 (1)	0.506 (1)	0.589 (1)	0.566 (1)
LDPC 2/3, no ARQ	0.637	0.605	0.710	0.684
LDPC 2/3, ARQ	0.641 (1)	0.610 (1)	0.715 (1)	0.689 (1)
LDPC 5/6, no ARQ	0.728	0.694	0.810	0.784
LDPC 5/6, ARQ	0.734 (1)	0.700 (1)	0.817 (1)	0.790 (1)
LDPC 16/18, no ARQ	0.747	0.713	0.829	0.805
LDPC 16/18, ARQ	0.753 (1)	0.718 (1)	0.835 (1)	0.811 (1)
LDPC 20/21, no ARQ	0.752	0.719	0.829	0.809
LDPC 20/21, ARQ	**0.759** (1)	**0.725** (1)	**0.838** (1)	**0.818** (1)
RS(255,239)+TCM, no ARQ	0.735	0.703	0.806	0.790

8 Information Rates in the Presence of Impulsive Noise

In this chapter, we apply the bit allocation algorithms from Chapter 6 to the DSL channel as described in Chapter 4; we use the same DSL loops as described in Chapter 7, additionally here we assume the presence of IN on the channel.

We compare the resulting information bitrates and goodput for the different precoding schemes and bit allocation algorithms in a G.fast downlink scenario with impulsive noise for different protection techniques. For each protection technique considered, we distinguish between two cases. First, we assume that the parameters (p_1, κ) of the impulsive noise model are known. In the second case, we have no information about the impulsive noise model.

In Sections 8.1, 8.2, 8.3, and 8.4 we consider respectively uncoded transmission, TCM, LDPC codes, and the concatenation of RS outer code + byte-interleaver + TCM inner code, all in the case where the parameters of the IN model are known. A comparison of the achieved information bitrates is presented in Section 8.5. Section 8.6 deals with the case where the parameters of the IN model are unknown. The use of ARQ is investigated in Section 8.7.

Table 8.1: Average information bitrates (Gbps) per line for uncoded transmission and IN with $\kappa = 10$ dB.

	100m	150m	200m	250m
LP-EZF	1.173	0.848	0.587	0.447
LP-CNS	1.031	0.739	0.516	0.402
NLP-EZF	1.281	0.912	0.626	0.471
NLP-CNS	1.271	0.889	0.616	0.462

8.1 Uncoded Transmission

In the case of uncoded transmission and under the assumption that the parameters of the IN model are known, the bit allocation algorithms make use of Tables 5.6 and 5.7 which contain the values of $\mathrm{SNR}_{\mathrm{avg}}$ for which the BER (averaged over the DMT symbols with and without IN) equals the target BER of 10^{-7}, for constellation sizes up to 2^{12}-QAM and respectively $\kappa = 10$ dB and $\kappa = 20$ dB. After applying the bit allocation algorithms on the different DSL loops, the following results are obtained. Figures 8.1 and 8.2 show the PSD mask, the transmit energy $\mathbb{E}[|x_i^{(k)}|^2]$ and the constellation energy $E_i^{(k)}$ for uncoded transmission with NLP-CNS and respectively $\kappa = 10$ dB and $\kappa = 20$ dB, along with the corresponding bitloading, for an arbitrary line ($i = 4$) in the 200 m DSL loop. Notice that for an increasing level of IN fewer tones are used, more specifically only the lower-frequency tones remain active. It is clear that IN has a large impact on the obtainable information bitrate. Tables 8.1 and 8.2 compare the resulting information bitrates for all combinations of precoding types and bitloading algorithms. We observe that NLP outperforms LP by about 22% and 5.6% for CNS and EZF, respectively. Regarding the difference between EZF and CNS, EZF outperforms CNS by about 18% and 2.4% for LP and NLP, respectively.

8.2 TCM

For TCM, similar computations are performed as for uncoded transmission in the presence of IN and the results are presented in this section. Tables 5.8 and 5.9 give the values of $\mathrm{SNR}_{\mathrm{avg}}$ that are required to achieve exactly the target BER equal to 10^{-7} for respectively $\kappa = 10$ dB and $\kappa = 20$ dB. These SNR values are determined specifically for TCM codewords with a constant constellation size on all tones in the DMT. To show that these table values are also valid for

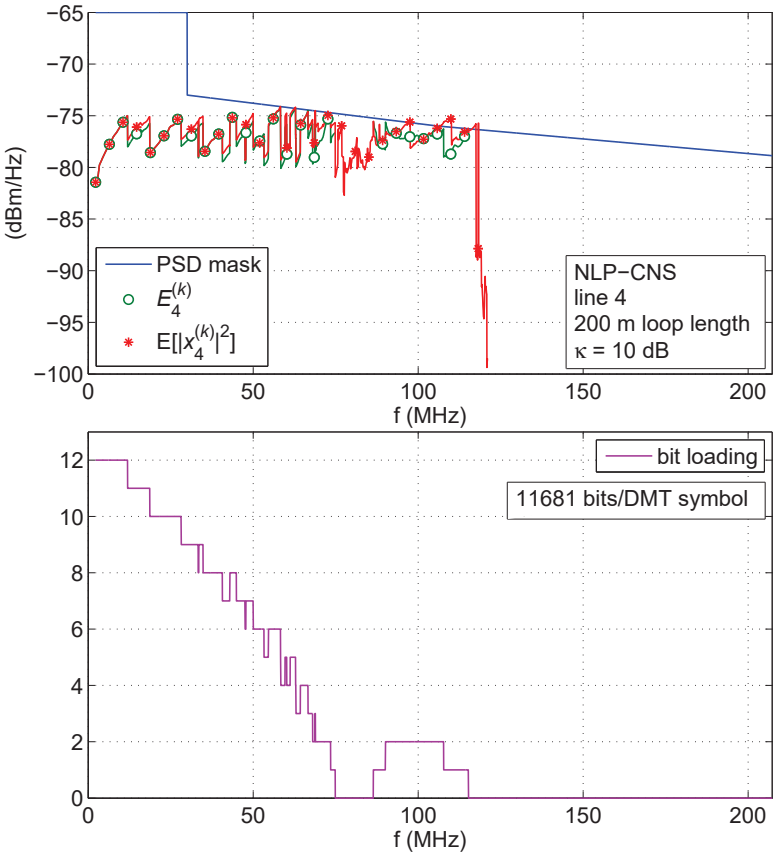

Figure 8.1: PSD values and bitloading for line 4 of the 200 m loop with $\kappa = 10$ dB, uncoded transmission and NLP-CNS.

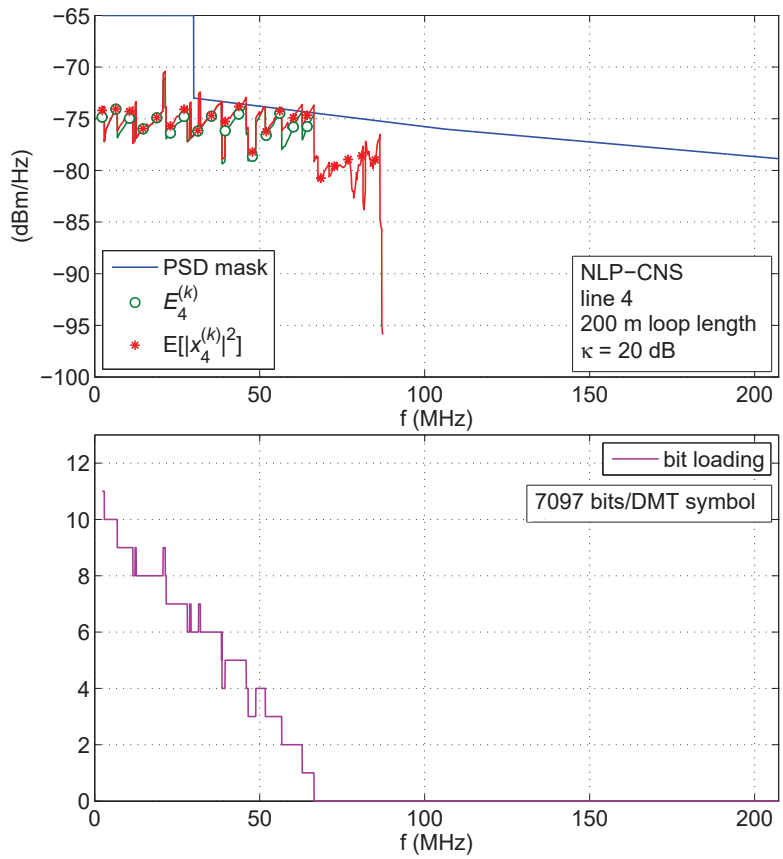

Figure 8.2: PSD values and bitloading for line 4 of the 200 m loop with $\kappa = 20$ dB, uncoded transmission and NLP-CNS.

Table 8.2: Average information bitrates (Gbps) per line for uncoded transmission and IN with $\kappa = 20$ dB.

	100m	150m	200m	250m
LP-EZF	0.734	0.515	0.374	0.276
LP-CNS	0.590	0.414	0.306	0.231
NLP-EZF	0.789	0.544	0.380	0.280
NLP-CNS	0.777	0.515	0.372	0.271

Table 8.3: Average information bitrates (Gbps) per line for TCM and IN with $\kappa = 10$ dB.

	100m	150m	200m	250m
LP-EZF	1.336	0.980	0.668	0.510
LP-CNS	1.323	0.843	0.663	0.435
NLP-EZF	1.455	1.053	0.717	0.543
NLP-CNS	1.440	1.038	0.705	0.534

TCM codewords with different constellation sizes on different tones in the DMT, simulations are performed on a real DSL loop for target BER values ranging from 10^{-2} to 10^{-7}. Similarly as for 10^{-7}, we obtained tables with SNR values for the other target BER values. The resulting BER is shown in Figure 8.3 versus the target BER for the 100 m DSL loop with 8 lines, IN with $\kappa = 10$ dB and NLP-EZF. It is clear that there is a good correspondence between resulting BER and target BER. Also for TCM, IN has a large impact on the obtainable information rates. Tables 8.3 and 8.4 show the resulting average information bitrate for EZF and CNS, for LP and NLP and for respectively $\kappa = 10$ dB and $\kappa = 20$ dB. We observe that the combination NLP-EZF achieves the highest bitrate, but only slightly better than NLP-CNS at the cost of a considerably higher computational complexity. Compared to NLP-CNS, NLP-EZF yields an information rate which is only 1.4% and 3.5% larger, for $\kappa = 10$ dB and $\kappa = 20$ dB, respectively.

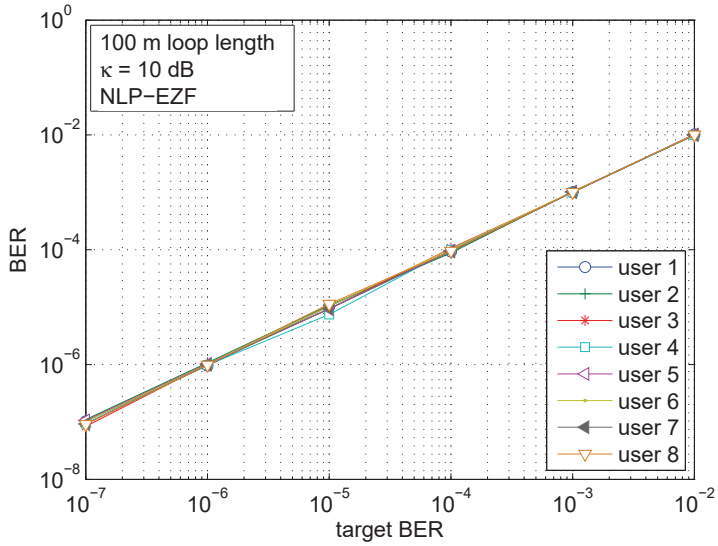

Figure 8.3: Resulting BER versus target BER for TCM, IN with $\kappa = 10$ dB, NLP-EZF and the 100 m DSL loop with 8 users.

Table 8.4: Average information bitrates (Gbps) per line for TCM and IN with $\kappa = 20$ dB.

	100m	150m	200m	250m
LP-EZF	0.863	0.611	0.439	0.329
LP-CNS	0.787	0.559	0.354	0.270
NLP-EZF	0.933	0.651	0.454	0.337
NLP-CNS	0.902	0.618	0.443	0.337

8.3 LDPC Codes

When the parameters of the IN model are known, the bit allocation algorithms make use of tables specified for each rate R_c and block length K, similar to Table 5.11, specifically obtained for the LDPC code with $R_c = 20/21$ and $K = 4320$. These tables contain the values of $\mathrm{SNR}_{\mathrm{avg}}$ for which the target BER of 10^{-7} is achieved for all QAM constellations up to 2^{12}−QAM and $\kappa = 10$ dB and $\kappa = 20$ dB. Although the LDPC codes here have different constellation sizes on the different tones in the DMT symbol, these SNR values remain valid as the error performance, which is dominated by the occurrence of IN, is completely determined by the average MI per bit, I_{avg}.

In the presence of IN, we use the SMSA decoder with simplified LLR computation, which does not require knowledge of the instantaneous SNR. Table 8.5 compares the LDPC code rates in terms of the average resulting information bitrate, with $K = 4320$ and NLP-CNS and for respectively a loop length of 100 m, 150 m, 200 m and 250 m. For each loop length, we indicated which code yields the largest information bitrate (marked in bold). We can conclude that either the rate 5/6 LDPC code or the rate 16/18 LDPC code provide the largest information bit rate for $\kappa = 10$ dB, depending on the loop length; as both codes give rise to nearly the same information rate on the 100 m loop, the rate 16/18 LDPC code is a good choice for all loop lengths considered. For $\kappa = 20$ dB, the rate 5/6 code outperforms all other codes. Compared to the case without IN, where the rate 20/21 LDPC code was optimum, we observe that for higher noise levels the code rate of the optimum code gets smaller; this is in agreement with the discussion regarding Figure 6.3. These results can be generalized for the other combinations of precoder and bitloading algorithm.

8.4 RS + Byte-Interleaver + TCM

For the concatenated code RS + byte-interleaver + TCM, we obtained SNR tables for the two RS codes RS(255,236) and RS(136,120). The values of $\mathrm{SNR}_{\mathrm{avg}}$ required to achieve the target BER equal to 10^{-7} are given in Tables 5.13 and 5.14, for $\kappa = 10$ dB and $\kappa = 20$ dB, respectively. The resulting bitloading for the concatenated code with RS(255,236) is shown together with the transmit energy $\mathbb{E}[|x_i^{(k)}|^2]$ and the constellation energy $E_i^{(k)}$ in Figures 8.4 and 8.5 for an arbitrary line ($i = 4$) of the 200 m DLS loop with LP-EZF and respectively $\kappa = 10$ dB and $\kappa = 20$ dB.

Tables 8.6 and 8.7 give the resulting average information bitrates for all combinations of precoder and bitloading algorithm. We observe that the combination NLP-EZF with RS(255,239) achieves the highest bitrate, but only slightly better than NLP-CNS with RS(255,239) at the cost of a considerably higher computational complexity. The information rate for NLP-EZF with

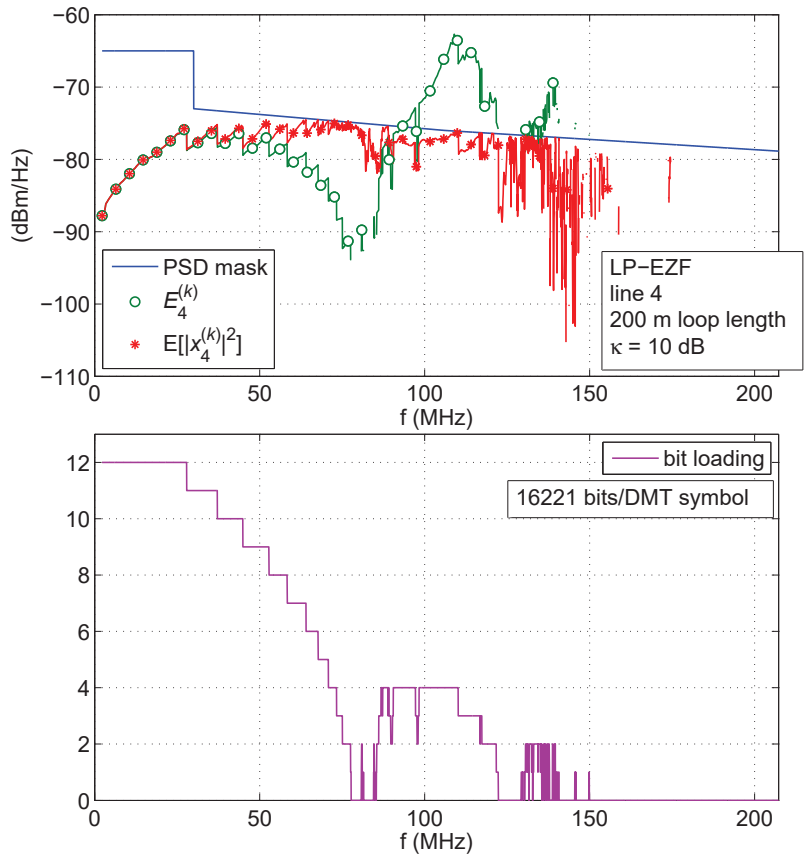

Figure 8.4: PSD values and bitloading for line 4 of the 200 m loop with $\kappa = 10$ dB, RS(255,239) + byte-interleaver + TCM and LP-EZF.

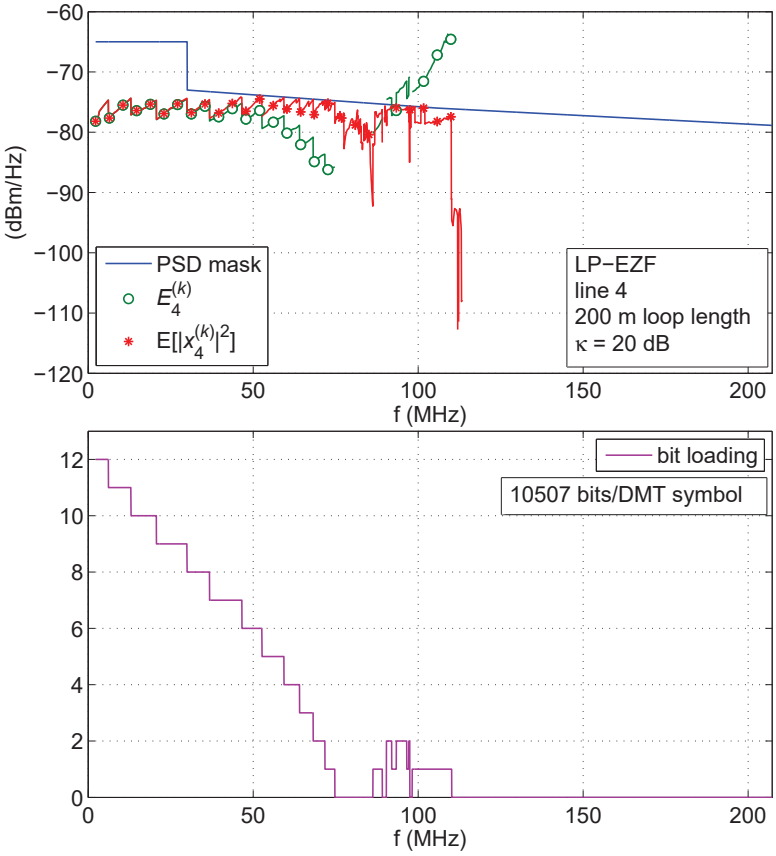

Figure 8.5: PSD values and bitloading for line 4 of the 200 m loop with $\kappa = 20$ dB, RS(255,239) + byte-interleaver + TCM and LP-EZF.

