UNIVERSITÀ DEGLI STUDI DI PADOVA
DEPARTMENT OF INFORMATION ENGINEERING
MASTER THESIS IN TELECOMMUNICATIONS ENGINEERING

FEEDBACK SCHEMES TO IMPROVE THE FINITE-LENGTH PERFORMANCE OF POLAR CODES

SUPERVISOR: Prof. Michele Zorzi, UNIVERSITY OF PADOVA
CO-SUPERVISOR: Prof. Rüdiger Urbanke, EPFL
CO-SUPERVISOR: Dr. Marco Mondelli, EPFL

MASTER CANDIDATE: *Stefano Olivotto*

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

ACADEMIC YEAR 2015-2016

Abstract

Polar codes, introduced by Arikan in 2009, are the first class of codes to provably achieve capacity of binary symmetric memoryless channels with low complexity. However, despite the remarkable properties of polar codes, their finite-length performance with successive cancellation decoding has been found to be not as good as other families of codes, such as LDPC and turbo codes, greatly limiting their practical impact. Much effort has been devoted to the improvement of their finite-length performance in terms of packet error rate on a single transmission.
This work, however, adopts a different perspective. We use feedback schemes to reduce the packet error rate, allowing in exchange for a moderate delay on the decoding. Three schemes based on successive cancellation are proposed and compared. For the most promising ones, mathematical models for the delay are developed, and their accuracy is verified. We first focus in the BEC, but extensions to the BAWGNC are provided. We then derive some simple bounds on the delay.

Sommario

I codici polari, introdotti da Arikan ne 2009, sono la prima classe di codici che in modo dimostrabile raggiungono la capacità di canali binari simmetrici senza memoria. Ad ogni modo, nonstante le notevoli proprietà dei codici polari, le loro prestazioni in regime di lunghezza di blocco finita con decodifica a cancellazioni sequenziali si sono verificate essere non all'altezza di altre famiglie di codici, quali gli LDPC e i turbo codici. Molti sforzi sono stati dedicati al miglioramento delle prestazioni in regime di lughezza di blocco finita per quanto concerne il tasso di errori di pacchetto.
Il presente lavoro, tuttavia, adotta una prospettiva differente. Schemi a feedback vengono usati per ridurre il tasso di errore di pacchetto, accettando per contro un ritardo moderato nella decodifica. Tre schemi basati sulla cancellazione sequenziale sono proposti e comparati. Per i più promettenti, dei modelli matematici per il ritardo sono sviluppati, e la loro accuratezza è verificata. Ci focalizzaremo dapprima sul canale binario a cancellazione, ma estensioni al canale binario a rumore gaussiano bianco additivo saranno fornite. Deriveremo poi qualche semplice limite sul ritardo.

Acknowledgments

I am deeply thankful to Michele Zorzi. He has been not only my supervisor at the University of Padova but also my mentor during my stay at the EPFL, and has constantly followed me with great patience and dedication. His advice has always been precious, and the talks I had with him greatly enriching.
My heart-felt thanks go to Rüdiger Urbanke, who warmly welcomed me at the LTHC and IPG. He has always been ready to help me whenever I needed, and the discussions I had with him have always been brilliant and full of food for thought. I never left his office unsatisfied, and he devoted me plenty of hours.
I owe a debt of gratitude I will probably never be able to pay back to Marco Modelli. He has been the person every Master student doing his Thesis should wish to meet. He is not only a walking encyclopedia on information theory, but his is also smart, brilliant and funny. This thesis would not certainly exist without him. His door has always been open for me to discuss whatever topic I wanted to discuss. He has a great enthusiasm for what he does, but, more importantly, he also possesses the exceptional capacity of infecting people with his great passion and enthusiasm. When he talks about his work, one immediately understands the commitment, the effort and the love he puts in it. His advice, not only on the Thesis or on my work, but also on my academic life and perspectives, has been brilliant and very valuable to me. My knowledge of academic life and of the work of a PhD student are mostly due to him, and I will never thank him enough for this.
I'm also really grateful to all the people of the IPG: researchers, professors, PhD students, post-docs and admin staff for the extraordinary way the welcomed me. They really made me feel part of the group, and made my stay in Lausanne great and un-

forgettable. Among them a special thanks to Damir Laurenzi is required. I will never understand how a sysadmin can be so kind and ready to help. It is thanks to him and his precious work and advice that the experimental part of this Thesis exists. The list of problems of mine he solved is enormous.
Finally I would like to thank my family who supported me during all my experience in Lausanne. They spurred me to embark on this adventure, and have constantly been there for me to help me with anything and give me the invaluable love only a great family can provide.