Table 8.5: Average information bitrates (Gbps) per line for all proposed LDPC codes with $K = 4320$ and NLP-CNS and loop lengths equal to 100 m, 150 m, 200 m and 250 m.

	R_c	1/2	2/3	5/6	16/18	20/21
$\kappa = 10$ dB	100 m	0.997	1.243	1.449	1.484	**1.486**
	150 m	0.792	0.962	1.079	**1.093**	1.082
	200 m	0.548	0.664	0.738	**0.743**	0.735
	250 m	0.410	0.494	0.558	**0.564**	0.555
$\kappa = 20$ dB	100 m	0.795	0.922	**0.993**	0.986	0.952
	150 m	0.566	0.644	**0.682**	0.677	0.654
	200 m	0.390	0.448	**0.485**	0.482	0.467
	250 m	0.290	0.336	**0.361**	0.357	0.345

RS(255,239) is only 1.1% and 2.7% larger than for NLP-CNS, for $\kappa = 10$ dB and $\kappa = 20$ dB, respectively.

For the difference between LP and NLP with RS(255,239); we observe that NLP outperforms LP by about 15% and 6.4% for CNS and EZF, respectively.

Similar as for the DSL channel without IN, RS(136,120) achieves a lower average information bitrate for all loop lenghts and levels of IN. The average information bitrate for RS(255,239) is 5.3% larger than for RS(136,120). Although the RS(136,120) code allows to load more coded bits on the tones, compared to RS(255,239), the larger parity overhead of the former gives rise to a smaller information bitrate.

8.5 Information Rate Comparison

In Figure 8.6, we compare the average information bitrate for all the protection strategies considered in this chapter so far, for NLP-EZF and respectively no IN, $\kappa = 10$ dB and $\kappa = 20$ dB. As far as the LDPC codes is concerned, we only show the results for the rate 16/18 code ($\kappa = 10$ dB) and rate 5/6 code ($\kappa = 20$ dB), which performs (nearly) the best for the considered IN levels. This conclusion can be generalized over all combinations of precoder and bitloading algorithm.

Compared to the case where IN is absent (see Figure 7.11), we observe that the presence of IN causes a considerable reduction of the information bitrate,

Table 8.6: Average information bitrates (Gbps) per line for RS + byte-interleaving + TCM and IN with $\kappa = 10$ dB.

		100m	150m	200m	250m
RS(255,239)	LP-EZF	1.338	0.993	0.677	0.513
	LP-CNS	1.310	0.848	0.666	0.502
	NLP-EZF	1.454	1.064	0.728	0.549
	NLP-CNS	1.442	1.051	0.718	0.543
RS(136,120)	LP-EZF	1.269	0.943	0.642	0.487
	LP-CNS	1.241	0.804	0.632	0.477
	NLP-EZF	1.378	1.010	0.691	0.522
	NLP-CNS	1.367	0.998	0.683	0.516

Table 8.7: Average information bitrates (Gbps) per line for RS + byte-interleaving + TCM and IN with $\kappa = 20$ dB.

		100m	150m	200m	250m
RS(255,239)	LP-EZF	0.889	0.634	0.448	0.339
	LP-CNS	0.800	0.568	0.415	0.272
	NLP-EZF	0.959	0.673	0.467	0.348
	NLP-CNS	0.941	0.644	0.458	0.340
RS(136,120)	LP-EZF	0.846	0.604	0.426	0.323
	LP-CNS	0.760	0.540	0.393	0.258
	NLP-EZF	0.912	0.640	0.445	0.331
	NLP-CNS	0.895	0.610	0.436	0.323

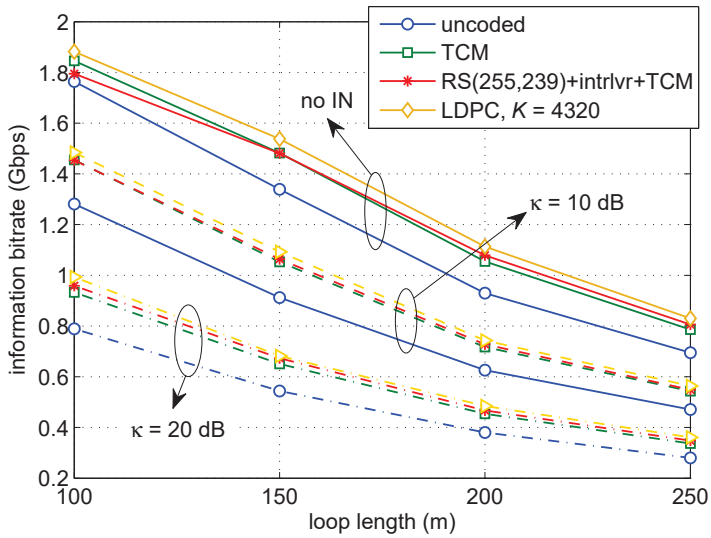

Figure 8.6: Comparison of the average information bitrate versus loop length for EZF-NLP, several protection strategies and respectively no IN, $\kappa = 10$ dB and $\kappa = 20$ dB.

under the constraint that the BER does not exceed 10^{-7}. This is because the presence of IN imposes the use of smaller constellations in order to meet the BER constraint.

8.6 Unknown Parameters of IN Model

When the parameters of the IN model are unknown, we perform the bitloading for the different protection strategies using the SNR tables which are valid in the *absence* of IN (i.e., the actual SNR is considered to be equal to $\mathrm{SNR}_{\mathrm{avg}}$), but with a positive offset Δ (in dB) added to each SNR entry; this corresponds to shifting the BER curves in the absence of IN to the right by Δ dB. This offset serves as a margin against IN, and will be selected such that the bitloading results in a BER (averaged over the DMT symbols with and without IN and over all lines) which equals the target BER. In a practical setting, the offset value Δ could be adapted according to observed error performance statistics.

As a few examples, we consider uncoded transmission with LP-CNS, TCM with LP-CNS, and the concatenation (with infinite byte interleaving) of RS(255, 239) + TCM with NLP-EZF. Figures 8.7, 8.8 and 8.9 show the corresponding BER (average over the DMT symbols with and without IN) versus Δ for each

of the 8 lines with DSL loop length 200 m for both $\kappa = 10$ dB and $\kappa = 20$ dB. We observe that for a given coding scheme, all lines experience virtually the same BER. The SNR offsets which yield the target BER of 10^{-7} on all lines for $\kappa = 10$ dB and $\kappa = 20$ dB are $\Delta = 8.0$ dB and $\Delta = 13.0$ dB for uncoded transmission, $\Delta = 8.5$ dB and $\Delta = 13.5$ dB for TCM, and $\Delta = 9.2$ dB and $\Delta = 14.2$ dB for RS(255, 239) + TCM. When comparing the SNR tables for the case of no IN (i.e., Tables 5.1, 5.2 and 5.12) with those for $\kappa = 10$ dB (i.e., Tables 5.6, 5.8 and 5.13) and $\kappa = 20$ dB (i.e., Tables 5.6, 5.9 and 5.14), we observe that the differences in SNR for a given coding scheme and precoder type are very close to the offset value mentioned above, irrespective of the considered constellation; more specifically, for $\kappa = 10$ dB and $\kappa = 20$ dB, the mean and standard deviation of these SNR differences are (8.0 dB, 0.05 dB) and (13.0 dB, 0.05 dB) for uncoded transmission with LP, (8.5 dB, 0.08 dB) and (13.5 dB, 0.05 dB) for TCM with LP, and (9.2 dB, 0.05 dB) and (14.2 dB, 0.05 dB) for RS(255, 239) + TCM with NLP. Hence, when applying the offset mentioned above to the SNR values from the table pertaining to no IN, we obtain SNR values which are very close (within about a tenth of a dB) to the SNR values from the tables for $\kappa = 10$ dB and $\kappa = 20$ dB. As a consequence, the bitloadings resulting from the table pertaining to no IN (after applying the proper offset Δ) yield the same information bitrates as in the case where the parameters of the IN model are known. We have verified that the same observation holds for all combinations of bitloading algorithms and precoder types.

A similar observation holds for LDPC codes. Comparing the SNR values from Table 5.11 (valid for the rate 20/21 LDPC code) for no IN with those for $\kappa = 10$ dB and $\kappa = 20$ dB, we observe that the mean and standard deviation of the differences in SNR are (9.1 dB, 0.04 dB) and (14.1 dB, 0.04 dB), respectively, in the case of LP. This indicates that the bitloading using the SNR values for no IN with an offset of $\Delta = 9.1$ dB (for $\kappa = 10$ dB) or $\Delta = 14.1$ dB (for $\kappa = 20$ dB) gives rise to the corresponding information bitrates from Table 8.5. This observation also holds for other LDPC codes and other combinations of precoder type and bitloading algorithm.

The BER averaged over the DMT symbols with and without IN is upper bounded by the BER related to the DMT symbols with IN. As the instantaneous SNR for the latter DMT symbols equals $\mathrm{SNR}_{\mathrm{avg}} \frac{1+p_1\kappa}{1+\kappa}$, the SNR difference between the SNR values in the table for no IN and those for a given κ are upper bounded by $10 \log \left(\frac{1+\kappa}{1+p_1\kappa} \right)$, which amounts to 9.4 dB and 14.5 dB for $\kappa = 10$ dB and $\kappa = 20$ dB, respectively. Compared to uncoded transmission and TCM, the offset values for the LDPC codes and the RS(255, 239) + TCM concatenation are closer to these upper bounds, because the latter coding schemes yield the steeper BER curves.

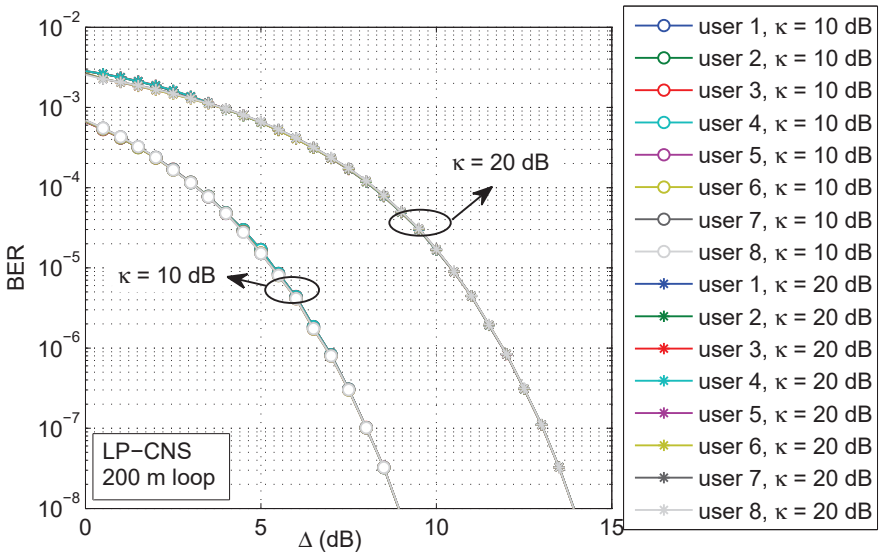

Figure 8.7: BER versus offset value Δ for 8 lines with DSL loop length 200 m, uncoded transmission and LP-CNS for both $\kappa = 10$ dB and $\kappa = 20$ dB.

8.7 ARQ

When the parameters of the IN model are known, we follow the same approach as in the absence of IN, outlined in Section 7.6. For line i, the residual BER (averaged over the DMT symbols with and without IN) after N_{retr} retransmissions (denoted $\text{BER}_i^{(N_{\text{retr}})}$) and the goodput (denoted GP_i) are the same functions of $\text{WER}_i^{(0)}$ and $\text{BER}^{(0)}$ as when IN is absent, with $\text{WER}_i^{(0)}$ and $\text{BER}^{(0)}$ denoting the BER and WER (both averaged over the DMT symbols with and without IN) in the absence of ARQ. Averaging $\text{BER}_i^{(N_{\text{retr}})}$, $\text{WER}_i^{(0)}$ and GP_i over all lines yields $\overline{\text{BER}}^{(N_{\text{retr}})}$, $\overline{\text{WER}}^{(0)}$ and $\overline{\text{GP}}$, respectively.

We first consider as an example the rate 20/21 LDPC code with $K = 4320$. Figure 8.10 shows as function of R_{b} the corresponding $\text{WER}^{(0)}$ and the $\text{BER}^{(0)}$ associated with $N_{\text{retr}} = 0$, in the cases of no IN and IN with $\kappa = 20$ dB. Clearly, the WER and the BER increase with IN level κ. In the absence of IN, the BER and WER curves are quite steep; whereas in the presence of IN they have a rather flat portion over a range of R_{b} that corresponds to large constellation sizes yielding a likely decoding error when a DMT symbol is occasionally hit by IN. Figure 8.11 shows $\overline{R}_{\text{b}}$ and the average goodput $\overline{\text{GP}}$ as a function of $\overline{R}_{\text{b}}$, for no IN and IN with $\kappa = 20$ dB. We see that, for given $\overline{R}_{\text{b}}$, the goodput is only moderately affected by the IN level.

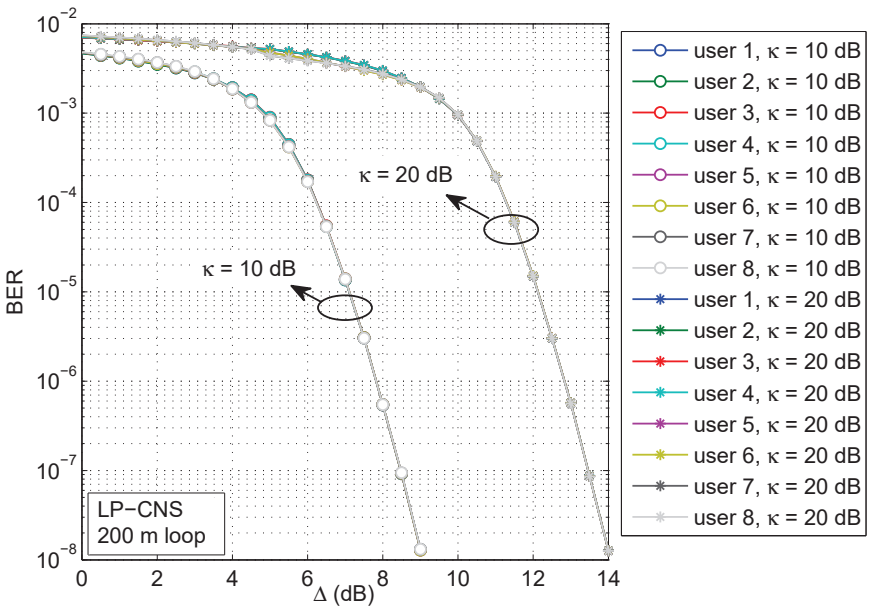

Figure 8.8: $\mathrm{BER}_{\mathrm{avg}}$ versus offset value Δ for 8 lines with DSL loop length 200 m, TCM and LP-CNS for both $\kappa = 10$ dB and $\kappa = 20$ dB.

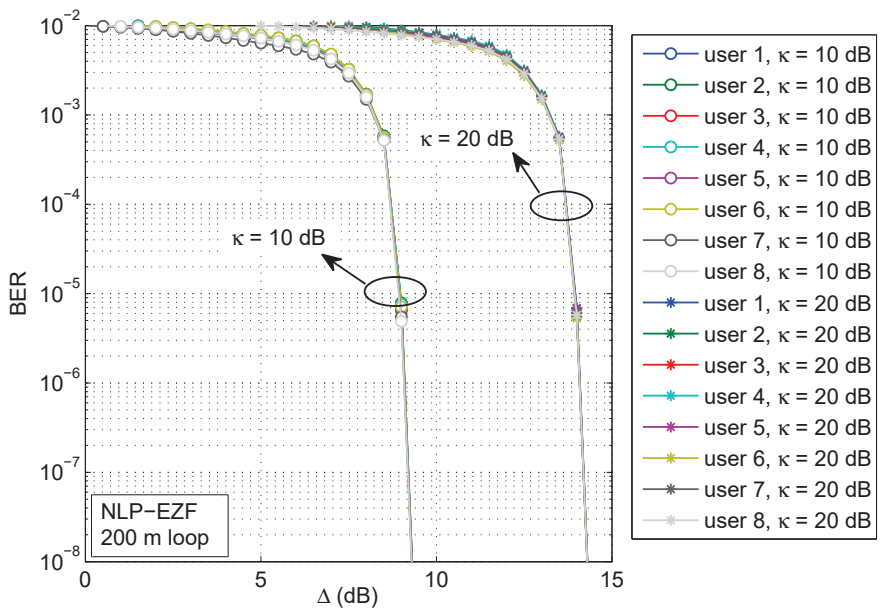

Figure 8.9: $\mathrm{BER}_{\mathrm{avg}}$ versus offset value Δ for 8 lines with DSL loop length 200 m, RS(255,239) + byte-interleaver + TCM and NLP-EZF for both $\kappa = 10$ dB and $\kappa = 20$ dB.

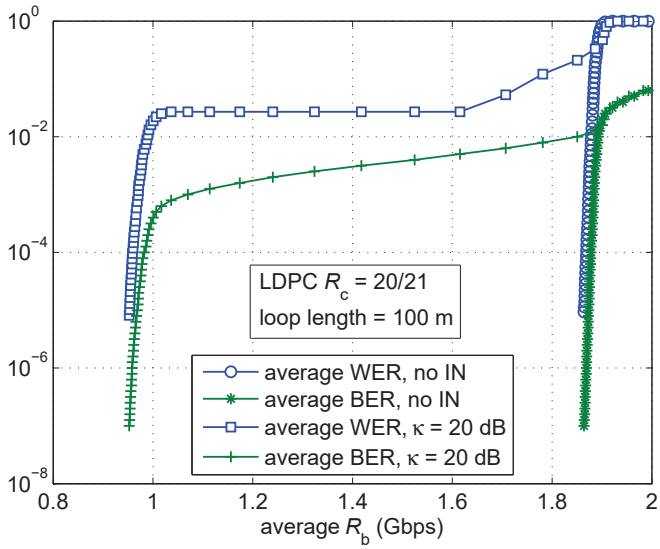

Figure 8.10: $\overline{\mathrm{WER}}^{(0)}$ and $\overline{\mathrm{BER}}^{(0)}$ versus $\overline{R}_\mathrm{b}$ for the rate 20/21 LDPC code with $K = 4320$ and $N_\mathrm{retr} = 0$.

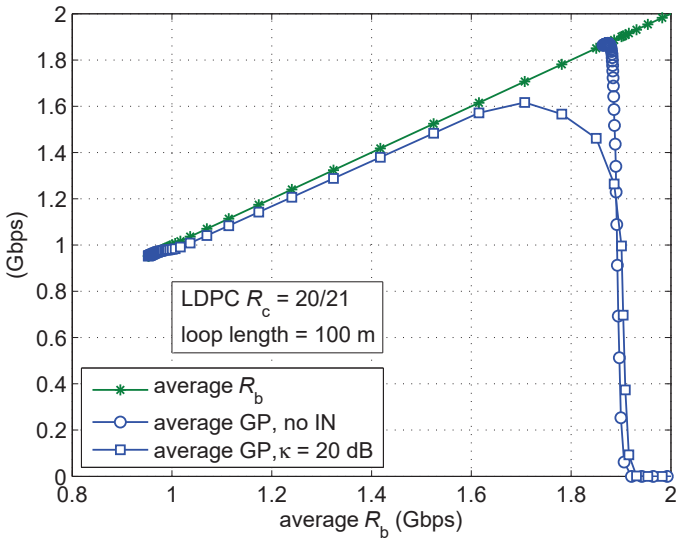

Figure 8.11: $\overline{R}_\mathrm{b}$ and $\overline{\mathrm{GP}}$ versus $\overline{R}_\mathrm{b}$ for the rate 20/21 LDPC code with $K = 4320$.