Contents

List of acronyms

ARQ	Automatic Repeat reQuest
BAWGNC	Binary Additive White Gaussian Noise Channel
B-DMC	Binary-Discrete Memoryless Channel
BEC	Binary Erasure Channel
BER	Bit Error Rate
BSC	Binary Symmetric Channel
CRC	Cyclic Redundancy Check
EPFL	École Polytechnique Fédérale de Lausanne
FED	Forward Error Correction
FTP	File Transfer Protocol
i.i.d.	independent identically distributed
IP	Internet Protocol
IPG	Information Processing Group
LDPC	Low Density Parity Check
LR	Likelihood Ratio
LLR	Log-Likelihood Ratio
LTHC	Communication Theory Laboratory
MC	Markov Chain
PER	Packet Error Rate
r.v.	random variable
SC	Successive cancellation
TCP	Transmission Control Protocol

List of symbols

P_e	Packet error probability
N	number of bits in a packet
n	number of polarization levels
K	number of information bits in a packet
R	code rate
R_{eff}	effective rate
C	channel capacity
W	channel symbol
$I(W)$	symmetric capacity of channel W
$Z(W)$	Bhattacharyya parameter of channel W
U	random variable associated to information symbol (or word)
U_0^N	row random vector $[U_0, ..., U_N]$
u	realization of information symbol
u_0^N	row vector $[u_0, ..., u_N]$
$\mathscr{U}$	set of all information symbols
$\mathcal{U}$	alphabet of information symbols
X	random variable associated to encoded symbol (or codeword)
X_0^N	row random vector $[X_0, ..., X_N]$
x	realization of encoded symbol
x_0^N	row vector $[x_0, ..., x_N]$
$\mathscr{X}$	set of all encoded symbols
$\mathcal{X}$	alphabet of encoded symbols

Y	random variable associated to received symbol (or received word)
Y_0^N	row random vector $[Y_0, ..., Y_N]$
y	realization of received symbol
y_0^N	row vector $[y_0, ..., y_N]$
$\mathscr{Y}$	set of all received symbols
$\mathcal{Y}$	alphabet of received symbols
$H(X)$	entropy of random variable X
$I(X;Y)$	mutual information between random variables X and Y
ε	erasure probability of the binary erasure channel
D	average delay

List of Figures

List of Algorithms

Chapter 1

Introduction

Data transmission has become a vital foundation of our society. The amount of data transmitted in the world per second is barely estimable, and it is experiencing an exponential growth.

Therefore the capability to transmit fast and reliably has become more and more significant. Channel coding addresses the topic of reliable data transmission. Its purpose is to maximize both the rate at which information is transmitted and its reliability. From its origins, with the seminal paper *A Mathematical Theory of Communication* by Claude E. Shannon [1], to present days, channel coding has provided a fundamental contribution into allowing fast data transmission.

The discovery of polar codes by Arikan in 2008 [2] represented a major breakthrough in coding theory. They are the first class of codes that provably achieve capacity for memoryless symmetric channels with low encoding and decoding complexity. Furthermore, their explicit construction and recursive structure make them especially suitable for fast and efficient hardware implementations [3], [4] and [5].

However, despite the great interest they have aroused, their practical impact and applications still remain quite negligible.

The main reason is that, in spite of the promising asymptotic properties of polar codes, at finite-length regimes they still perform poorly in comparison to LDPC codes and turbo code, which also benefit of an additional decade of research and development.

The aim of this Thesis is to provide a way to overcome these weak points. However, unlike many proposed solutions whose aim is to mainly improve the packet error rate for short and moderate block lengths on a single transmission, the novelty of this work consists in the usage of feedback schemes. We therefore accept the presence of a delay in information reception, but in exchange we greatly reduce the packet error probability.

In confirmation of our choice to use the feedback, its employment has been recently considered as a way to obtain capacity-achieving polar codes that are able adapt the rate to a channel that is not fully known [6] [7].

The Thesis is organized as follows:

- **Chapter 2** provides a quick introduction to channel coding and an essential overview of polar codes.
- **Chapter 3** deals with the three proposed feedback schemes. They are presented in an increasing order of complexity and performance.
- **Chapter 4** presents the performance of the schemes applied to the binary erasure channel. Performance measurements are obtained by simulation of the schemes.
- **Chapter 5** provides mathematical models for the proposed schemes. These models are built using the theory of Markov chains. It will also present an analysis of the accuracy of the models, by comparing them to the simulated results.
- **Chapter 6** is an application of the most sophisticated scheme to list decoding of polar codes. This will allow to evaluate the performance improvement that can be provided by the use of list decoding.
- **Chapter 7** is devoted to the application of the last scheme to the binary additive white Gaussian noise channel, and its mathematical model.
- **Chapter 8** is devoted to the derivation of some simple bounds and asymptotic behaviors of metrics of interest.
- **Chapter 9** draws some conclusions on the work presented in the Thesis, and suggests some possible future work along its lines.

1.1 Notation

We use upper case letters U, X and Y to denote random variables associated respectively to the information, encoded and received symbols. We use lower case letters u, x and y to denote their realizations.

We use the notation U_i^j to denote a row vector of length N whose components are $U_i^j = [U_i, U_{i+1}, ..., U_j]$ if $i \leq j$, or the null vector if $j > i$, and similarly for X_i^j, Y_i^j, u_i^j, x_i^j and y_i^j.

We also use notations $U_{0,e}^{N-1}$ and $U_{0,o}^{N-1}$, for N even, to denote the two subvectors of length $\frac{N}{2}$ given by the elements of vector U_0^{N-1} with even and odd indices respectively.

$$\begin{aligned} U_{0,e}^{N-1} &= [U_0, U_2, ..., U_{N-2}] \\ U_{0,o}^{N-1} &= [U_1, U_3, ..., U_{N-1}] \end{aligned} \tag{1.1}$$

1.2 Preliminary Definitions

Matrix F on $\mathbb{GF}(2)$ is defined as

$$F \triangleq \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{1.2}$$

The Kronecker product between a matrix $A = (a_{ij})$, m-by-n, and a matrix $B = (b_{ij})$, r-by-s, is defined as

$$A \otimes B \triangleq \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix} \tag{1.3}$$

which is a mr-by-ns matrix.

We denote by $F^{\otimes n}$ the 2^n-by-2^n matrix defined recursively as $F^{\otimes n} = F \otimes F^{\otimes(n-1)}$ and $F^{\otimes 1} = F$.

Definition 1.1 *The* bit-reversal *operation over n bits is defined as the operation that associates to a positive integer $0 \leq i \leq 2^n - 1$ the positive integer $0 \leq j \leq 2^n - 1$ obtained by reversing the binary representation of i over n bits. More formally, bit-reversal is obtained as follows:*

Algorithm 1 Bit-reversal

1: $i = [b_0, b_1, ..., b_{n-2}, b_{n-1}]_2$, $b_k \in \{0, 1\}$ $\forall k = 0, ..., n-1$

2: $[b_0, b_1, ..., b_{n-2}, b_{n-1}]_2 \rightarrow [b_{n-1}, b_{n-2}, ..., b_1, b_0]_2$

3: $j = [b_{n-1}, b_{n-2}, ..., b_1, b_0]_2$

Example 1.1 *We want to bit reverse* $i = 3$ *over* $n = 4$ *bits. We apply the algorithm:*

Algorithm 2 Bit-reversal of $i = 3$ over $n = 4$ bits

1: $i = [0, 0, 1, 1]_2$

2: $[0, 0, 1, 1]_2 \rightarrow [1, 1, 0, 0]_2$

3: $12 = [1, 1, 0, 0]_2$

Therefore the bit-reversed of 3 *over* 4 *bits is* 12*.*

We will denote by B_n the permutation matrix that performs on row vectors the bit-reversal permutation of indices $[0, ..., n-1]$. Clearly, $B_2 = I_2$ (identity matrix), and $B_n = {B_n}^{-1}$.