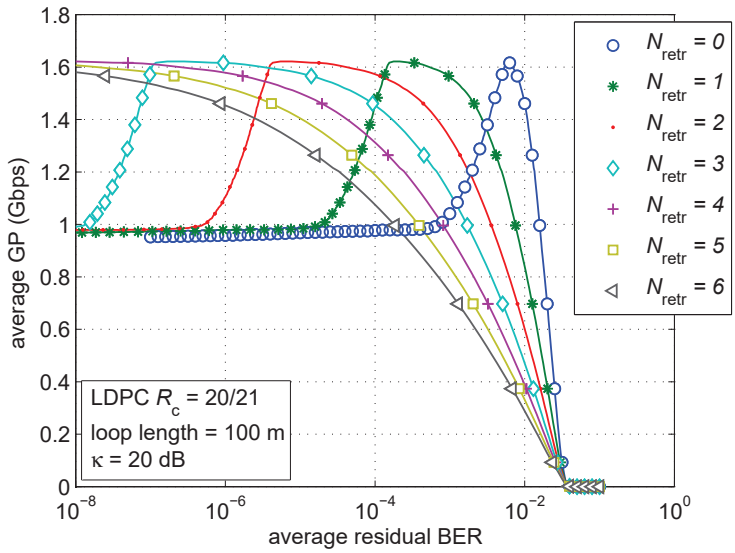

Figure 8.12: Average goodput $\overline{\mathrm{GP}}$ versus $\overline{\mathrm{BER}}^{(N_{\mathrm{retr}})}$ for the rate 20/21 LDPC code with $K = 4320$ and various N_{retr} .

The average goodput $\overline{\mathrm{GP}}$ as a function of the average residual BER, $\overline{\mathrm{BER}}^{(N_{\mathrm{retr}})}$, for various N_{retr} is shown in Figure 8.12 with IN ($\kappa = 20$ dB). Comparing Figure 7.12 (no IN) with Figure 8.12 ($\kappa = 20$ dB), we see that in the absence of IN the maxima of the goodput for the various N_{retr} are broader (because of the steeper curves of $\mathrm{WER}^{(0)}$ versus R_{b}) and wider apart. In the presence of IN, we need $N_{\mathrm{retr,opt}} = 4$ for the considered example to achieve the maximum average goodput $\overline{\mathrm{GP}}_{\mathrm{opt}}$ under the constraint $\mathrm{BER}^{(N_{\mathrm{retr}})} \leq 10^{-7}$.

As discussed in Section 5.6, the latency constraint limits the maximum number of retransmissions to $N_{\mathrm{retr,max}} = 6$. For the considered example, the latency constraint does not restrict the average goodput, as $N_{\mathrm{retr,opt}} \leq N_{\mathrm{retr,max}}$.

For the different error control strategies and using NLP-CNS, Tables 8.8 and 8.9 show $\overline{\mathrm{GP}}$ under the error performance constraint $\mathrm{BER} \leq 10^{-7}$, for $\kappa = 10$ dB and $\kappa = 20$ dB, considering the cases of ARQ and no ARQ. For uncoded transmission and TCM, the information word consists of 4320 bits and the LDPC codes have $K = 4320$ information bits; for the concatenation of the RS outer code, the infinite byte interleaver and TCM, only the traditional RS(255, 239) code is considered. When using ARQ, we distinguish between the cases where there is no constraint on N_{retr}, and where N_{retr} has to satisfy the latency constraint $N_{\mathrm{retr}} \leq N_{\mathrm{retr,max}}$, with $N_{\mathrm{retr,max}} = 6$ (see discussion on latency with ARQ in Section 5.6.2). When in the case of ARQ one can achieve, for a specific

Table 8.8: Maximum achievable goodput (Gbps) for BER $\leq 10^{-7}$ for NLP-CNS and IN with $\kappa = 10$ dB.

		100 m	150 m	200 m	250 m
$\kappa = 10$ dB	uncoded, no ARQ	1.264	0.885	0.613	0.460
	uncoded, ARQ	1.684 (3)	1.267 (3)	0.872 (3)	0.657 (3)
	TCM, no ARQ	1.435	1.034	0.702	0.532
	TCM, ARQ	1.722 (4)	1.332 (4)	0.920 (4)	0.685 (4)
	LDPC 1/2, no ARQ	0.997	0.792	0.548	0.410
	LDPC 1/2, ARQ	1.070 (4)	0.903 (4)	0.679 (5)	0.508 (5)
	LDPC 2/3, no ARQ	1.243	0.962	0.664	0.494
	LDPC 2/3, ARQ	1.374 (4)	1.134 (4)	0.822 (5)	0.618 (6)
	LDPC 5/6, no ARQ	1.449	1.079	0.738	0.558
	LDPC 5/6, ARQ	1.630 (4)	1.317 (4)	0.923 (4)	0.688 (4)
	LDPC 16/18, no ARQ	1.484	*1.093*	*0.743*	*0.564*
	LDPC 16/18, ARQ	1.706 (4)	1.359 (4)	0.955 (4)	0.711 (4)
	LDPC 20/21, no ARQ	*1.486*	1.082	0.735	0.555
	LDPC 20/21, ARQ	**1.768** (4)	**1.388** (4)	**0.970** (4)	**0.722** (4)
	RS+TCM, no ARQ	1.442	1.051	0.718	0.543

Table 8.9: Maximum achievable goodput (Gbps) for BER $\leq 10^{-7}$ for NLP-CNS and IN with $\kappa = 20$ dB.

		100 m	150 m	200 m	250 m
$\kappa = 20$ dB	uncoded, no ARQ	0.773	0.513	0.370	0.270
	uncoded, ARQ	1.562 (4)	1.123 (4)	0.772 (4)	0.575 (4)
	TCM, no ARQ	0.898	0.616	0.441	0.324
	TCM, ARQ	1.568 (4)	1.162 (4)	0.783 (4)	0.588 (4)
	LDPC 1/2, no ARQ	0.795	0.566	0.390	0.290
	LDPC 1/2, ARQ	1.052 (4)	0.903 (5)	0.701 (5)	0.527 (5)
	LDPC 2/3, no ARQ	0.922	0.644	0.448	0.336
	LDPC 2/3, ARQ	1.322 (4)	1.107 (6)	**0.842** (6)	**0.635** (6)
	LDPC 5/6, no ARQ	*0.993*	*0.681*	*0.485*	*0.361*
	LDPC 5/6, ARQ	1.518 (4)	1.184 (5)	<u>0.847</u> (8) 0.795 (6)	<u>0.637</u> (8) 0.595 (6)
	LDPC 16/18, no ARQ	0.986	0.677	0.482	0.357
	LDPC 16/18, ARQ	1.573 (4)	1.204 (5)	0.836 (9) 0.801 (6)	0.628 (9) 0.606 (6)
	LDPC 20/21, no ARQ	0.952	0.654	0.467	0.345
	LDPC 20/21, ARQ	<u>**1.616**</u> (4)	<u>**1.208**</u> (4)	0.810 (4)	0.611 (4)
	RS+TCM, no ARQ	0.941	0.644	0.458	0.340

code, $\overline{\text{GP}} = \overline{\text{GP}}_{\text{opt}}$ with $N_{\text{retr,opt}} \leq 6$, the corresponding entry shows $\overline{\text{GP}}_{\text{opt}}$ along with the value of $N_{\text{retr,opt}}$. When in the case of ARQ one cannot achieve $\overline{\text{GP}} = \overline{\text{GP}}_{\text{opt}}$ with $N_{\text{retr,opt}} \leq 6$ for a specific code, the entry shows (i) $\overline{\text{GP}}_{\text{opt}}$ along with the value of $N_{\text{retr,opt}}$ (which is larger than 6); and (ii) the maximum value of $\overline{\text{GP}}$ (which is less than $\overline{\text{GP}}_{\text{opt}}$) that can be achieved under the latency constraint $N_{\text{retr}} \leq 6$, along with the corresponding N_{retr} (which equals 6). For each loop length, we indicated in the case of ARQ which code yields the largest $\overline{\text{GP}}$, when the latency constraint applies (marked in bold) or when the latency constraint does not apply (marked with underlining); when no ARQ is used, the numbers in italic indicate which codes achieve the largest $\overline{\text{GP}}$, for the considered loop lengths. We make the following observations:

- Comparing Tables 7.9-7.12 (no IN) with Tables 8.8-8.9, we see that when ARQ is used, the amount of IN has a rather small effect (less than about 10%) on $\overline{\text{GP}}_{\text{opt}}$, but more retransmissions than in the absence of IN are needed to achieve $\overline{\text{GP}}_{\text{opt}}$. However, when no ARQ is used, the achieved $\overline{\text{GP}}$ is considerably smaller than $\overline{\text{GP}}_{\text{opt}}$ when IN is present, and strongly dependent on the IN level. This indicates that ARQ provides considerable robustness against IN, at the expense of latency caused by retransmissions.

- Compared to ARQ with proper coding, the traditional concatenation of a RS(255, 239) outer code with a TCM inner code (separated by an infinite byte interleaver) offers much less protection against IN.

- In the absence of ARQ, (close to) optimum codes are the rate 16/18 and rate 5/6 LDPC codes, for $\kappa = 10$ dB and $\kappa = 20$ dB, respectively.

- In the presence of ARQ with a latency constraint $N_{\text{retr}} \leq 6$, Figure 8.13 compares the $\overline{\text{GP}}$ for all the protection strategies considered and respectively no IN, $\kappa = 10$ dB and $\kappa = 20$ dB. As far as LDPC is concerned, we only show the results for the code rate which performs best for the considered IN level and loop length. The rate 20/21 LDPC code (4 retransmissions) is optimum for $\kappa = 10$ dB; for $\kappa = 20$ dB, the optimum codes are the rate 20/21 LDPC code (4 retransmissions) for the 100 m and 150 m loops, and the rate 2/3 LDPC code (6 retransmission) for the 200 m and 250 m loops, but on these longer loops the rate 20/21 LDPC code (4 retransmissions) is only slightly worse than the latter code. No substantial increase of $\overline{\text{GP}}$ is achieved when removing the latency constraint.

When the parameters of the IN model are not known, one can perform the bitloading using the SNR tables for the case without IN, with an offset added to the SNR values. When ARQ is used, the offset can be selected such that the average goodput is maximized; the number of allowed retransmissions can be selected such that the average residual BER does not exceed the target BER of 10^{-7}. When no ARQ is used, the offset can be selected such that the average BER equals the target BER of 10^{-7}.

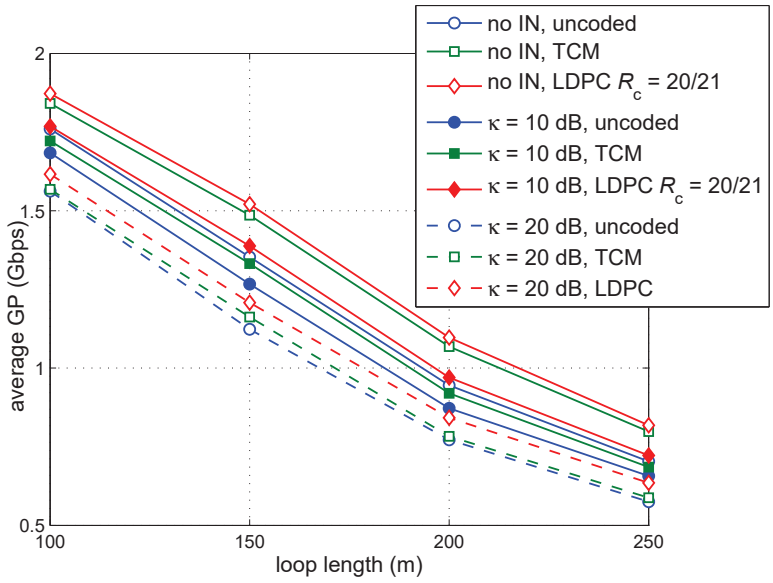

Figure 8.13: Comparison of $\overline{\mathrm{GP}}$ versus loop length for CNS-NLP, several protection strategies and respectively no IN, $\kappa = 10$ dB and $\kappa = 20$ dB.

Part II

Video Transmission over Wireless Channel

9

Application Layer ARQ for Protecting Video Packets over an Indoor MIMO-OFDM Link with Correlated Block Fading

Published in IEEE Journal on Selected Areas in Communications.

The quality of experience (QoE) of IP-packetized streaming video is affected by both packet loss and packet delay variations. When the network delivering the video content contains a wireless link, occasional deep fades give rise to bursts of packet losses. In order to maintain a sufficient video QoE at the end user, video packets must be protected against losses by means of a suitable form of error control. In this contribution, we consider an indoor radio MIMO-OFDM transceiver operating over a Rayleigh block-fading channel with arbitrary corre-

lation in the time and frequency dimensions, which makes use of an application layer Automatic Repeat reQuest (ARQ) protocol to provide additional protection of the video content against packet loss. We analyze the resulting residual packet loss performance, under a latency constraint imposed by the requirement of a small TV channel switching delay. This analysis makes direct use of the fading characterization (correlation functions in time and frequency dimensions) of the indoor environment, rather than relying on a Markov model that only approximately describes the packet loss process. Numerical results are obtained by Monte Carlo integration combined with an efficient importance sampling technique devised for the problem at hand. Assuming a 2.4 GHz wireless link, we point out how to select the system parameters (number of antennas, number of retransmissions) in order to achieve a residual packet loss performance yielding a satisfactory QoE for HDTV transmission.

9.1 Introduction

The Internet Protocol (IP) allows the provision of a mix of multimedia services (video, audio, voice, data, gaming, etc.) to an end user, by breaking up the bit streams generated by the various services into IP packets and sending these packets over the network. In this part of the dissertation, we consider the delivery of these multimedia services via a wireless channel, and focus on the reliability of the received video data.

Wireless channels are frequency-selective and subject to time-varying fading. These impairments distort the transmitted signal, and give rise to bursts of bit errors at the receiver when deep fades occur within the signal bandwidth. IP packets affected by bit errors are erased at the receiver, yielding lost packets at the destination. In order to achieve a sufficient QoE for (high-quality) video services, judicious system design should protect the video stream against packet loss.

The distortion caused by the frequency-selectivity of the channel can be dealt with by resorting to multicarrier modulation (OFDM) [57], which turns the frequency-selective channel into a number of parallel frequency-flat channels. The OFDM signal format has been included in various standards for both wireless and wired communications such as ADSL, DVB-T, WiFi and WiMax [58].

Several authors have considered the protection of the video stream against packet loss by resorting to FEC, to ARQ protocols, or to combinations thereof (e.g., [59–70]). Most often, in their analysis the packet loss process is described by a simplified Markov model, which makes abstraction of the fine details of the underlying phyical processes (fading, noise,...) that cause transmission errors, and, therefore, only approximately captures the packet loss process.

Here, we consider the transmission of a video stream over an indoor radio channel. The transmitter and receiver are equipped with multiple antennas,

which gives rise to a multiple-input multiple-output (MIMO) wireless channel. The diversity offered by the multiple spatial channels between transmitter and receiver increases the robustness against fading, e.g., by using space-time block codes on the physical (PHY) layer [17–19]. The frequency-selectivity of the channel is dealt with by adopting the OFDM signal format. The channel is described as Rayleigh block fading, with correlation in both the time and frequency dimensions. Additional protection (to the video stream only) is provided by means of selective-repeat ARQ (SR-ARQ) [71, 72], with the number of retransmissions being restricted by a latency constraint. In order to reduce the cost of the wireless transceiver, the Digital Subscriber Line Access Multiplexer (DSLAM) operates as the retransmitting node. The resulting round-trip time between the DSLAM and the wireless receiver is sufficiently small for the SR-ARQ to be effective; the resulting transmission overhead is considerably smaller than for FEC, under the same latency constraint [59, 61].

In [59], we have investigated to what extent the combination of the RS code or the SR ARQ protocol with the space-time PHY layer code improves the reliability of the video transmission over a wireless channel subject to Rayleigh fading with frequency-flat transmission channel and independent block fading. We have pointed out that SR-ARQ and RS erasure coding give rise to a diversity gain yielding improved error performance, and have presented simple analytical expressions for this gain. Both SR-ARQ and RS erasure coding yield the same maximum possible diversity gain. However, when using RS erasure coding this maximum diversity gain cannot be achieved because of practical limitations on the allowed transmission overhead. The performance analysis assumes that the channel state is the same for all OFDM subcarriers. This assumption is valid when the signal bandwidth does not exceed the 90% coherence bandwidth of the channel. For the considered 60 GHz indoor radio channel under NLOS conditions, the 90% coherence bandwidth is about 6 MHz [73], so that our analysis is valid for bitrates up to 12 Mbit/s (assuming QPSK transmission). When the signal bandwidth is larger than the 90% coherence bandwidth, different subcarriers experience different channel states (which could be exploited to increase the PHY layer diversity by means of frequency-interleaving and coding across the subcarriers of an OFDM block). In this chapter, we use a more general fading model with correlation in both frequency and time dimensions over a 2.4 GHz indoor wireless link and we examine the effect of application layer ARQ.

This chapter is organized as follows. In Section 9.2, we provide a system description involving the packetization of compressed video, the PHY layer of the MIMO-OFDM system, the Rayleigh fading channel model, and the SR-ARQ protocol that protects only the video traffic against packet loss. We provide in Section 9.3 the packet error performance analysis for various scenarios, for various combinations of the number of transmit and receive antennas, with or without additional video packet protection from SR-ARQ, under the assumption of correlated block fading in the time and frequency dimensions. Rather

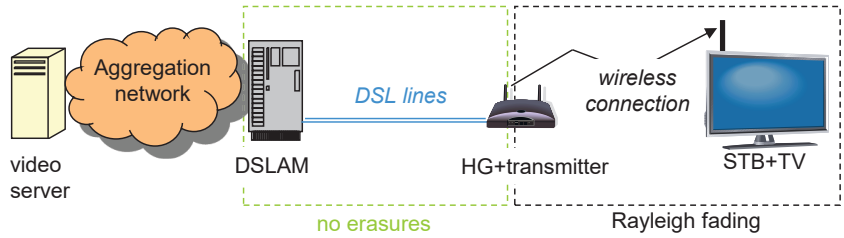

Figure 9.1: Concatenation of DSL connection and wireless connection.

than resorting to a simplified Markov model for describing the packet loss, we make in our analysis direct use of the fading correlation function in the time and frequency dimensions. This analysis is a generalization of [59], where a frequency-flat transmission channel with independent block fading was considered. In Section 9.4, we present numerical results, related to High Definition TV (HDTV) [74] transmission over a 2.4 GHz indoor wireless link; these results have been obtained by means of an importance sampling technique [75–77], that we devised for the problem at hand. Finally, in Section 9.5 conclusions are drawn regarding system performance and complexity, and some remarks are formulated regarding the use, in the considered system, of additional video protection by means of FEC rather than SR-ARQ. A major conclusion is that SR-ARQ yields diversity gain, which is limited by the ratio of the allowed latency and the time interval between retransmissions, at the expense of only a very small transmission overhead.