Chapter 2

Channel coding and Polar Codes

2.1 Introduction to Channel Coding

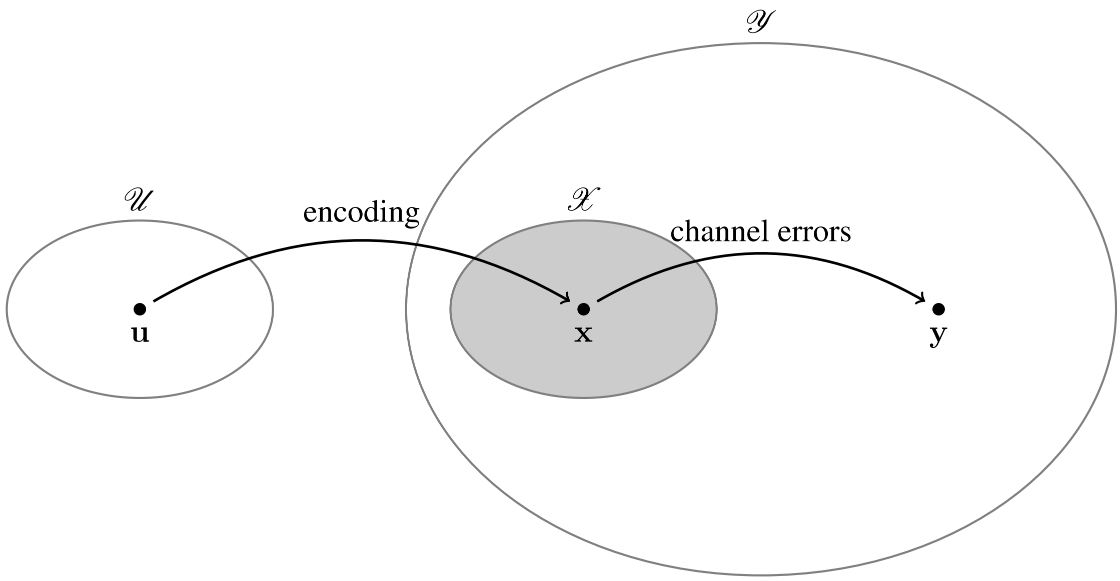


Figure 2.1: symbol encoding and transmission.

Our purpose is to transmit an information symbol (or word) $\mathbf{u} \in \mathscr{U}$ through a channel. *Channel coding* [8] is a technique that consists in replacing the information word $\mathbf{u} \in \mathscr{U}$ with a codeword $\mathbf{x} \in \mathscr{X}$, which will be the word that will be transmitted through the channel. The channel introduces some errors, and therefore the receiver receives a word $\mathbf{y} \in \mathscr{Y}$, as depicted in Fig. 2.1. Decoding then consists in taking the received word $\mathbf{y}$ and associating it with a codeword $\hat{\mathbf{x}} \in \mathcal{X} \subset \mathcal{Y}$[1] which is close (according to some metric) to $\mathbf{y}$. Then, from codeword $\hat{\mathbf{x}}$ the information message $\hat{\mathbf{u}}$ is retrieved. The association received word-codeword can be thought as a partition of all possible received words $\mathscr{Y}$ into regions, each one associated with only one codeword, that is

$$\mathscr{Y} = \bigcup_{\mathbf{x}_i \in \mathscr{X}} \mathcal{R}_{\mathbf{x}_i} \tag{2.1}$$

and

$$\mathcal{R}_{\mathbf{x}_i} \cap \mathcal{R}_{\mathbf{x}_j} = \emptyset \quad \forall i \neq j \tag{2.2}$$

Since we assume the encoding-decoding map $\mathbf{u} \leftrightarrow \mathbf{x}$ to be bijective, we must have $|\mathscr{U}| = |\mathscr{X}|$.

Words $\mathbf{u}$, $\mathbf{x}$ and $\mathbf{y}$ can be represented as vectors whose entries are defined on alphabets $\mathbb{U}$, $\mathbb{X}$ and $\mathbb{Y}$ respectively, therefore we can write $\mathscr{U} = \mathcal{U}^K$, $\mathscr{X} \subset \mathcal{X}^N$ and $\mathscr{Y} = \mathcal{Y}^N$, where K and N are the lengths of the vectors

In this Thesis only binary transmission will be considered, therefore from now on we will always assume that alphabets $\mathbb{U}$ and $\mathbb{X}$ are binary, that is, $\mathbb{U} = \mathbb{X} = \{0, 1\}$, which implies $\mathcal{U} = \{0, 1\}^K$ and $\mathcal{X} \subset \{0, 1\}^N$, and also $|\mathscr{U}| = |\mathscr{X}| = 2^K$

If a channel takes binary input, then it is said to be a *binary channel*.