9.2 System Description

We consider the case where video content is sent from the video server to the end user, as shown in Figure 9.1. A source, the video server, broadcasts the video data. Via an aggregation network, this video data reaches a DSLAM. The DSLAM sends the data related to a mix of services (video, audio, voice, data, gaming, etc.), over a Digital Subscriber Line (DSL) [78] to the user Home Gateway (HG). From the HG, the video data is sent through a wireless LAN to the Set Top Box (STB).

The video stream is encoded (compressed) by exploiting the temporal and spatial redundancy that is present in uncompressed video frames [16, 79, 80]. The resulting Transport Stream (TS) consists of a sequence of MPEG-TS packets, each containing 188 bytes (including a 4-byte header). For transmission over IP networks, 7 MPEG-TS packets (together with header information) are encapsulated in an IP packet [81].

On the Medium Acces Control (MAC) sublayer of the data link layer, a

Cyclic Redundancy Check (CRC) is added. This CRC allows the detection of packets that are corrupted by transmission errors; corrupted packets are not forwarded to the network layer, but are discarded ('erased') and therefore considered as lost. We assume there are no data link layer retransmissions (because the adverse affect of fading is reduced by means of space-time channel coding on the physical layer). Considering the sizes of the payload (7 TS packets) and of the headers/trailers added by the various protocol layers, the MAC packets arriving at the HG have a size of 1374 byte (10992 bit).

As far as the physical (PHY) layer is concerned, we only consider the wireless link between the HG and the STB. On the PHY layer of the HG transmitter, the L bits to be sent for every data-link-layer packet are mapped onto an M-point signal constellation. The resulting M-ary data symbols are transmitted at a rate R_s (in symbols per second) over the wireless channel; hence the duration of a packet equals $L/(R_s \log_2(M))$. The transmission makes use of OFDM [57]. The sequence of data symbols at rate R_s is demultiplexed into N_t parallel symbol streams, each of rate R_s/N_t. These N_t symbol streams are modulated onto N_t distinct tones, that have a frequency separation of (slightly more than) R_s/N_t, and the sum of these modulated tones is transmitted. The transmitted signal can be viewed as a sequence of OFDM blocks, each having a duration of N_t/R_s, and containing N_t data symbols (i.e., one symbol on each of the N_t tones). The bandwidth occupied by the resulting transmitted signal is (slightly more than) R_s. The transmission of an L-bit packet involves $L/(N_\mathrm{t} \log_2(M))$ OFDM blocks. The OFDM modulation turns the frequency-selective fading channel into a set of N_t flat-fading parallel channels.

Each tone of the OFDM system is affected by slow frequency-flat Rayleigh fading, and the fading on different tones is correlated. We restrict our attention to the case where the channel coherence time and the channel coherence bandwidth are much larger than the packet duration $L/(R_s \log_2(M))$ and the tone spacing R_s/N_t, respectively. Therefore, we assume that

- the fading gain on a given tone is constant over the packet duration, and the fading gains are correlated from one packet to the next.

- there are N_F sets of $N_\mathrm{t}/N_\mathrm{F}$ consecutive tones, with all tones within a same set experiencing the same fading gain, and the fading gains being correlated from one set to the next.

This fading model corresponds to correlated block fading in both time and frequency.

On the PHY layer of the STB receiver, the M-ary data symbols are detected, and demapped to bits. On the MAC sublayer, the recovered bits are grouped into packets of size L, and error detection based on the CRC is performed. When an error is detected, the packet is erased; otherwise, the packet is passed to the higher layers.

Because of fading, some of the tone signals are occasionally strongly attenuated. To alleviate the damaging impact of fading on the detection of the M-ary data symbols, some form of channel coding is used on the PHY layer: the transmitter introduces some redundancy in the time, frequency and/or spatial dimensions, that is exploited at the receiver to correct transmission errors. In this chapter, we consider the use of multiple transmit and receive antennas. A MIMO system with N_{tr} transmit and N_{r} receive antennas allows the introduction of space-time coding as discussed in more detail in Section 2.4.

All multimedia services that make use of the wireless link benefit from the protection against fading, provided by the PHY layer diversity $N_{\mathrm{tr}}N_{\mathrm{r}}$. However, this protection might not be sufficient for achieving a sufficient QoE for video services. The relation between packet error performance and QoE is rather complicated: on one hand the video decoder has error concealment capabilities, but on the other hand the effect of a single packet loss might propagate; the interested reader is referred to [70, 82–86]. In this paper, we rely on the recommendation [87], which states that, for several video compression standards (i.e., MPEG-2, H.264/AVC and VC-1), IP packet loss should be limited to at most one error event per hour for Standard Definition TV (SDTV) and to one error event per four hours for HDTV, with an error event being defined as the loss of a small number of IP packets; for HDTV it is estimated that only one out of three error events will yield a visible impairment (because of the error concealment techniques), so that on average only one visual distorsion in 12 hours would result.

We consider in this paper additional protection by means of application layer retransmissions, applied exclusively to the video packets. We restrict our attention to SR-ARQ, which involves the retransmission (upon request from the STB receiver) of copies of only the lost packets, with the retransmitting node not waiting for acknowledgement before transmitting a next packet [71, 72].

The time interval T_{retr} between (re)transmission instants of the same packet is the sum of the packet duration $L/(R_s \log_2(M))$ and the round-trip delay T_{RTT} ; the latter is the sum of the two-way propagation delay, the duration of the acknowledgment message, and the processing delays at the receiver and the transmitter as given in (2.15). Since each retransmission gives rise to a latency of T_{retr}, the maximum latency introduced by the SR-ARQ protocol equals $N_{\mathrm{retr}}T_{\mathrm{retr}}$, with N_{retr} denoting the maximum number of retransmissions per packet. The latency caused by the SR-ARQ protocol contributes to the TV channel switching delay. In order to achieve acceptable values for this delay, the latency should be limited. Therefore, we select as the retransmitting node the DSLAM rather than the video source, because the former yields the smaller round-trip delay. Of course, the retransmitting network node needs sufficient intelligence, in order to handle retransmission requests related to specific video packets; in addition, this node must have a retransmission buffer containing video packets that have not yet been correctly received by the STB. Augment-

ing the functionality of the DSLAM increases its complexity and cost. One could also envisage to use the HG as retransmitting node. As the HG is a consumer product, the DSLAM appears to be the economically justified choice for operating as the retransmitting node. However, the HG would offer the shorter round-trip delay.

9.3 System Analysis

In this section, we present the analysis of the system under study. We derive the probability of unrecoverable packet loss, and investigate the burstiness of the unrecoverable packet losses. As a performance measure, we consider the average number of bursts of unrecoverable packet losses, over a reference time interval.

Assuming that the number $(L/\log_2(M))$ of symbols per packet is much larger than the number (N_t) of OFDM tones, for a MIMO-OFDM system using N_tr transmit and N_r receive antennas, there are $N_\mathrm{tr}N_\mathrm{r}N_\mathrm{F}$ fading gains involved in the transmission of single packet; N_F denotes the number of tone sets, with all $N_\mathrm{t}/N_\mathrm{F}$ tones in a given set experiencing the same fading gain. These $N_\mathrm{tr}N_\mathrm{r}N_\mathrm{F}$ fading gains determine the channel state vector $\mathbf{v}(l)$ of dimension N_F, related to the considered packet indexed by l. The mth component of $\mathbf{v}(l)$ is given by

$$\mathrm{v}_m(l) = \sum_{i=1}^{N_\mathrm{tr}} \sum_{j=1}^{N_\mathrm{r}} \left| h_m^{(i,j)}(l) \right|^2 \qquad m = 1, \ldots, N_\mathrm{F} \tag{9.1}$$

where $h_m^{(i,j)}(l)$ is the complex fading gain related to the mth tone set on the link between the ith transmit antenna and the jth receive antenna. The fading gains are normalized, such that $\mathrm{E}[|h_m^{(i,j)}(l)|^2] = 1$. Fading gains related to different pairs (i, j) are statistically independent. Each of the N_r received signals is affected by additive white Gaussian noise. The noise contributions at different antennas, on different tones and in different symbol intervals are statistically independent.

Considering SR-ARQ with a maximum of N_retr retransmissions, an unrecoverable packet loss occurs when the first transmission and all N_retr retransmissions of a packet are erased. Hence, an unrecoverable packet loss involves $N_\mathrm{retr} + 1$ channel state vectors, which we denote $\mathbf{v}(0), \ldots, \mathbf{v}(N_\mathrm{retr})$. For given channel state vectors, the probability of an unrecoverable packet loss is given by

$$P_\mathrm{unrec}(\mathbf{v}(0), \ldots, \mathbf{v}(N_\mathrm{retr})) = \prod_{l=0}^{N_\mathrm{retr}} P_\mathrm{p,e}(\mathbf{v}(l)) \tag{9.2}$$

where $P_\mathrm{p,e}(\mathbf{v}(l))$ is the probability that the lth copy of the considered packet is erased $(l = 0, \ldots, N_\mathrm{retr})$. The probability $P_\mathrm{p,e}(\mathbf{v}(l))$ equals the probability that

at least one data symbol from the packet is detected in error:

$$P_{\mathrm{p,e}}(\mathbf{v}(l)) = 1 - \prod_{m=1}^{N_{\mathrm{F}}} \left(1 - P_{\mathrm{s}}(\mathrm{v}_m(l))\right)^{K_{\mathrm{F}}} \tag{9.3}$$

where $P_{\mathrm{s}}(\mathrm{v}_m(l))$ is the probability that a symbol, transmitted on a tone belonging to the mth tone set, is detected in error, while $K_{\mathrm{F}} = L/(N_{\mathrm{F}} \log_2(M))$ denotes the number of symbols transmitted per tone set and per packet. The probability $P_{\mathrm{s}}(\mathrm{v}_m(l))$ depends on the symbol constellation and on the type of space-time coding. For a QPSK constellation one obtains

$$P_{\mathrm{p,e}}(\mathbf{v}(l)) = 1 - \prod_{m=1}^{N_{\mathrm{F}}} \left(1 - \mathrm{BER}(\mathrm{v}_m(l))\right)^{L/N_{\mathrm{F}}} \tag{9.4}$$

with $\mathrm{BER}(\mathrm{v}_m(l))$ denoting the bit error probability [17–20]:

$$\mathrm{BER}(\mathrm{v}_m(l)) = Q\left(\sqrt{\frac{2E_{\mathrm{b}}\eta \mathrm{v}_m(l)}{N_0}}\right) \tag{9.5}$$

In (9.5), E_{b} denotes the received energy per bit of the video packet; N_0 is the one-sided power spectral density of the noise at the receiver; $\eta = 1$ for uncoded transmission ($N_{\mathrm{tr}} = 1$) and $\eta = 1/2$ for Alamouti space-time coding ($N_{\mathrm{tr}} = 2$); $Q(u)$ is the complement of the cumulative distribution function of a zero-mean unit-variance Gaussian random variable.

The average probability of an unrecoverable packet error is obtained by averaging (9.2) over the joint distribution of the $N_{\mathrm{retr}} + 1$ channel state vectors $\mathbf{v}(0), \ldots, \mathbf{v}(N_{\mathrm{retr}})$:

$$P_{\mathrm{unrec,p}} = \mathrm{E}[P_{\mathrm{unrec}}(\mathbf{v}(0), \ldots, \mathbf{v}(N_{\mathrm{retr}}))] \tag{9.6}$$

As the channel state vectors are correlated, it appears impossible to obtain a closed-form expression for $P_{\mathrm{unrec,p}}$.

Although a closed-form expression for $P_{\mathrm{unrec,p}}$ is not available, we will now point out that $P_{\mathrm{unrec,p}}$ is proportional to $(E_{\mathrm{b}}/N_0)^{-D}$ at high E_{b}/N_0, and determine the value of D. Let us consider a bit error pattern containing N_{b} bit errors, that gives rise to an unrecoverable packet loss. With each erroneous bit are associated $N_{\mathrm{tr}}N_{\mathrm{r}}$ complex-valued Rayleigh fading gains. Stacking all $N_{\mathrm{b}}N_{\mathrm{tr}}N_{\mathrm{r}}$ fading gains related to the erroneous bits into a vector $\mathbf{h}$, we denote by r the rank of the autocorrelation matrix of $\mathbf{h}$. It can be shown that, for increasing E_{b}/N_0, the probability of occurrence of the considered bit error pattern, averaged over the fading gains, is proportional to $(E_{\mathrm{b}}/N_0)^{-d}$, with $d = N_{\mathrm{tr}}N_{\mathrm{r}}r$ [20]. The dominating bit error patterns yielding unrecoverable packet loss are those for which the rank r of the associated correlation matrix is minimum. As an unrecoverable packet loss implies that at least one bit in

each of the $N_{\text{retr}} + 1$ copies of the considered packet must be in error (and the corresponding fading gains are assumed to have a full rank correlation matrix), the minimum rank equals $N_{\text{retr}} + 1$. Hence, at high E_{b}/N_0, $P_{\text{unrec,p}}$ is proportional to $(E_{\text{b}}/N_0)^{-D}$, with $D = (N_{\text{retr}} + 1)N_{\text{tr}}N_{\text{r}}$ denoting the diversity order. Hence, the number of transmit antennas, the number of receive antennas and the number of retransmissions all contribute positively to the diversity order; therefore, increasing any of these quantities gives rise to a steeper negative slope of $P_{\text{unrec,p}}$ as a function of E_{b}/N_0. When no ARQ is used (i.e., $N_{\text{retr}} = 0$), the resulting diversity order equals the spatial diversity order $N_{\text{tr}}N_{\text{r}}$ offered by the MIMO system [17–19]. Hence, the introduction of SR-ARQ increases the diversity by a factor of $N_{\text{retr}} + 1$ as compared to the PHY layer diversity.

Because of the temporal correlation of the fading, the packet losses tend to occur in bursts: typically, relatively short intervals containing packet losses are followed by relatively long intervals without packet losses, and vice versa. Let us denote by n the number of lost packets in an interval of N inter-packet intervals; it is to be understood that a packet is lost when it has been erased during its transmission and its N_{retr} retransmissions. We consider the average number $E[n|n > 0]$ of packets in an interval of length N_{int} that contains at least one lost packet, which can be computed as

$$\mathrm{E}[n|n > 0] = \frac{\mathrm{E}[n]}{1 - \Pr[n = 0]} = \frac{N_{\text{int}} \cdot P_{\text{unrec,p}}}{1 - \Pr[n = 0]} \tag{9.7}$$

Clearly, $\mathrm{E}[n|n > 0]$ increases with N_{int}. However, when N_{int} ranges between (roughly) the average burst size and the average number of packets between bursts, an interval of size N_{int} that is affected by packet loss typically contains a single burst of lost packets, so that $\mathrm{E}[n|n > 0]$ is essentially constant within this range of N_{int} and equal to the average number N_{burst} of lost packets per burst. Hence, examination of $\mathrm{E}[n|n > 0]$ as a function of N_{int} reveals the value of N_{burst}. The average number of unrecoverable packet losses in a long reference interval containing N_{ref} inter-packet intervals equals $N_{\text{ref}}P_{\text{unrec,p}}$. The average number of bursts of unrecoverable packet losses in this interval of size N_{ref} then equals $N_{\text{ref}}P_{\text{unrec,p}}/N_{\text{burst}}$.

Let us investigate the average overhead $\mathrm{E}[\text{ovh}]$ related to the retransmission protocol. The average number $\mathrm{E}[\#\text{transm}]$ of transmissions per packet is related to the average overhead by $\mathrm{E}[\#\text{transm}] = 1 + \mathrm{E}[\text{ovh}]$. It is easily verified that

$$\mathrm{E}[\#\text{transm}] = 1 + \sum_{i=1}^{N_{\text{retr}}} \Pr[\#\text{transm} > i] \tag{9.8}$$

The quantity $\Pr[\#\text{transm} > i]$ can be expressed as

$$\Pr[\#\text{transm} > i] = \mathrm{E}\left[\prod_{l=0}^{i-1} P_{\text{p,e}}(\mathbf{v}(l))\right] \tag{9.9}$$

It follows from (9.2) that $\Pr[\#\text{transm} > i]$ in (9.9) can be interpreted as the average probability of unrecoverable packet loss for SR-ARQ with a maximum of $i-1$ retransmissions, which at high E_b/N_0 is proportional to $(E_\text{b}/N_0)^{-iN_\text{tr}N_\text{r}}$. Hence, the dominating contribution to $\text{E}[\text{ovh}] = \text{E}[\#\text{transm}] - 1$ is the term $\Pr[\#\text{transm} > 1] = \text{E}[P_{\text{p,e}}(\mathbf{v}(0))]$, which at high E_b/N_0 is proportional to $(E_\text{b}/N_0)^{-N_\text{tr}N_\text{r}}$. Therefore, E[ovh] is determined mainly by the PHY layer diversity.

9.4 Numerical Results

We consider the transmission of compressed HDTV according to the configuration shown in Figure 9.1. The compressed video bitrate equals 7.5 Mbps. The maximum allowed latency introduced by the SR-ARQ protocol is set at 200 ms. The error performance target is an average number of at most one unrecoverable packet burst per 4 hours [87]. The DSLAM acts as retransmitting node; the corresponding interval (T_retr) between retransmission instants is about 50 ms [88, 89], so the maximum number (N_retr) of retransmissions meeting the latency constraint is 4. The required retransmission buffer size is $T_\text{retr}N_\text{retr}R_\text{p}$ packets, which amounts to about 150 video packets or 1.5 Mbit when $N_\text{retr} = 4$.

The link between the HG and the STB is a 2.4 GHz indoor wireless connection [90] with a bandwidth of 20 MHz; assuming nonline-of-sight (NLOS) conditions, this connection is modeled as a Rayleigh fading channel, with a Doppler spread f_D equal to 8 Hz (corresponding to a speed of motion of about 1 m/s). We assume that the Doppler spectrum $S_\text{D}(f)$ has a Gaussian shape:

$$S_\text{D}(f) = \frac{1}{\sqrt{2\pi}\sigma_\text{F}} \exp\left(\frac{-f^2}{2\sigma_\text{F}^2}\right) \tag{9.10}$$

with $\sigma_\text{F} = f_\text{D}/4$ (so that $S_\text{D}(f_\text{D})$ is negligibly small as compared to $S_\text{D}(0)$). The resulting time-correlation $R_\text{D}(u)$ is the inverse Fourier transform of $S_\text{D}(f)$:

$$R_\text{D}(u) = \exp(-2\pi^2\sigma_\text{F}^2 u^2) \tag{9.11}$$

The power delay profile of the indoor channel is assumed to be exponentially decaying with a time constant τ of 30 ns. The correlation $R_\text{H}(\Delta F)$ between two channel transfer function values at f and $f + \Delta f$ is the Fourier transform of the power delay profile:

$$R_\text{H}(\Delta f) = \frac{1}{1 + j2\pi\tau\Delta f} \tag{9.12}$$

We split the channel bandwidth B_RF into N_F subbands of width B_RF/N_F. The correlation between channel transfer functions at the center and the edge of a subband equals $R_\text{H}(B_\text{RF}/(2N_\text{F}))$. We select N_F such that $|R_\text{H}(B_\text{RF}/(2N_\text{F}))| \geq 0.9$, so that all OFDM subcarriers in a subband experience essentially the same fading as the subcarrier at the center of the subband. For $\tau = 30$ ns and $B_\text{RF} =$

Table 9.1: Correlation values for the 2.4 GHz wireless link.

	$i=0$	$i=1$	$i=2$	$i=3$	$i=4$
$R_{\mathrm{D}}(iT_{\mathrm{retr}})$	1	0.821	0.454	0.169	0.042
$R_{\mathrm{H}}(iB_{\mathrm{RF}}/N_{\mathrm{F}})$	1	0.53-j0.499	0.22-j0.414	0.111-j0.314	NA

20 MHz, we get $N_{\mathrm{F}} = 4$ subbands. As the diversity order does not depend on the number of subbands, the value of N_{F} is not very critical.

Denoting by $h_m(l)$ the complex fading gain experienced by a carrier in the mth subband during the lth transmission of a packet ($m = 1, \ldots, N_{\mathrm{F}}$; $l = 0, \ldots, N_{\mathrm{retr}}$), we take

$$\mathrm{E}[h^*_{m_1}(l_1)h_{m_2}(l_2)] = R_{\mathrm{D}}\big((l_2 - l_1)T_{\mathrm{retr}}\big)R_{\mathrm{H}}\left((m_2 - m_1)\frac{B_{\mathrm{RF}}}{N_{\mathrm{F}}}\right) \quad (9.13)$$

It follows from (9.11) that $R_{\mathrm{D}}(iT_{\mathrm{retr}})$ can be expressed as ρ^{i^2}, with $\rho = \exp(-2\pi^2\sigma_{\mathrm{F}}^2 T_{\mathrm{retr}}^2)$. The factors from (9.13) corresponding to the channel parameters of the 2.4 GHz wireless link are given in Table 9.1; note that the temporal fading correlation between packets separated by T_{retr} is given by $\rho = 0.821$. The entries for negative i can be easily derived from Table 9.1, taking into account that $R_{\mathrm{D}}(.)$ is even-symmetric, and $R_{\mathrm{H}}(.)$ has complex-conjugate symmetry.

As there is no closed-form expression for $P_{\mathrm{unrec,p}}$, we have to obtain $P_{\mathrm{unrec,p}}$ by other means. A straightforward error-counting brute force simulation would require excessively long simulation times, especially for the very low values of $P_{\mathrm{unrec,p}}$ that are required to meet the QoE for HDTV. Therefore, we will obtain $P_{\mathrm{unrec,p}}$ by evaluating the expectation in (9.6) by means of Monte-Carlo (MC) integration [75–77]. Conventional MC integration evaluates $P_{\mathrm{unrec}}(\mathbf{v}(0), \ldots, \mathbf{v}(N_{\mathrm{retr}}))$ as the arithmetical average of N_{MC} independent realizations of the set $\{\mathbf{v}(0), ..., \mathbf{v}(N_{\mathrm{retr}})\}$ according to the actual joint distribution of the involved $(N_{\mathrm{retr}} + 1)N_{\mathrm{F}}N_{\mathrm{r}}N_{\mathrm{tr}}$ complex fading gains; accuracy improves with increasing N_{MC}. Better accuracy can be obtained by combining MC integration with importance sampling (IS), which involves using a properly biased joint fading distribution and computing a weighted average. In Appendix 9.A.1 we have derived a suitably biased fading distribution for the problem at hand. We have verified (results not reported) the accuracy of the MC-IS technique for the case of independent fading, by comparing the result from the MC-IS technique with analytical results (the latter are obtained by means of numerical integration [59]).

First, we consider the scenario where $N_{\mathrm{tr}} = N_{\mathrm{r}} = 1$, both for the above channel model and a model where the fading is independent from one packet to

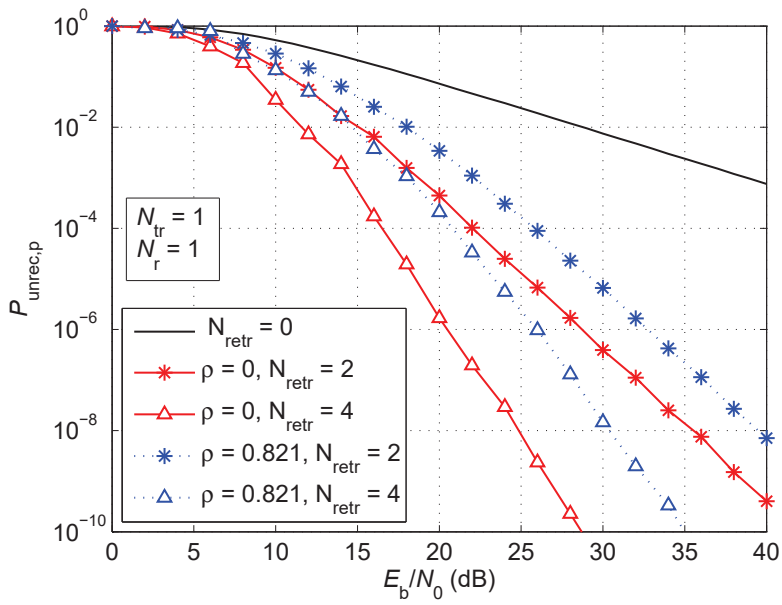

Figure 9.2: $P_{\text{unrec,p}}$ with and without temporal fading correlation.

the next (the latter case corresponds to replacing in (9.13) $R_{\text{D}}((l_2 - l_1)T_{\text{retr}})$ by a Kronecker delta $\delta(l_2 - l_1)$). Figure 9.2 shows the corresponding $P_{\text{unrec,p}}$, for $N_{\text{retr}} = 0, 2, 4$. We observe that the temporal correlation of the fading gives rise to performance degradation, as compared to the case of uncorrelated fading. This can be understood by noting that, when the first transmission of a packet is not successful because of deep fading, the probability that the retransmissions of that packet also experience deep fading is larger when the fading is correlated over time, as compared to uncorrelated fading. This observation illustrates the importance of taking temporal correlation of the fading into account.

We have computed $P_{\text{unrec,p}}$ as a function of E_{b}/N_0 for N_{retr} ranging from 0 to 4, taking the temporal correlation of the fading process into account. We have considered the scenarios $(N_{\text{tr}}, N_{\text{r}}) = (1,1)$, $(1,2)$, $(2,1)$ and $(2,2)$, for which the results are shown in Figures 9.3-9.6. The following observation can be made:

1. For the different scenarios, the high-E_{b}/N_0 behavior of $P_{\text{unrec,p}}$ confirms that the diversity order is indeed equal to $N_{\text{tr}}N_{\text{r}}(N_{\text{retr}} + 1)$, as can be verified from the slopes of the curves at high E_{b}/N_0.

2. The use of SR-ARQ gives rise to a considerable performance improvement as compared to the case $N_{\text{retr}} = 0$. The performance gain resulting from

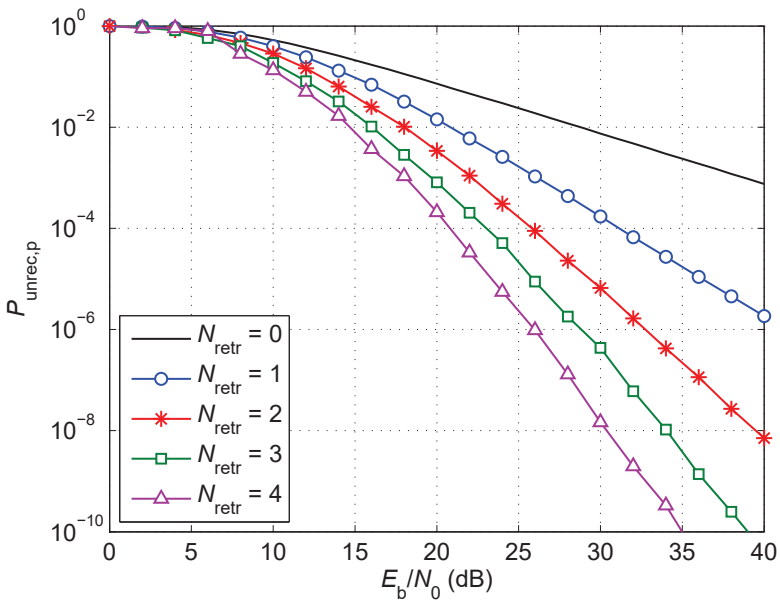

Figure 9.3: $P_{\text{unrec,p}}$ for $(N_{\text{tr}}, N_{\text{r}}) = (1, 1)$ and $\rho = 0.821$.

one additional retransmission decreases with increasing N_{retr}.

According to the method outlined in Section 9.3, we have determined the average number N_{burst} of unrecoverable packets per error burst, along with the average number of error bursts in a reference period of 4 hrs, for the various scenarios.

The performance results for the channel with time-correlated fading are summarized in Table 9.2 and Table 9.3, which for the several scenarios indicate the average number of unrecoverable packet losses per error burst and the value of E_{b}/N_0 at which the average number of error bursts in 4 hrs equals 1 (which is the performance target for HDTV [87]). Note that in order to achieve satisfactory performance, one can trade-off the hardware cost (increasing with number of transmit and receive antennas) of the wireless transceiver versus the increased latency and DSLAM retransmission buffer size (increasing with N_{retr}); note that all types of multimedia traffic benefit from an increase of the number of antennas, whereas the SR-ARQ is applied only to the video traffic.

Finally, we investigate the average transmission overhead E[ovh] of the SR-protocol with time-correlated fading. Figure 9.7 shows the average overhead as a function of E_{b}/N_0, for various combinations of $(N_{\text{tr}}, N_{\text{r}}, N_{\text{retr}})$. At high E_{b}/N_0, the average overhead is mainly equal to $P_{\text{p,e,avg}}$ (which is obtained by averaging $P_{\text{p,e}}(\mathbf{v}(l))$ over the associated fading gains), and, therefore, is only

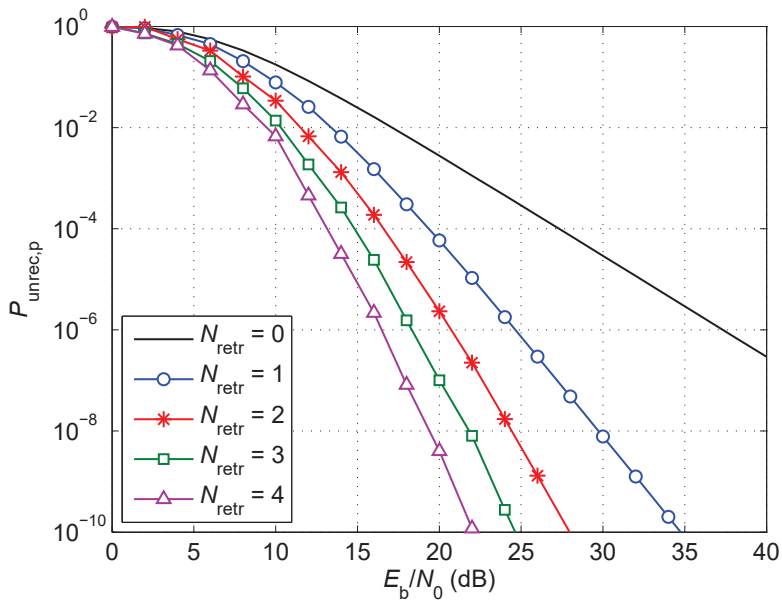

Figure 9.4: $P_{\text{unrec,p}}$ for $(N_{\text{tr}}, N_{\text{r}}) = (1, 2)$ and $\rho = 0.821$.

Table 9.2: Average number of unrecoverable packet losses per error burst.

	N_{burst} @ E[#unrec. bursts in 4 hr] = 1				
	$N_{\text{retr}} = 0$	$N_{\text{retr}} = 1$	$N_{\text{retr}} = 2$	$N_{\text{retr}} = 3$	$N_{\text{retr}} = 4$
$N_{\text{tr}} = 1,\ N_{\text{r}} = 1$	1	1.8	3.4	4.0	3.8
$N_{\text{tr}} = 1,\ N_{\text{r}} = 2$	1.3	3.2	3.1	2.9	3.1
$N_{\text{tr}} = 2,\ N_{\text{r}} = 1$	1.3	3.2	3.0	2.9	3.1
$N_{\text{tr}} = 2,\ N_{\text{r}} = 2$	2.6	2.4	2.3	2.2	2.2

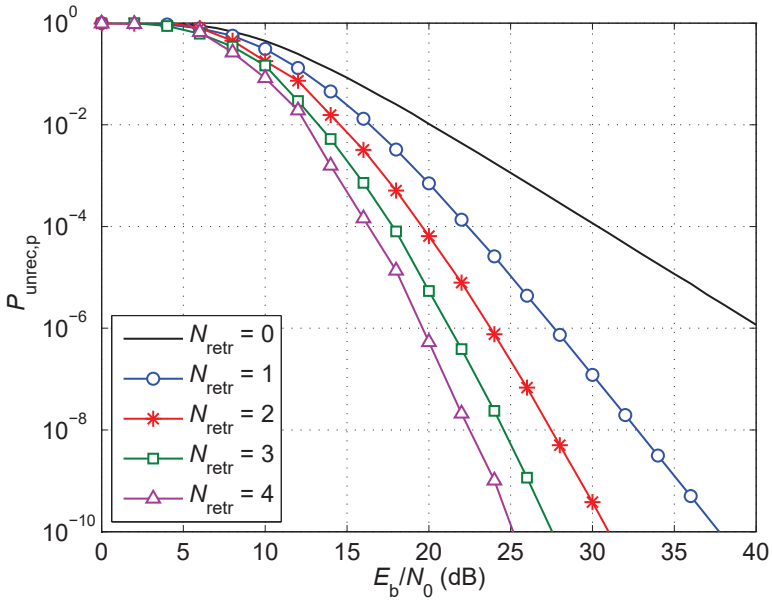

Figure 9.5: $P_{\text{unrec,p}}$ for $(N_{\text{tr}}, N_{\text{r}}) = (2, 1)$ and $\rho = 0.821$.

Table 9.3: Performance summary.

	E_{b}/N_0 @ $E[\#\text{unrec. bursts in 4 hr}] = 1$				
	no ARQ	$N_{\text{retr}} = 1$	$N_{\text{retr}} = 2$	$N_{\text{retr}} = 3$	$N_{\text{retr}} = 4$
$N_{\text{tr}} = 1,\ N_{\text{r}} = 1$	78.5 dB	45.8 dB	35.8 dB	31.4 dB	28.2 dB
$N_{\text{tr}} = 1,\ N_{\text{r}} = 2$	42.0 dB	26.8 dB	22.4 dB	20.2 dB	18.3 dB
$N_{\text{tr}} = 2,\ N_{\text{r}} = 1$	45.0 dB	29.8 dB	25.4 dB	23.2 dB	21.3 dB
$N_{\text{tr}} = 2,\ N_{\text{r}} = 2$	25.3 dB	19.3 dB	17.0 dB	15.1 dB	13.7 dB

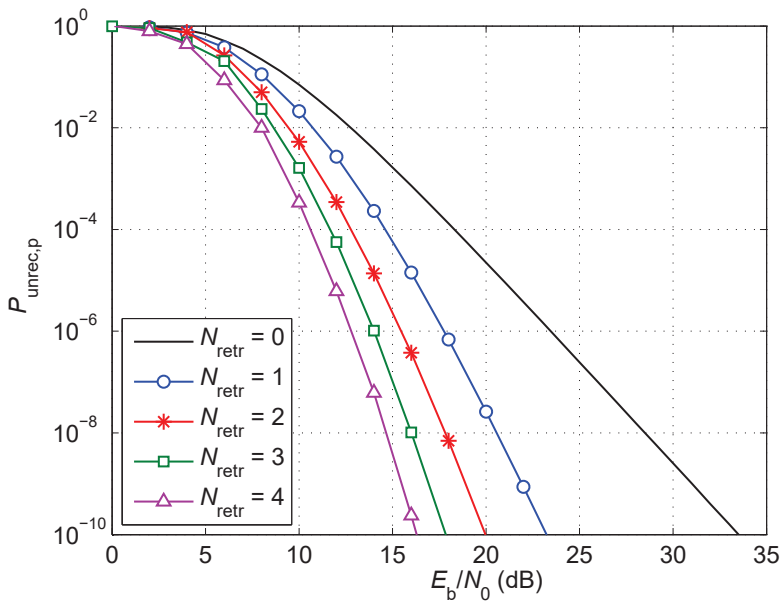

Figure 9.6: $P_{\text{unrec,p}}$ for $(N_{\text{tr}}, N_{\text{r}}) = (2, 2)$ and $\rho = 0.821$.

weakly dependent on N_{retr}; this is because when the first transmission of a packet fails, the first retransmission is much more likely to be correctly received than to be erased. For the E_{b}/N_0 values from Table 9.3, E[ovh] is always less than 10^{-3}.

9.5 Conclusions and Remarks

In this Chapter, we have analyzed an OFDM system for video transmission over a Rayleigh fading frequency-selective MIMO wireless link, with space-time coding on the PHY layer and additional protection against video packet loss by means of application layer SR-ARQ. Our theoretical findings have been illustrated in a case study involving HDTV transmission over a 2.4 GHz indoor wireless link, with severe restrictions on latency and on residual packet loss rate. In order to obtain accurate numerical results within reasonable computing time, a suitable importance sampling technique has been devised. The conclusions of our work can be summarized as follows.

1. SR-ARQ gives rise to a diversity gain, yielding improved error performance. The resulting diversity order equals $(N_{\text{retr}} + 1)N_{\text{tr}}N_{\text{r}}$.

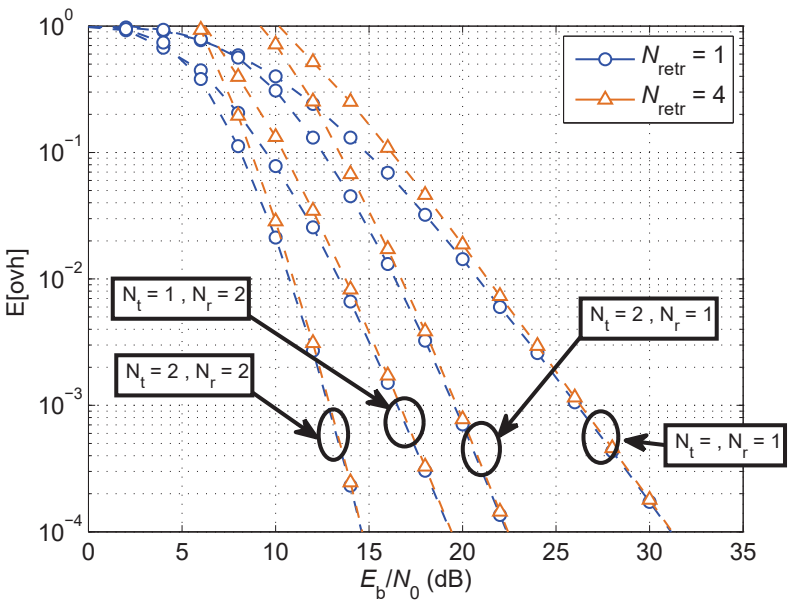

Figure 9.7: Average transmission overhead.

2. The application of SR-ARQ comes with a cost: the DSLAM must be able to handle retransmission requests that are specific to video packets, and needs a retransmission buffer with a size of $N_{\text{retr}}T_{\text{retr}}R_{\text{pack}}$ packets per TV channel; in addition, SR-ARQ increases the system latency by $N_{\text{retr}}T_{\text{retr}}$ seconds.

3. In order to achieve the required diversity order, trade-off exists between the number of antennas (which affects the hardware cost of the wireless transceiver) and the number of allowed retransmissions (which affects latency and DSLAM retransmission buffer size).