Channel coding works because the encoding operation introduces some redundancy. This is obtained by taking $N > K$, and therefore the space of all possible words has higher dimension than the space of information words. This means that decision regions will have dimension N, whereas information words will have dimension K. Therefore a projection is performed. As a consequence, many vectors $\mathbf{y}$ of dimension N will be associated to a single information word. This is the operation that allows error robustness, and accounts for the *forward error correction* (FEC) capability of channel coding, that is the possibility of correcting channel errors directly from the received word, without need for retransmission.

We stress the fact that the key requirement for FEC capability is not $|\mathscr{Y}| > |\mathscr{U}|$, but the fact than word $\mathbf{y}$ (and therefore $\mathbf{x}$) is longer than word $\mathbf{u}$.

[1] $\mathcal{X} \subset \mathscr{Y}$ or, more generally, $\mathscr{X}$ is isomorphic to some subset $\mathscr{X}'$ of $\mathscr{Y}$

We define the rate of the code as

$$R \triangleq \frac{K}{N} \tag{2.3}$$

The communication scenario we considered so far is then depicted in Fig. 2.2, where for the sake of clarity and simplicity we merged into a single block *decoder* the operations of decision $\mathbf{y} \to \hat{\mathbf{x}}$ and inverse map $\hat{\mathbf{x}} \to \hat{\mathbf{u}}$.

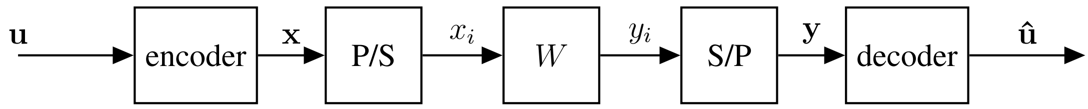


Figure 2.2: communication scenario.

The block error probability is defined as

$$P_e \triangleq \mathbb{P}[\hat{\mathbf{u}} \neq \mathbf{u}] \tag{2.4}$$

and it is one of the most important metrics that are used to evaluate the performance of a code.
In information theory words $\mathbf{u}$ and $\mathbf{x}$ are modeled as random vectors characterized by some probability mass distributions $p_{\mathbf{Y}}(\cdot)$ and $p_{\mathbf{X}}(\cdot)$. Clearly, since there is a bijection between $\mathbf{u}$ and $\mathbf{x}$, their laws can be derived one from the other. Channel $W : \mathcal{X} \to \mathcal{Y}$ is characterized by transitions probabilities $p_{\mathbf{Y}|\mathbf{X}}(\cdot|\cdot)$ or $f_{\mathbf{Y}|\mathbf{X}}(\cdot|\cdot)$, depending on whether the channel has discrete or continuous output. The notation $W(\cdot|\cdot)$ will also be used to denote the transition probabilities. Channels are always assumed to be memoryless, which implies $p_{\mathbf{Y}|\mathbf{X}}(\mathbf{b}|\mathbf{a}) = \prod_{i=1}^{n} p_{Y|X}(b_i|a_i)$. In this case, we also write W^N to denote the channel $W^N : \mathcal{X}^N \to \mathcal{Y}^N$ obtained by N uses of W: $W(y_1^N|x_1^N) = \prod_{i=1}^{n} W(y_i|x_i)$.

Definition 2.1 *The entropy $H(X)$ of random variable X is defined as*

$$H(X) \triangleq \sum_{a \in \mathcal{X}} p_X(a) \log_2 \frac{1}{p_X(a)} \tag{2.5}$$

if X is discrete, or as

$$H(X) \triangleq \int_{a \in \mathcal{X}} f_X(a) \log_2 \frac{1}{f_X(a)} da \tag{2.6}$$

if it is continuous.