Instead of providing additional protection of the video packets by means of SR-ARQ, one could envisage to use FEC instead. Assume packet errors occur in bursts with an average size of N_{burst} lost packets per error burst. When the code is able to correct n_{corr} error bursts, the resulting diversity order equals $D = (n_{\text{corr}} + 1)N_{\text{tr}}N_{\text{r}}$. Taking into account that an (N,K) Reed-Solomon code is able to recover $N - K$ packet losses, $N - K$ should be larger than $n_{\text{corr}}N_{\text{burst}}$ in order that the code be effective. The resulting transmission overhead $(N - K)/K$ then exceeds $n_{\text{corr}}N_{\text{burst}}/K$. Considering the maximum allowed latency of 200 ms, the maximum value of K is 150. Supposing we aim at $D = 8$, the resulting ratios n_{corr}/K for $(N_{\text{tr}}, N_{\text{r}}) = (1,1), (1,2), (2,1), (2,2)$ amount to 5%, 2%, 2%

and 0.7%, respectively. Hence, the overheads introduced by FEC are much higher than for ARQ.

9.A Appendix

9.A.1 Monte Carlo Integration with Importance Sampling

Let us consider the expectation of a function $F(\mathbf{w})$ of a random vector $\mathbf{w}$, with probability density function (pdf) $p(\mathbf{w})$:

$$\mathrm{E}_p[F(\mathbf{w})] = \int F(\mathbf{w})p(\mathbf{w})\mathrm{d}\mathbf{w} \tag{9.14}$$

The subscript of the expectation operator refers to the pdf of $\mathbf{w}$. When the integral in (9.14) is difficult to evaluate analytically or numerically, we can obtain $\mathrm{E}_p[F(\mathbf{w})]$ by means of Monte-Carlo integration.

Conventional Monte-Carlo integration involves approximating $\mathrm{E}_p[F(\mathbf{w})]$ as

$$\mathrm{E}_p[F(\mathbf{w})] \approx \frac{1}{N_{\mathrm{MC}}} \sum_{k=1}^{N_{\mathrm{MC}}} F(\mathbf{w}_k) \tag{9.15}$$

where the vectors $\mathbf{w}_k$ are generated independently according to the pdf $p(\mathbf{w})$. Any degree of accuracy can be obtained by taking N sufficiently large.

The accuracy of conventional Monte-Carlo integration can be improved by means of importance sampling. Importance sampling is based on the transformation of (9.14) into

$$\mathrm{E}_p[F(\mathbf{w})] = \int F(\mathbf{w})\frac{p(\mathbf{w})}{q(\mathbf{w})}q(\mathbf{w})\mathrm{d}\mathbf{w} \tag{9.16}$$

where $q(\mathbf{w})$ is also a pdf. From (9.16) the following Monte-Carlo integration method can be derived:

$$\mathrm{E}_p[F(\mathbf{w})] = \mathrm{E}_q\left[F(\mathbf{w})\frac{p(\mathbf{w})}{q(\mathbf{w})}\right] \approx \frac{1}{N_{\mathrm{MC}}} \sum_{k=1}^{N_{\mathrm{MC}}} F(\mathbf{w}_k)\frac{p(\mathbf{w}_k)}{q(\mathbf{w}_k)} \tag{9.17}$$

where the vectors $\mathbf{w}_k$ are generated independently according to the pdf $q(\mathbf{w})$. Note that (9.17) reduces to conventional Monte-Carlo integration (9.15) when $q(\mathbf{w}) = p(\mathbf{w})$. The choice of $q(\mathbf{w})$ affects the accuracy of the approximation (9.17); the mean-square approximation error resulting from (9.17) is given by:

$$\begin{aligned}\mathrm{E}_q&\left[\left|\frac{1}{N_{\mathrm{MC}}} \sum_{k=1}^{N_{\mathrm{MC}}} F(\mathbf{w}_k)\frac{p(\mathbf{w}_k)}{q(\mathbf{w}_k)} - \mathrm{E}_p[F(\mathbf{w})]\right|^2\right] \\ &= \frac{1}{N_{\mathrm{MC}}}\left(\left(\int F^2(\mathbf{w})\frac{p^2(\mathbf{w})}{q(\mathbf{w})}\mathrm{d}\mathbf{w}\right) - (\mathrm{E}_p[F(\mathbf{w})])^2\right)\end{aligned} \tag{9.18}$$

Selecting

$$q(\mathbf{w}) = C\,F(\mathbf{w})p(\mathbf{w}) \tag{9.19}$$

with the normalization constant C determined by $C^{-1} = \int F(\mathbf{w})p(\mathbf{w})\mathrm{d}\mathbf{w}$, it is easily verified that the mean-square approximation error (9.18) is zero, irrespective of N_{MC}; this indicates that (9.17) yields the exact value of $\mathrm{E}[F(\mathbf{w})]$, even for $N_{\mathrm{MC}} = 1$. However, this choice is not practical, because we assumed from the start that C^{-1} is hard to evaluate analytically. Nevertheless, we can try to select for $q(\mathbf{w})$ a reasonable approximation to (9.19), such that the corresponding random vectors $\mathbf{w}_k$ are easily generated, and the coefficient of $1/N_{\mathrm{MC}}$ in (9.18) is much smaller than for the case $q(\mathbf{w}) = p(\mathbf{w})$. Hence, for a given accuracy, the importance sampling technique requires much smaller values of N_{MC}.

In the context of the present paper, we take $F(\mathbf{w})$ equal to the conditional probability $P_{\mathrm{unrec}}(\mathbf{v}(0), \ldots, \mathbf{v}(N_{retr}))$ from (9.2), with $\mathbf{w}$ containing the real and imaginary parts of all complex fading gains $\mathbf{h}_n^{(i,j)}(m)$ that are involved in $\mathbf{v}(0), \ldots, \mathbf{v}(N_{retr})$. Hence, $\mathrm{E}_p[F(\mathbf{w})] = P_{\mathrm{unrec,p}}$. We select

$$q(\mathbf{w}) = C\, F_{\mathrm{app}}(\mathbf{w})p(\mathbf{w}) \tag{9.20}$$

where $F_{\mathrm{app}}(\mathbf{w})$ is obtained by replacing in (9.2) $P_{\mathrm{p,e}}(\mathbf{v}(m))$ by its union upper-bound ;

$$\begin{aligned} P_{\mathrm{p,e}}(\mathbf{v}(m)) &\leq \frac{L}{N_{\mathrm{F}}} \sum_{n=1}^{N_{\mathrm{F}}} \mathrm{BER}(\mathrm{v}_n(m)) \\ &\leq \frac{L}{2N_{\mathrm{F}}} \sum_{n=1}^{N_{\mathrm{F}}} \exp\left(\frac{-E_{\mathrm{b}}\eta \mathrm{v}_n(m)}{N_0}\right) \end{aligned} \tag{9.21}$$

where the second line in (9.21) results from the inequality $Q(u) \leq (1/2)\exp(-u^2/2)$ [20]. Taking into account that $p(\mathbf{w})$ is the joint pdf of correlated zero-mean Gaussian random variables, it follows that $q(\mathbf{w})$ is a mixture of joint pdfs of correlated zero-mean Gaussian random variables. Hence, the random vectors $\mathbf{w}_k$ that are distributed according to $q(\mathbf{w})$ from (9.20) can be generated using standard techniques.

10 Concluding Remarks and Directions for Future Research

In this final chapter, we make a conclusion of this doctoral thesis and present some directions for future research. In Section 10.1, a brief overall conclusion of the obtained results is given. Section 10.2 discusses some topics that fell outside the scope of this dissertation; we propose a hybrid ARQ scheme, briefly look at the rateless Raptor code, and summarize some possible extensions to specific topics addressed in this book. Section 10.3 provides a complete list of our publications.

10.1 Summarizing Conclusions

On the DSL link, the occurrence of crosstalk and impulsive noise may have a detrimental impact on the received signal. Error correcting codes may improve the performance as compared to uncoded transmission. We considered TCM, the concatenation of a RS outer code, a byte interleaver and a TCM inner code,

and LDPC codes of several code rates and block lengths. Also a retransmission protocol, i.e. SR-ARQ, is introduced. Precoding can alleviate the damaging impact of crosstalk. We both considered linear and nonlinear precoding. LP performs worse at higher frequencies where the crosstalk channel is stronger than the direct channel. This issue is solved by NLP. On the other hand, NLP has the disadvantage of a larger BER due to the modulo operation. Also, the constellations require a larger average energy. Therefore, LP outperforms NLP at the lower frequencies where the direct channel is strong. As a performance measure, we propose a target BER and for the different protection strategies, we derived via simulations for which SNR values our target BER is achieved, for the different sizes of QAM-constellations from BPSK up to 2^{12}-QAM. Our system is subject to power constraints, i.e. an aggregate transmit power constraint and a power spectral density constraint. Moreover, we propose two bitloading algorithms, the CNS algorithm and the EZF algorithm, to efficiently load the bitloading on the different tones of the OFDM symbol. The EZF algorithm searches for the optimal solution and is therefore more computation complex, while the CNS algorithm is based on column norm scaling and has very low complexity.

These preliminary results are applied to real DSL channels, from which the channel matrices are at our disposal, corresponding to loop lengths of respectively 100 m, 150 m, 200 m and 250 m. We found that, in an environment without impulsive noise, the LDPC code of highest rate, i.e. $R_c = 20/21$ and with largest block length $K = 4320$ gives us the highest goodput and outperforms the lower rate LDPC codes and the other error correction codes. Furthermore, a minor gain can be obtained in the resulting goodput from adding SR-ARQ to our protection scheme.

On the other hand, in an environment where impulsive noise might occur, we benefit from the use of some additional redundancy added by a lower rate LDPC code. We found that for IN with $\kappa = 10$ dB, the LDPC code of rate 16/18 results in the highest goodput. For $\kappa = 20$ dB, the LDPC code of rate 5/6 is preferred. In both cases with $K = 4320$.

To determine the bitloading on the different tones of the DMT symbol, the EZF algorithm achieves highest bitrates. Although, due to its much lower complexity, we prefer the CNS algorithm. For the choice of precoder, it is absolutely clear that NLP is preferred; it allows us the use the higher tones in the bandwidth that suffer from large crosstalk.

In the second part of this work, we tackled the transmission of video over a wireless channel subject to Rayleigh fading. If a deep fade arises, a burst of packets may be erased if no suitable protection is provided. We propose space-time coding on the PHY layer and additional protection against video packet loss by means of application layer SR-ARQ and find that SR-ARQ gives rise to a diversity gain, yielding improved error performance. In order to achieve the

required diversity order, trade-off exists between the number of antennas and the number of allowed retransmissions.

10.2 Future Work

10.2.1 Extensions to Our Work that Qualify for Future Research

Here follows a short list of some other topics that may be interesting to be examined:

- ▷ For the crosstalk as introduced in Chapter 4, we assume that the channel matrix is perfectly known. A future subject of investigation could be the effect on the performance in the case that the channel matrix is the product of an estimation and that it may vary from the real channel state. An important question is how the bitloading algorithms should be adjusted so that the target performance is still reached.

- ▷ In Section 5.5.2, we discussed the performance of the concatenation of TCM with interleaver and RS code, and assumed that the interleaver is infinite large. As a consequence, the byte errors could be regarded as independent. In reality, this assumption might not correspond to the actual situation.

  If the interleaver depth equals N_{int} TCM codewords (equal to N_{int} DMT symbols as a TCM codeword covers an entire DMT symbol), a RS(n, k) codeword is unrecoverable if more than $n - k$ erasures are located in the codeword after deinterleaving. The group of N_{int} consecutive DMTs consists of DMTs that are hit and DTMs that are not hit by IN. Depending on the state s_l of the DMT (state 1 and 0 indicate respectively that a DMT is hit and not hit by IN), the distribution of the number of byte errors in the TCM codeword will be different.

  Assume that a total of N_{e} byte errors occurs in the group of N_{int} DMTs with $N_{\text{e},l}$ byte errors in the lth DMT, $l = 1, \ldots, N_{\text{int}}$, and $N_{\text{e},1} + \ldots + N_{\text{e},N_{\text{int}}} = N_{\text{e}}$. Then, the probability of an unrecoverable RS codeword

equals

$$P_{\text{unrec,RS}} = \sum_{j=n-k+1}^{n} \sum_{N_{\text{e}}=j}^{N_{\text{int}}N_{\text{byte}}} \mathrm{C}_{j}^{N_{\text{e}}} \mathrm{C}_{n-j}^{N_{\text{int}}N_{\text{byte}}-N_{\text{e}}} \cdot \sum_{N_{\text{e},1}+\ldots+N_{\text{e},N_{\text{int}}}=N_{\text{e}}} \left(\prod_{l=1}^{N_{\text{int}}} \Pr\left[\text{the } l^{\text{th}} \text{ DMT contains } N_{\text{e},l} \text{ byte errors}\right] \right) \tag{10.1}$$

where N_{byte} denotes the number of bytes in a DMT symbol and C_{j}^{i} stands for the number of combinations of j elements out of a set of i elements. With $P_{\text{b}}(N_{\text{e},l}|s_l)$ equal to the probability that a DMT with state s_l contains $N_{\text{e},l}$ byte errors, we obtain

$$P_{\text{unrec,RS}} = \sum_{j=n-k+1}^{n} \sum_{N_{\text{e}}=j}^{N_{\text{int}}N_{\text{byte}}} \mathrm{C}_{j}^{N_{\text{e}}} \mathrm{C}_{n-j}^{N_{\text{int}}N_{\text{byte}}-N_{\text{e}}} \cdot \sum_{N_{\text{e},1}=0}^{N_{\text{e}}} \sum_{N_{\text{e},2}=0}^{N_{\text{e}}-N_{\text{e},1}} \cdots \sum_{N_{\text{e},N_{\text{int}-1}}=0}^{N_{\text{e}}-N_{\text{e},1}-\ldots-N_{\text{e},N_{\text{int}-2}}} P_{\text{b}}(N_{\text{e},1}|s_1) P_{\text{b}}(N_{\text{e},2}|s_2) \ldots P_{\text{b}}(N_{\text{e},N_{\text{int}}}|s_{N_{\text{int}}}) \tag{10.2}$$

where the states s_l are generated following the Markov process as described in Section 4.5. The equation in (10.2) corresponds to a convolution.

▷ Even shorter DSL loops and larger bitrates are envisaged in the future. Can the system as proposed in this dissertation be improved to increase the bitrate efficiency and which methods should be applied?

10.2.2 Hybrid ARQ

In this book, we considered both FEC and ARQ as protection of the transmitted data. We would like to explore the combination of both strategies, i.e. a hybrid ARQ (HARQ) scheme in which the receiver sends a request for redundant packets in order to recover lost packets. We propose that the redundant packets are updated continuously at the serving network node using Galois field arithmetic. The main advantage of the proposed scheme is the small buffer size at the transmitter, which is independent of the round-trip time. In traditional SR-ARQ systems, the serving network node has to maintain a buffer for keeping copies of already transmitted packets for which retransmissions might be requested. The size of the retransmission buffer is proportional to the RTT and the maximum number of retransmissions allowed. Therefore, the buffer for

HARQ can be designed to be much smaller than for SR-ARQ. This advantage is particular important in the context of streaming on-demand video services, where the buffer size is proportional to the number of users served by the considered node (say 1000). In spite of the availability of Terabyte hard disks, on some devices and in some systems is the memory capacity still limited.

It has been shown in Chapter 2 that FEC introduces an overhead bit rate that is (up to several orders of magnitude) larger than that of ARQ. Indeed, FEC adds a fixed number of redundant packets to each block of information packets, whereas ARQ retransmits only when packets are actually lost. Here, we introduce a HARQ scheme that operates from a network node and only requests a retransmit when a packet is actually lost (thus having a similar overhead bit rate as ARQ, much smaller than the one of traditional FEC).

10.2.2.1 Description of the HARQ Scheme

At the transmission instant of the kth information packet $\mathbf{D}(k)$, the serving node computes on the fly L redundant packets $\mathbf{S}_1(k), \ldots, \mathbf{S}_L(k)$, that might be requested by the user in order to recover lost information packets. We assume that a packet contains $J \log_2(N_{\text{HARQ}})$ bits. The redundant packets are represented by row vectors containing J elements from $\text{GF}(N_{\text{HARQ}})$, and are computed recursively as

$$\mathbf{S}_l(k) = \mathbf{D}(k) + \alpha^{l-1}\mathbf{S}_l(k-1) \quad \text{for } l = 1, \ldots, L \tag{10.3}$$

where α denotes a primitive element of $\text{GF}(N_{\text{HARQ}})$, which satisfies $\alpha^{N_{\text{HARQ}}-1} = 1$. The L redundant packets contained in $\mathbf{S}(k)$ are stored at the serving network node during the inter-packet interval between the transmission instants of $\mathbf{D}(k)$ and $\mathbf{D}(k+1)$, requiring a buffer size of L packets. When requested by the user, redundant packets from $\mathbf{S}(k)$ are transmitted between the information packets $\mathbf{D}(k)$ and $\mathbf{D}(k+1)$. The information packets and the redundant packets also contain their sequence number, to allow identification by the user.

Our HARQ scheme is based on the fact that, under certain conditions, knowing at the user side the first L' ($\leq L$) rows of the state matrix at instants k and $k+m$ allows to recover up to L' lost information packets from the set $\{\mathbf{D}(k+1), \ldots, \mathbf{D}(k+m)\}$ of transmitted information packets.

A loss period starts at $k = k_0$ when the user knows the state matrix $\mathbf{S}(k_0-1)$ but packet $\mathbf{D}(k_0)$ is lost, and ends when sufficient redundant information has been received to recover the state matrix for the first time since the loss of $\mathbf{D}(k_0)$. When the number of lost information packets during the current loss period reaches $L+1$, the set of lost information packets cannot be recovered. In this case, the user waits for the serving network node's response to the request for the entire state matrix (i.e., all L rows), that has been issued when the Lth information packet loss occurred. When this response is affected by packet loss,

the user persistently requests for the entire state matrix until it is correctly received. The current loss period, which is referred to as an unrecoverable loss period, ends upon the correct reception of the entire state matrix.

A simplified analysis gets us the probability of an unrecoverable loss period with $L+1$ information packet erasures, denoted as $P_{\text{unrec,ARQ}}$,

$$P_{\text{unrec,HARQ}} = \frac{P_{\text{e,p}}(1-(1-P_{\text{e,p}})^r)^L}{\prod_{i=1}^{L}(1-(1-P_{\text{e,p}})^r P_i)} \approx r^L P_{\text{e,p}}^{L+1} \tag{10.4}$$

with $P_{\text{e,p}}$ denoting the packet loss probability in downstream directions, r equals the number of packets transmitted during the RTT, and P_i is the probability that a request for i state vectors does not give rise to a correctly received response given by

$$P_i = 1-(1-P_{\text{e,p}})^i \approx iP_{\text{e,p}} \tag{10.5}$$

where we assume that the erasure probability of a request message is negligible as compared to $P_{\text{e,p}}$. The resulting average transmission overhead amounts to $\text{E[ovh]} \approx P_{\text{e,p}}$.

The probability of an unrecoverable packet and the overhead for SR-ARQ is given by (2.16) and (2.18), respectively, where $P_{\text{e,DTU}}$ must be substituted by $P_{\text{e,p}}$.

We present some numerical results for this simplified analysis in Figures 10.1-10.3 for $r = 30$:

- Figure 10.1 shows $P_{\text{unrec,HARQ}}$ from (10.4) along with its asymptote for small $P_{\text{unrec,HARQ}}$, as a function of $P_{\text{e,p}}$, for $L = 2, 4, 6$. Also shown is $P_{\text{unrec,HARQ}} = P_{\text{e,p}}$, which corresponds to the case where no attempt is made to recover lost packets. We observe that $P_{\text{unrec,HARQ}}$ converges to its asymptote (10.4) for small $P_{\text{e,p}}$, and to $P_{\text{e,p}}$ for large $P_{\text{e,p}}$.

- Figure 10.2 displays P_{unrec} for HARQ with $L = 4$ and $r = 30$, and for SR-ARQ with $N_{\text{retr}} = 1, 2, 3$. We observe that a minimum value of $N_{\text{retr}} = 2$ (corresponding to a retransmit buffer of 60 packets) is required in order to beat the performance of HARQ in the region that is visible.

- Figure 10.3 shows the average downstream transmission overhead E[ovh] for HARQ with $L = 4$ and $r = 30$ and for SR-ARQ with $N_{\text{retr}} = 1, 2, 3$, along with the low-$P_{\text{e,p}}$ asymptote $\text{E[ovh]} = P_{\text{e,p}}$. We observe that for values of small $P_{\text{e,p}}$, the actual overhead is very small, and close to $P_{\text{e,p}}$ for all systems.

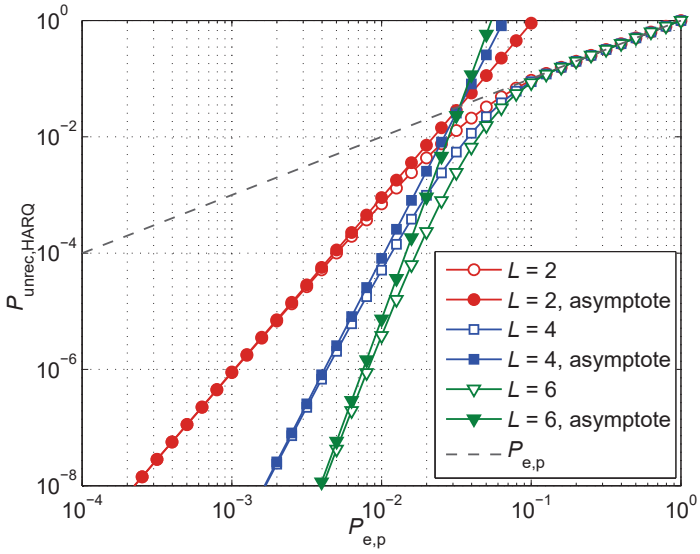

Figure 10.1: $P_{\text{unrec,HARQ}}$ as a function of $P_{\text{e,p}}$, for $r = 30$ and $L = 2, 4, 6$.

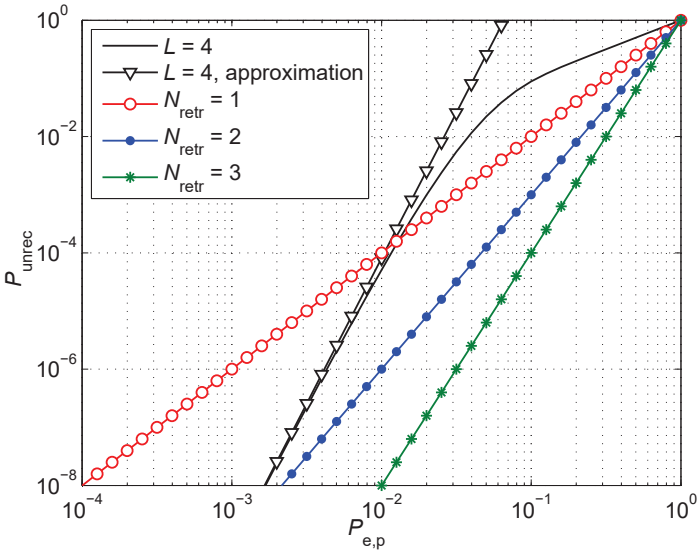

Figure 10.2: $P_{\text{unrec,HARQ}}$ for HARQ with $L = 4$ and $r = 30$, and for SR-ARQ with $N_{\text{retr}} = 1, 2, 3$.

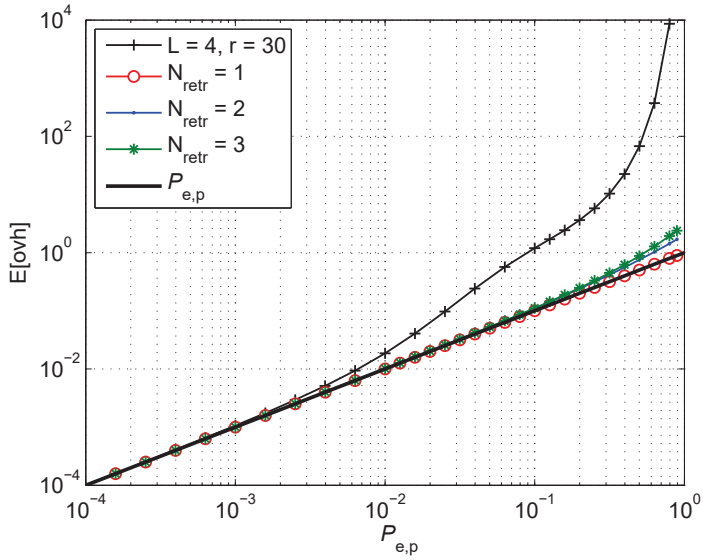

Figure 10.3: E[ovh] for HARQ with $L = 4$ and $r = 30$, and for SR-ARQ with $N_{\text{retr}} = 1, 2, 3$.

10.2.2.2 Some Preliminary Conclusions

The HARQ scheme that we propose for future research involves the computation of L redundant packets per information packet, using Galois field arithmetic, and requires a buffer of L packets at the serving node to temporarily store the continuously updated redundant information until the transmission of the next information packet. Redundant packets are requested by the user when confronted with packet losses. The user is able to recover up to L information packets per loss period, by solving a set of (up to L) linear equations. Unrecoverable packet loss occurs when more than L information packets are lost in a loss period.

For small $P_{\text{e,p}}$, the HARQ scheme and the SR-ARQ scheme yield essentially the same (very small) average transmission overhead.

For $r = 30$, we have shown that in order to outperform the HARQ scheme that operates with a buffer size of only 4 packets, the SR-ARQ scheme would require a buffer size of 60 packets. Note that these are the buffer sizes per served user. In order to fully appreciate the difference in buffer size requirements, one should take into consideration that the total required buffer size at the DSLAM that serves about 1000 users amounts to 4000 packets and 60000 packets for HARQ and SR-ARQ, respectively.

The presented HARQ scheme is more computationally demanding than SR-ARQ, mainly because of the computation of recursions (10.3) at the serving node and at the user side, respectively. A detailed examination of the complexity comparison should be performed.

Another point of attention is the singularity of the set of linear equations in (10.3), that needs to be solved to recover possible lost packets, and how to overcome this possible singularity.

Also, the HARQ scheme can be expanded by including a timer at the user side, in order to limit the latency of the packet recovery process. This is motivated by the observation that it makes no sense to recover packets that are anyway to late to be delivered to the play-out buffer of the video application.

10.2.3 Other Correcting Codes

A first attempt to tackle the investigation of other codes is made in [91] where we compare RS codes with the popular Raptor codes. Raptor codes [92] have significantly lower decoding complexity than Reed-Solomon codes. The success of Raptor codes is witnessed by their adoption in the recent multimedia communication standards such as 3GPP MBMS (Multimedia Broadcast/Multimedia Service) [93] and DVB-H [94]. We want to compare the RS codes with the Raptor codes in terms of performance and decoding complexity.

10.2.3.1 RS Erasure Coding

In the sequel, a (video) information packet refers to video payload of the IP packet, and contains L bits. Per group of k of these video information packets, an additional $n-k$ parity packets are transmitted. This results in a systematic packet codeword of n packets. The parity packets are constructed such that taking from each packet the ith block of q bits yields an RS(n,k) codeword, for all $i = 1, 2, \ldots, L/q$. This construction is illustrated in Figure 10.4. Hence, when e packets from the packet codeword are erased, each of the L/q RS codewords is affected by exactly e symbol erasures. The code parameters n and k should be selected such that the overhead and latency are limited to reasonable values.

In [91], an efficient erasure decoding algorithm is presented based on [27]. The resulting decoding complexity of a RS packet codeword is dominated by $4 \cdot 2^q\, q\, L$ additions in $\mathbb{Z}$ and $2^q\, q\, L$ multiplications in $\mathbb{Z}$, with L the number of bits in a video packet. Note that the decoding complexity of the shortened RS(n,k) code is determined by the size 2^q of the Galois field, rather than the specific code parameters n and k.

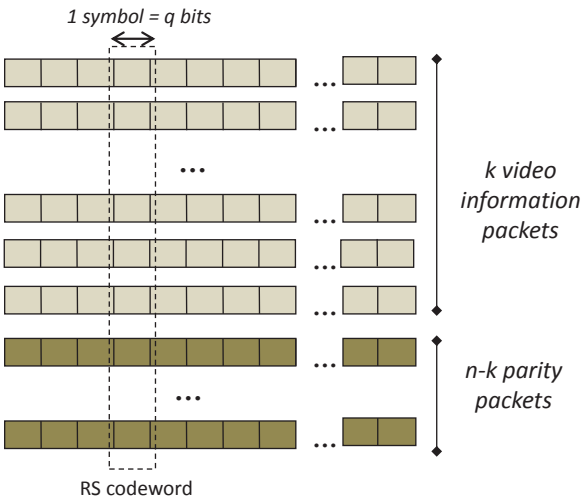

Figure 10.4: Construction of a packet codeword of the RS(n,k) code.

10.2.3.2 Raptor Codes

The Raptor (n,k) codes are non-systematic binary codes. Raptor codes are rateless, which means that for a given information word a potentially infinite number of code bits can be produced. Raptor codes are designed such that a high probability of decoding occurs when $n = k(1 + \epsilon)$ code bits are received, where k is the number of information bits and the overhead parameter ϵ is a design parameter of the Raptor code.

The construction of a Raptor packet codeword is illustrated in Figure 10.5. From each group of k video information packets, n code packets are computed. The sequence of each ith bit of all code packets forms the Raptor codeword corresponding to the information word that contains each ith bit of all video information packets, for all $i = 1, 2, \ldots, L$. The latency introduced by the Raptor code equals the duration of the packet codeword.

Raptor codes [92] are built on LT codes by first applying a pre-code to the information bits, followed by an LT code operating on the pre-code output bits. As pre-code, we use an irregular LDPC code. The LT codes were invented by Luby [95]. Each packet in the codeword is obtained from a bitwise exclusive-or (XOR) of a uniformly random selection of d information packets, where d is the degree of the code packet, and d is specified by a suitable degree distribution, called the weakened LT (wLT) distribution [92].

The LT decoding is achieved using the belief propagation algorithm on a bipartite graph that represents the information packets and the coded packets;

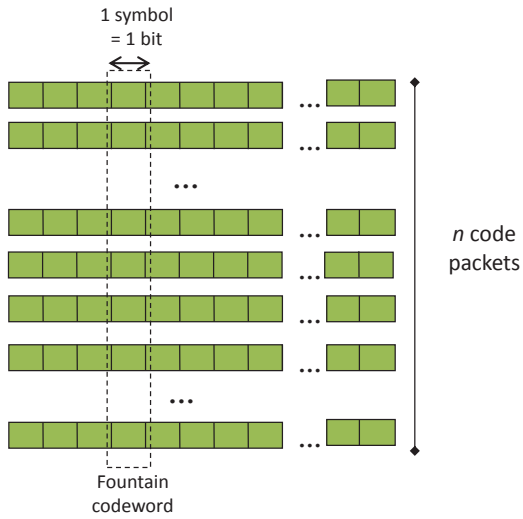

Figure 10.5: Construction of a packet codeword of the Raptor code on the application layer.

each coded packet is connected to the information packets that contribute to the considered coded packet. First, all code packets with degree one are identified. The corresponding information packets are trivially resolved, and the degree of all neighbours in the graph of the resolved information packets is reduced by one. This allows to resolve the information packets that are connected to coded packets with original degree of two. This process is repeated until all information packets have been resolved (successful decoding) or only code packets with degree higher than one remain (decoding fails). As this decoding process requires only XOR operations, the decoding complexity for LT codes is much smaller than for RS codes.

The total overhead of the Raptor codes, ϵ depends on the overheads of the two encoding blocks (pre-code and wLT code):

$$(1+\epsilon) = (1+\epsilon_{\text{LDPC}})(1+\epsilon_{\text{wLT}}) \tag{10.6}$$

In [92] it has been suggested to set $\epsilon_{\text{wLT}} = \epsilon/2$.

Decoding of the Raptor code is done in two steps. First, the LT decoder operates on the received packet codeword and returns a number of packets that have been generated by the LDPC encoder. Then the LDPC decoder attemps to recover the information packets, making use of the LT decoder output. In [91], we derive that the decoding complexity of the LT part is given by $k(1+\epsilon)(\text{E}[d]-1)$ XOR operations on packets. LDPC decoding can use the same simplified

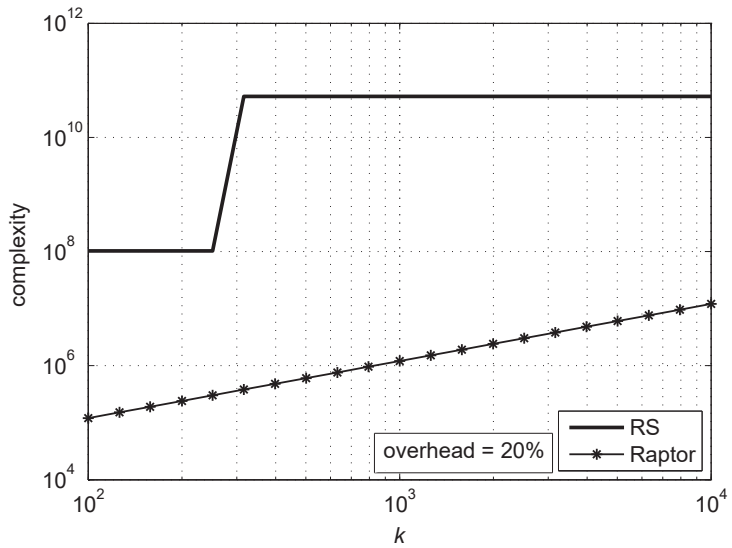

Figure 10.6: Comparison of the decoding complexity of the RS and Raptor codes.

belief propagation algorithm on a Tanner graph that represents the LDPC code. The corresponding decoding complexity equals $k(1+\epsilon_{\text{LDPC}})(\text{E}[\text{ld}]-1)$ XOR operations on packets, with E[ld] the average left degree in the Tanner graph.

10.2.3.3 Some Numerical Results

Considering an IP packet of $L = 10^4$ bits, we have displayed in Figure 10.6, the decoding complexity of a packet codeword when using the RS or Raptor code, as a function of the number k of information packets in a packet codeword. The total transmission overhead is equal to 20%. Assuming a 64-bit processor, one unit of complexity corresponds to one addition in $\mathbb{Z}$, one multiplication in $\mathbb{Z}$ or 64 XOR operations. The curves confirm the higher decoding complexity of the RS code. The curve of the RS code increases step-by-step, as the decoding complexity is a function of the size 2^q of the GF; we restrict our attention to values of q that are multiples of 8. A RS code with $q = 8$ (for which the maximum value of k is 212 when considering a 20% overhead) has the same decoding complexity as a Raptor code with $k = 89105$ information packets. It is clear that for the same decoding complexity, the Raptor codes are allowed to be much longer than the RS codes.

In Figure 10.7 we show the simulated decoder performance of the Raptor

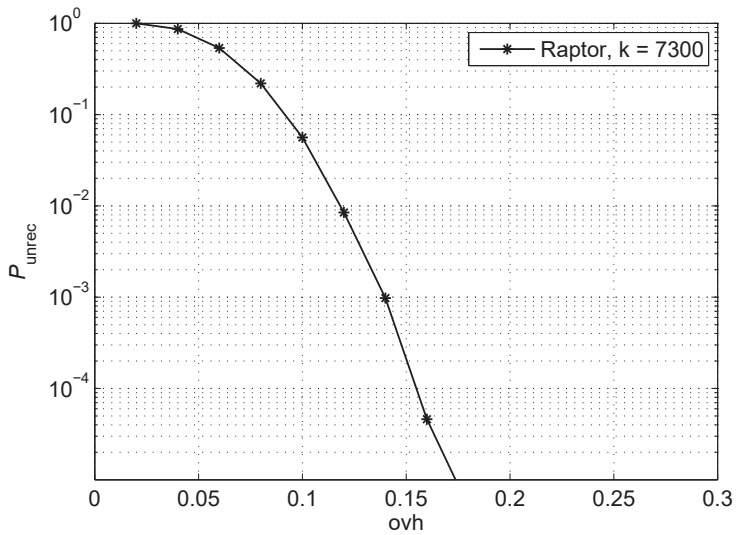

Figure 10.7: Performance of the Raptor code as a function of the overhead in the absence of erasures.

code as a function of the overhead, under the assumption that none of the packets of the packet codeword has been erased. In this case, the simplified decoding might fail because of the random nature of the code (e.g., for some of the coded packets, the LT decoder cannot reduce their degree to 1). For the Raptor code we have selected $k = 7300$ which results in a latency of 10 seconds (with $L = 10^4$ bits and a video bit rate $R_{\text{video}} = 7.3$ Mbit/s), which is acceptable for streaming video-on-demand applications. As pre-code of the Raptor code, we use a right-Poisson, left-regular LDPC code, with left degree equal to 4 [92].

In Figure 10.8, we compare the different codes in the context of streaming on-demand video. We take $R_{\text{video}} = 7.3$ Mbit/s, $L = 10^4$ and allow a transmission overhead of 20%. We limit the latency to a maximum of 10 seconds, which corresponds to Raptor codewords of maximum 7300 video information packets. We consider Raptor codes designed for respectively $\epsilon = 0.05$, $\epsilon = 0.1$, $\epsilon = 0.15$ and $\epsilon = 0.2$ but transmit a higher overhead equal to 20%, which is possible due to the rateless nature of the code. For the RS code, we select the code with $n = 254$ and $k = 212$. Our performance target is to achieve an average of no more than 1 decoding error in 4 hours. For a given decoding error probability P_{unrec}, the average number of decoding errors in 4 hours is given by $4 \cdot 3600 R_{\text{video}} P_{\text{unrec}}/(kL) \approx 10^7 P_{\text{unrec}}/k$. In the absence of error control, the

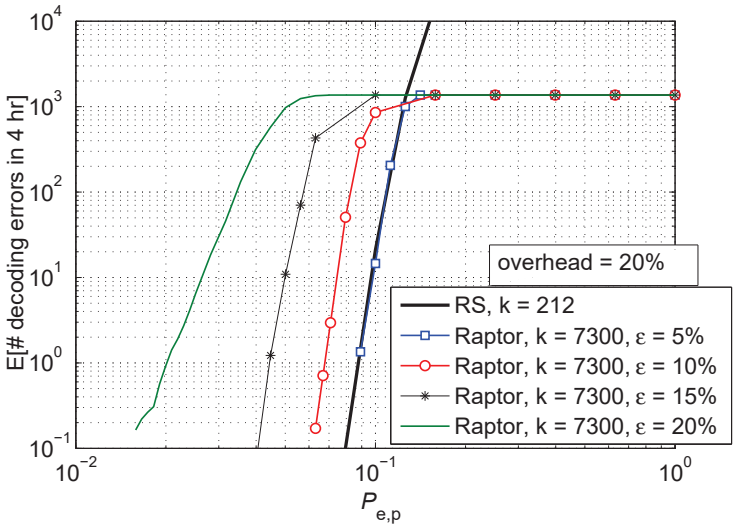

Figure 10.8: E[# decoding errors in 4 hr] for the Raptor and RS code.

erasure probability $P_{\mathrm{e,p}}$ should be limited to about 10^{-7} in order to achieve on average not more than 1 erasure in 4 hours. Figure 10.8, shows the performance of the Raptor codes and the RS code as a function of the erasure probability $P_{\mathrm{e,p}}$ at the input of the decoder. The RS code and the Raptor code designed for $\epsilon = 0.05$ have similar performance and outperform the other Raptor codes. These two codes achieve the performance target for $P_{\mathrm{e,p}} < 9 \times 10^{-2}$. The Raptor code designed for $\epsilon = 0.2$ has the lowest performance. The price to pay for the RS code is its decoding complexity, which is larger than for the Raptor codes by a factor of about 12.