Definition 2.2 *The conditional entropy $H(X|Y)$ of random variable X given random variable Y is defined as*

$$\begin{aligned} H(X|Y) &\triangleq \sum_{b\in\mathcal{Y}} p_Y(b) H(X|Y=b) \\ &= \sum_{b\in\mathcal{Y}} p_Y(b) \sum_{a\in\mathcal{X}} p_{X|Y}(a|b) \log_2 \frac{1}{p_{X|Y}(a|b)} \end{aligned} \tag{2.7}$$

where probability mass distributions change to probability density functions and sums to integrals according to whether alphabets are discrete or continuous.

Definition 2.3 *Then, mutual information between random variables and Y is defined as*

$$\begin{aligned} I(X;Y) &\triangleq H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= I(Y;X) \end{aligned} \tag{2.8}$$

Definition 2.4 *If $x_1^n = [x_1, ..., x_n]$ and $y_1^n = [y_1, ..., y_n]$ are the codeword and the received word respectively at the input and output of memoryless channel W characterized by transition probabilities $p_{\mathbf{Y}|\mathbf{X}}(y_1^n|x_1^n) = \prod_{i=1}^n p_{Y|X}(y_i|x_i)$, then the capacity of channel W is defined as*

$$C \triangleq \max_{p_X(x)} I(X;Y) \quad \textit{[bit/channel use]} \tag{2.9}$$

Capacity gives a measure on the maximum information we can transmit reliably trough a channel in the sense specified by Shannon's channel coding theorem:

Theorem 2.1 *Consider a transmission of information rate R, as defined in eq.2.3, bits per channel use over a channel W of capacity C, as defined in eq. 2.9, bits per channel use. Then,*

1. *if $R < C$ and packet length N is sufficiently large, it is possible to build an encoder-decoder procedure that makes P_e as small as desired, i.e. that ensures reliable communications;*

2. *on the converse, if* $R > C$ *no encoder-decoder procedure can ensure reliable communications.*

Definition 2.5 *For a binary discrete memoryless channel (B-DMC), that is a channel that is memoryless and which has* $\mathbb{X} = \mathcal{X} = \{0, 1\}$*, it is also customary to define the symmetric capacity*

$$I(W) \triangleq \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} \frac{1}{2} W(y|x) \log_2 \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)} \tag{2.10}$$

which is the mutual information of W by taking independent inputs with uniform distribution.

Definition 2.6 *A B-DMC* W *is said to be* symmetric *if there exists a permutation* π *operating on* $\mathcal{Y}$ *such that:*

1. $\pi = \pi^{-1}$
2. $W(\pi(y)|1) = W(y|0) \;\; \forall y \in \mathcal{Y}$

Proposition 2.1 *For a symmetric B-DMC the symmetric capacity is equal to its capacity, that is,* $I(W) = C$

In this work only symmetric channels will be considered, therefore we will mainly use symmetric capacity $I(W)$.
For binary memoryless channels capacities are $0 \leq C, I(W) \leq 1$.

2.2 Overview of Polar Codes

Polar codes [2] [9] are a novel coding technique invented by Arikan in 2009. It is the first encoding-decoding scheme that provably achieves capacity with low complexity, namely, $\Theta(N \log N)$, with block length N).

2.2.1 Code Construction

The encoding procedure is recursive, and starts by taking two copies of the same B-DMC channel $W : \mathcal{X} \to \mathcal{Y}$. The two copies are polarized by combining them in the way depicted in Fig. 2.3. U_i denotes the uncoded bits, X_i the coded bits and Y_i the

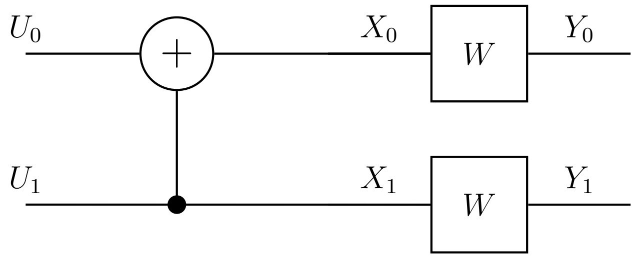


Figure 2.3: base case.