10.2.3.4 Some Conclusions

It is clear that depending on the requirements of the system, e.g., a low decoding complexity, low memory requirements, low latency constraints,... other codes or ARQ protocols will satisfy our needs with similar error performance. In the comparison of examination in this section, the Raptor code designed for $\epsilon = 0.05$ performs as good as the RS code, at the gain of a 10 times smaller decoding complexity.

Further research might look into the effect of the arguments and degree distribution of the Raptor code in more detail and explore the combination of LT codes with other LDPC codes, e.g., the regular LDPC codes as used in Part I of this book.

10.3 Publications

Our work, of which was largely reported in this dissertation, has been presented in the following refereed journal and conference publications:

Journal Publications

- EURASIP Journal on Advances in Signal Processing: [59]
- IEEE Journal on Selected Areas in Communications: [96]

Conference Publications

- IEEE Symposium on Communications and Vehicular Technology in the Benelux: [97]
- NewCOM++ ACoRN Joint Workshop: [98]
- IEEE International Symposium on Information Theory and Its Applications (ISITA): [91]
- IEEE International Conference on Communications (ICC): [99, 100]

Bibliography

[1] (2015) Cisco visual networking index: Forecast and methodology, 2014-2019 white paper. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

[2] G. Ungerboeck and I. Csajka, "On improving data-link performance by increasing the channel alphabet and introducing sequence coding," in *Proc. IEEE International Symposium on Information Theory*, Jun. 1976, p. 53.

[3] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, pp. 55–67, Jan. 1982.

[4] "ITU-T Recommendation G.992.1: Single-pair high-speed digital subscriber line (SHDSL) transceivers."

[5] "ITU-T Recommendation G.993.2: Very high speed digital subscriber line transceivers 2 (VDSL2)."

[6] P. Elias, "Coding for noisy channels," *IRE Convention Record*, vol. 3, part 4, pp. 37 – 46, 1955.

[7] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260 – 269, Apr. 1967.

[8] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423, Jul. 1948.

[9] R. G. Gallager, *Low Density Parity Check Codes*. MA: MIT Press, 1963.

[10] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *IEEE Transactions on Information Theory*, vol. 32, pp. 1646 – 1646, Aug. 1996.

[11] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *Electronics Letters*, vol. 45, pp. 399 – 431, Mar. 1999.

[12] M. Fossorier, M. Mihaljevic, and H. Imai, "Reduced complexity iterative decoding of low-density parity check codes based on belief propagation," *IEEE Transactions on Communications*, vol. 47, pp. 673 – 680, May 1999.

[13] A. Emran and M. Elsabrouty, "Simplified variable-scaled min sum LDPC decoder for irregular LDPC codes," in *Proc. IEEE Consumer Communications and Networking Conference (CCNC)*, Jan. 2014, pp. 518 – 523.

[14] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Transactions on Communications*, vol. 52, pp. 670 – 678, Apr. 2004.

[15] M. Ardakani and F. R. Kschischang, "A more accurate one-dimensional analysis and design of irregular LDPC codes," *IEEE Transactions on Communications*, vol. 52, pp. 2106 – 2114, Dec. 2004.

[16] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, pp. 619 – 637, Feb. 2001.

[17] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 8, pp. 1451–1458, October 1998.

[18] E. Biglieri, R. Calderbank, A. Constantinides, A. Goldsmith, A. Paulraj, and H. V. Poor, "MIMO wireless communications," *Cambridge University Press*, 2007.

[19] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: performance criterion and code construction," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 744–765, March 1998.

[20] J. Proakis, *Digital Communications.* McGraw Hill, 2000.

[21] I. S. J. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300 – 304, Jun. 1960.

[22] G. C. Clark Jr. and J. B. Cain, *Error-Correction Coding for Digital Communications.* Springer, 1981.

[23] S. B. Wicker and V. K. Bhargava, *Reed-Solomon codes and their Applications.* IEEE Press, 1994.

[24] Y. M. Sugiyama, S. H. Kasahara, and T. Namekawa, "A method for solving the key equation for decoding of Goppa codes," *Information and Control*, vol. 27, pp. 87 – 89, Jan. 1975.

[25] R. E. Blahut, *Theory and practice of error control codes.* Reading, Mass.: Addison Wesley, 1984.

[26] G. D. J. Forney, "On decoding BCH codes," *IEEE Transactions on Information Theory*, vol. IT-11, pp. 393 – 403, Oct. 1965.

[27] F. Didier, "Efficient erasure decoding of Reed-Solomon codes," 2009. [Online]. Available: arXiv:0901.1886

[28] T. Starr, J. M. Cioffi, and P. J. Silverman, *Understanding Digital Subscriber Line Technology.* Prentice Hall, 1999.

[29] P. Golden, H. Dedieu, and K. Jacobsen, *Fundamentals of DSL technology.* Auerbach Publications, 2006.

[30] V. Oksman, H. Schenk, A. Clausen, J. Cioffi, M. Mohseni, G. Ginis, C. Nuzman, J. Maes, M. Peeters, K. Fisher, and P.-E. Eriksson, "The ITU-T's new G.vector standard proliferates 100 Mb/s DSL," *IEEE Communications Magazine*, vol. 48, no. 10, pp. 140 – 148, Oct. 2010.

[31] M. Guenach, J. Maes, M. Timmers, O. Lamparter, J.-C. Bisschoff, and M. Peeters, "Vectoring in DSL systems: practices and challenges," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM),*, Houston, Texas, Dec. 2011.

[32] R. Essiambre, G. Kramer, P. Winzer, G. Foschini, and B. Goebel, "Capacity limits of optical fiber networks," *Journal of Lightwave Technology*, vol. 28, pp. 662–701, Feb. 2010.

[33] P. Ödling, T. Magesacher, S. Host, P. Borjesson, M. Berg, and E. Areizaga, "The fourth generation broadband concept," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 62 – 69, Jan. 2009.

[34] M. Timmers, M. Guenach, C. Nuzman, and J. Maes, "G.fast: evolving the copper access network," *IEEE Communications Magazine*, vol. 51, no. 8, pp. 74 – 79, Aug. 2013.

[35] I. Almeida, A. Klautau, and C. Lu, "Capacity analysis of G.fast systems via time-domain simulations," in *Proc. IEEE International Conference on Communications (ICC)*, Budapest, Hungary, Jun. 2013.

[36] G. Ginis and J. M. Cioffi, "Vectored transmission for digital subscriber line systems," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 5, pp. 1085 – 1104, Jun. 2002.

[37] R. Cendrillon, G. Ginis, E. Van den Bogaert, and M. Moonen, "A near-optimal linear crosstalk precoder for downstream VDSL," *IEEE Transactions on Communications*, vol. 55, no. 5, pp. 860 – 863, May 2007.

[38] R. Strobel, R. Stolle, and W. Utschick, "Wideband modeling of twisted-pair cables for MIMO applications," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM),*, Dec. 2013, pp. 2828 – 2833.

[39] R. Kirkby, "Text for 'Realistic Impulsive Noise Model'," ETSI TM6 011T20, Feb. 2001.

[40] I. Mann, S. McLaughlin, W. Henkel, R. Kirby, and T. Kessler, "Impulse noise generation with appropriate amplitude, length, inter-arrival, and spectral characteristics," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 5, pp. 901–912, Jun. 2002.

[41] D. Zhang, K. Ho-Van Khuong, and T. Le-Ngoc, "Impulse noise detection techniques for retransmission to reduce delay in DSL systems," in *Proc. IEEE International Conference on Communications (ICC)*, Ottawa, Canada, Jun. 2012.

[42] "ITU-T Recommendation G.9701: Fast access to subscriber terminals (G.fast) - Physical layer specification."

[43] H. A. Suraweera, C. Chai, S. J., and A. J., ""analysis of impulse noise mitigation techniques for digital television systems," in *Proc. 8th International OFDM Workshop*, Hamburg, Sep. 2003, pp. 172 – 176.

[44] A. J., M. Feramez, and H. A. Suraweera, "Optimum noise thresholds in decision directed impulse noise mitigation for OFDM," in *Proc. International Symposium on Communication Systems Networks and Digital Signal Processing (CSNDSP 2004)*, Newcastle upon Tyne, UK, Jul. 2004, pp. 168 – 171.

[45] A. J. and H. A. Suraweera, "Impulse noise mitigation for OFDM using decision directed noise estimation," in *Proc. IEEE International Symposium on Spread Spectrum Techniques and Applications (ISSSTA 2004)*, Sydney, Australia, Aug. 2004, pp. 174 – 178.

[46] D. Fertonami and G. Colavolpe, "On reliable communications over channels impaired by bursty impulse noise," *IEEE Transactions on Communications*, vol. 57, pp. 2024 – 2030, Jul. 2009.

[47] J. Mitra and L. Lampe, "Convolutionally coded transmission over markov-gaussian channels: analysis and decoding metrics," *IEEE Transactions on Communications*, vol. 58, pp. 1939 – 1949, Jul. 2010.

[48] J. Smith, "Odd-bit quadrature amplitude-shift keying," *IEEE Transactions on Communications*, vol. 23, pp. 385 – 389, Mar. 1975.

[49] P. K. Vitthaladevuni, "Exact ber computation for cross QAM constellations," *IEEE Transactions on Wireless Communications*, vol. 4, pp. 3039 – 3050, Nov. 2005.

[50] L.-F. Wei, "Trellis-coded modulation with multidimensional constellations," *IEEE Transactions on Information Theory*, vol. 33, pp. 483–501, Jul. 1987.

[51] "ITU-T Recommendation G.9960: Unified high-speed wireline-based home networking transceivers – system architecture and physical layer specification."

[52] Y. Li and W. E. Ryan, "Mutual-information-based adaptive bit-loading algorithms for LDPC-coded OFDM," *IEEE Transactions on Wireless Communications*, vol. 6, pp. 1670 – 1680, May 2007.

[53] J. Maes, P. Spruyt, and S. Vanhastel. (2014, Jul.) G.fast breaks through the gigabit barrier. TechZine Alcatel-Lucent. [Online]. Available: http://www2.alcatel-lucent.com/techzine/g-fast-breaks-gigabit-barrier/

[54] D. Z. Filho, R. R. Lopes, R. Ferrari, and R. Suyama, "Bit loading for precoded DSL systems," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 3, 2007, pp. 353–356.

[55] J. Maes and C. Nuzman, "Energy efficient discontinuous operation in vectored G.fast," in *Proc. IEEE International Conference on Communications (ICC)*, Sydney, Australia, Jun. 2014.

[56] G. D. J. Forney and G. Ungerboeck, "Modulation and coding for linear gaussian channels," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2384 – 2415, Oct. 1998.

[57] J. Bingham, "Multicarrier modulation for data transmission: an idea whose time has come," *IEEE Communications Magazine*, vol. 28, pp. 5–14, May 1990.

[58] H. Liu and G. Li, *OFDM-Based Broadband Wireless Networks: Design and Optimization*. Wiley-Blackwell, 2005.

[59] J. Neckebroek, F. Vanhaverbeke, D. De Vleeschauwer, and M. Moeneclaey, "Protection of video packets over a wireless Rayleigh fading link: FEC versus ARQ," *EURASIP Journal on Advances in Signal Processing*, 2008, article ID 852697.

[60] F. Vanhaverbeke, F. Simoens, M. Moeneclaey, and D. De Vleeschauwer, "Binary erasure codes for packet transmission subject to correlated erasures," in *2006 Pacific-Rim Conference on Multimedia*, Hangzhou, China, Nov. 2006, pp. 48–55.

[61] N. Degrande, K. Laevens, D. De Vleeschauwer, and R. Sharpe, "Increasing the user perceived quality for IPTV services," *IEEE Communications Magazine*, vol. 46, no. 2, pp. 94–100, February 2008.

[62] G. Liang and B. Liang, "Balancing interruption frequency and buffering penalties in VBR video streaming," *Proc. IEEE International Conference on Computer Communications (INFOCOM 2007)*, pp. 1406–1414, May 2007, Anchorage, USA.

[63] T. Stockhammer, H. Jenkac, and G. Kuhn, "Streaming video over variable bit-rate wireless channels," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 268–277, April 2002.

[64] F. Borgonovo and A. Capone, "Efficiency of error-control schemes for real-time wireless applications on the Gilbert channel," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 1, pp. 246–258, January 2005.

[65] M. Tun, K. K. Loo, and J. Cosmas, "Error-resilent performance of Dirac video codec over packet-erasure channel," *IEEE Transactions on Broadcasting*, vol. 53, no. 3, pp. 649–659, September 2007.

[66] S. W. Kim, S. Y. Kim, S. Kim, and J. Heo, "Performance analysis of forward error correcting codes in IPTV," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 2, pp. 376–380, May 2008.

[67] M. Luby, T. Stockhammer, and M. Watson, "Application layer FEC in IPTV services," *IEEE Communications Magazine*, vol. 46, no. 5, pp. 94–101, May 2008.

[68] C.-Y. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video transmission over burst-error wireless channels," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 5, pp. 756–773, May 1999.

[69] J. Xu, X. Shen, J. W. Mark, and J. Cai, "Quasi-optimal channel assignment for real-time video in OFDM wireless systems," *IEEE Transactions on Wireless Communications*, vol. 7, no. 4, pp. 1417– 1427, April 2008.

[70] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, no. 5, May 1998.

[71] H. O. Burton and D. D. Sullivan, "Errors and error control," *Proc. IEEE*, vol. 60, pp. 1293–1301, November 1972.

[72] R. A. Comroe and D. J. Costello Jr., "ARQ schemes for data transmission in mobile radio systems," *IEEE Journal on Selected Areas in Communications*, pp. 472–481, July 1984.

[73] H. Yang, P. Smulders, and M. Herben, "Channel characteristics and transmission performance for various channel configurations at 60 GHz," *EURASIP journal on Wireless Communications and Networking - special issue on millimeter-wave wireless communication systems: theory and applications*, 2007, art. No 19613.

[74] M. Robin and M. Poulin, *Digital Television Fundamentals.* McGraw-Hill, 2000.

[75] P. J. Smith, M. Shafi, and H. Gao, "Quick simulation: A review of importance sampling techniques in communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 597–613, May 1997.

[76] R. Srinivasan, *Importance Sampling: Applications in Communications and Detection.* Springer, 2002.

[77] M. H. Kalos and P. A. Whitlock, *Monte Carlo Methods.* J. Wiley, 2008.

[78] "ITU-T Recommendation G.995.1: Overview of digital subscriber line (DSL) recommendations."

[79] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: an Introduction to MPEG-2.* Springer, 1996.

[80] P. D. Symes, *Digital Video Compression.* McGraw-Hill, 2003.

[81] J. Greengrass, J. Evans, and A. C. Begen, "Not all packets are equal, part 1: Streaming video coding and SLA requirements," *IEEE Internet Computing*, vol. 13, pp. 70–75, January-February 2009.

[82] ——, "Not all packets are equal, part 2: The impact of network packet loss on video quality," *IEEE Internet Computing*, vol. 13, pp. 74 – 82, March-April 2009.

[83] S. Kanumuri, P. C. Cosman, A. R. Reibman, and V. A. Vaishampayan, "Modeling packet-loss visibility in MPEG-2 video," *IEEE Transactions on Multimedia*, vol. 8, no. 2, pp. 341–355, 2006.

[84] J. Zhang, J. F. Arnold, and M. R. Frater, "A cell-loss concealment technique for MPEG-2 coded video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 4, pp. 659–665, June 2000.

[85] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, July 2003.

[86] A. R. Reibman and D. Poole, "Characterizing packet-loss impairments in compressed video," *Proc. IEEE International Conference on Image Processing ICIP2007*, pp. 77–80, September 2007, San Antonio, USA.

[87] "Technical report TR-126, triple-play services quality of experience (QoE) requirements," DSL Forum, Tech. Rep., December 2006, http://www.broadband-forum.org/technical/download/TR-126.pdf.

[88] A. Gurtov and S. Floyd, "Modelling wireless links for transport protocols," *ACM Computer Communications Review*, vol. 34, no. 2, pp. 85–96, April 2004.

[89] C. Hoene and A. Wolisz, "Measuring the impact of slow user motion on packet loss and delay over IEEE 802.11b wireless links," *Proceedings of the 28th Annual IEEE International Conference on Local Computer Networks (LCN '03)*, pp. 652–662, October 2003, Bonn, Germany.

[90] T. A. Wysocki and H.-J. Zepernick, "Characterization of the indoor radio propagation channel at 2.4 GHz," *Journal of Telecommunications and Information Technology*, vol. 1, no. 3-4, pp. 84–90, 2000.

[91] J. Neckebroek, M. Moeneclaey, and E. Magli, "Comparison of Reed-Solomon and Raptor codes for the protection of video on-demand on the erasure channel," in *Proc. IEEE International Symposium on Information Theory and Its Applications (ISITA)*, Taichung, Taiwan, Oct. 2010.

[92] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[93] *3GPP TS 26346 v.6.2.0*, Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs, September 2009. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/26346.htm

[94] *ETSI EN 302 304 V1.1.1*, Digital Video Broadcasting (DVB), November 2004, transmission System for Handheld Terminals (DVB-H).

[95] M. Luby, "LT codes," in *Proc. of IEEE Symposium on Foundations of Computer Science*, Vancouver, Canada, November 2002, pp. 271–280.

[96] J. Neckebroek, H. Bruneel, and M. Moeneclaey, "Application layer ARQ for protecting video packets over an indoor MIMO-OFDM link with correlated block fading," *IEEE Journal on Selected Areas in Communications*, pp. 467–475, 2010.

[97] J. Neckebroek, F. Vanhaverbeke, and M. Moeneclaey, "The impact of Rayleigh fading on packet loss in FEC-protected real-time packet-based transmission systems," in *Proc. IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*, Delft, The Netherlands, Nov. 2007.

[98] ——, "Comparison of FEC and ARQ for protection of video data over a wireless Rayleigh fading link," in *Proc. NewCOM++ - ACoRN Joint Workshop*, Barcelona, Spain, Mar. 2009.

[99] J. Neckebroek, M. Moeneclaey, M. Guenach, M. Timmers, and J. Maes, "Comparison of error-control schemes for high-rate communication over short DSL loops affected by impulsive noise," in *Proc. IEEE International Conference on Communications (ICC)*, Budapest, Hungary, Jun. 2013.

[100] J. Neckebroek, M. Moeneclaey, W. Coomans, M. Guenach, P. Tsiaflakis, R. Moraes, and J. Maes, "Novel bitloading algorithms for coded G.fast DSL transmission with linear and nonlinear precoding," in *Proc. IEEE International Conference on Communications (ICC)*, London, UK, Jun. 2015.