symbols received by the receiver.
This construction corresponds to matrix operation

$$X_0^1 = U_0^1 B_2 F = U_0^1 F \tag{2.11}$$

We remark that 2.11 is a linear transformation.
The result of this operation is a new channel $W_2 : \mathcal{X}^2 \to \mathcal{Y}^2$, defined by

$$W_2(y_0, y_1 | u_0, u_1) = W(y_0 | u_0 \oplus u_1) W(y_1 | u_1) \tag{2.12}$$

Since the transformation of 2.11 is invertible, we have

$$I(U_0, U_1; Y_0, Y_1) = I(X_0, X_1; Y_0, Y_1) \tag{2.13}$$

Furthermore, if U_0 and U_1 are taken independent and uniformly distributed in $\{0, 1\}$, X_0 and X_1 will also be uniformly distributed and independent. Therefore, by taking uniform and independent inputs at both sides, we have

$$I(W_2) = I(U_0, U_1; Y_0, Y_1) = I(X_0, X_1; Y_0, Y_1) = 2I(W) \tag{2.14}$$

where the last equality follows from the independence of X_0 and X_1.
From the chain rule and the independence of U_0 and U_1, we obtain

$$I(U_0, U_1; Y_0, Y_1) = I(U_0; Y_0, Y_1) + I(U_1; Y_0, Y_1, U_0) \tag{2.15}$$

From this decomposition two channels naturally arise, defined as follows:

- W^-: $\mathcal{X} \to \mathcal{Y}^2$, $U_0 \longmapsto (Y_0, Y_1)$, U_1 is unknown and treated as noise
- W^+: $\mathcal{X} \to \mathcal{Y}^2 \times \mathcal{X}$, $U_1 \longmapsto (Y_0, Y_1, U_0)$, U_0 is supposed to be known

with

$$\begin{aligned} I(W^-) &= I(U_0; Y_0, Y_1) \\ I(W^+) &= I(U_1; Y_0, Y_1, U_0) \end{aligned} \tag{2.16}$$

Channels W^+ and W^- are called *synthetic* or *virtual* channels.
Consistently with the indexing of Fig. 2.3, we adopt the following index convention:

1. indices are from 0 to $2^n - 1$, with $N = 2^n$
2. index transformation $i \to 2i$ is associated polarization in the $-$ direction
3. index transformation $i \to 2i + 1$ is associated to polarization in the $+$ direction
4. successive cancellation starts from index 0 (first bit) and ends at index $N - 1$ (last bit)

The channel are characterized by transition probabilities

$$\begin{aligned} W^-(y_0, y_1|u_0) &= W_2^{(0)}(y_0, y_1|u_0) \\ &= \sum_{u_1 \in \mathcal{X}} \frac{1}{2} W(y_0|u_0 \oplus u_1) W(y_1|u_1) = \frac{1}{2} \sum_{u_1 \in \mathcal{X}} W_2(y_0^1|u_0^1) \\ W^+(y_0, y_1, u_0|u_1) &= W_2^{(1)}(y_0, y_1, u_0|u_1) \\ &= \frac{1}{2} W(y_0|u_0 \oplus u_1) W(y_1|u_1) = \frac{1}{2} W_2(y_0^1|u_0^1) \end{aligned} \tag{2.17}$$

Here it can already be intuitively understood where the polarization comes from: it is clear that channel W^- will be worse than W since, in order to know U_0, one needs to know both X_0 *and* X_1 (or equivalently U_1), whereas in W the knowledge of X_0 only allowed to know U_0. U_1 is related to Y_1 via the uncertainty introduced by channel W, and therefore in W^- an additional source of uncertainty is introduced with respect to W. On the other hand, W^+ is intuitively better than W. In fact, since by construction U_0 is already known, U_1 can be reconstructed if we know X_0 *or* X_1, whereas in W the knowledge of X_1 was mandatory to know U_1. Therefore in W^+ an additional source of information is introduced with respect to W .
With the index notation previously introduced, it results that bad channels are mostly located in the first bits.
A more formal justification of polarization is given by eq. 2.18.

$$I(U_1; Y_0, Y_1, U_0) = H(U_1) - H(U_1|Y_0, Y_1, U_0) \geq H(U_1) - H(U_1|Y_1) = I(W) \quad (2.18)$$

Therefore, using eq. 2.18 and eq. 2.15 the following theorem is obtained:

Theorem 2.2 *Let W be a D-BMC and W^+ and W^- as in eq. 2.16. Then*

$$I(W^-) \leq I(W) \leq I(W^+) \quad (2.19)$$

and

$$\frac{I(W^-) + I(W^+)}{2} = I(W) \quad (2.20)$$

with $I(W^-) = I(W) = I(W^+)$ if and only if $I(W) = 0$ or $I(W) = 1$

At first glance, the hypothesis of knowing U_0 in synthetic channel W^+ may seem a bit artificial and arbitrary, especially, beyond the mere definitions, from a more practical point of view. In fact, since the purpose of this scheme is to decode a block of bits, for sure U_0 and U_1 are not known, and, as we said, channel W^- is the worst. Hence, there will be some error probability on the decoding of U_0. Therefore, how can we assume U_0 to be known for W^+? The reason is that, as we will see later, at the limit for $N \to \infty$ U_0 will be correctly decoded with probability 1 or $\frac{1}{2}$ (which is the same as guessing). If it

is $\frac{1}{2}$, then we simply do not put any information in it (that is, the sender and the receiver agree beforehand on what the value of U_0 will be). If it is 1 then for sure we will know it without errors. Therefore we see that in both cases it is correct to assume that once we are about to decode U_1, U_0 is correctly known.
The previous scheme is the founding block, and works for a packet of 2 bits. The following step, for packets of 4 bits, is shown in Fig. 2.4:

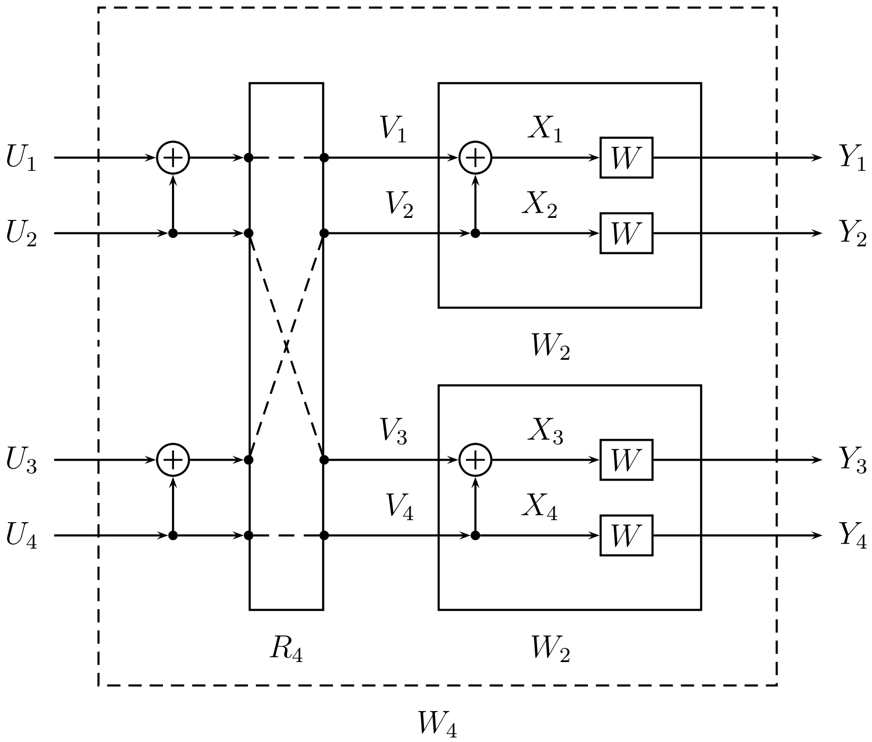


Figure 2.4: code construction for n=2.

The extension to packets of N bits is done recursively, using the scheme in Fig. 2.5, which exemplifies the recursion $(W_{N/2}, W_{N/2}) \rightarrow W_N$. In Fig. 2.5 matrix R_N performs the reshuffle operation by splitting vector S_0^{N-1} into two vectors $V_0^{N/2-1} = S_{0,e}^{N-1}$ and $V_{N/2}^{N-1} = S_{0,o}^{N-1}$.
The encoding relation can be written as $x_0^{N-1} = u_0^{N-1} G_N$, where matrix G_N can be decomposed as the product $G_N = B_N F^{\otimes n}$ for $N = 2^n$.

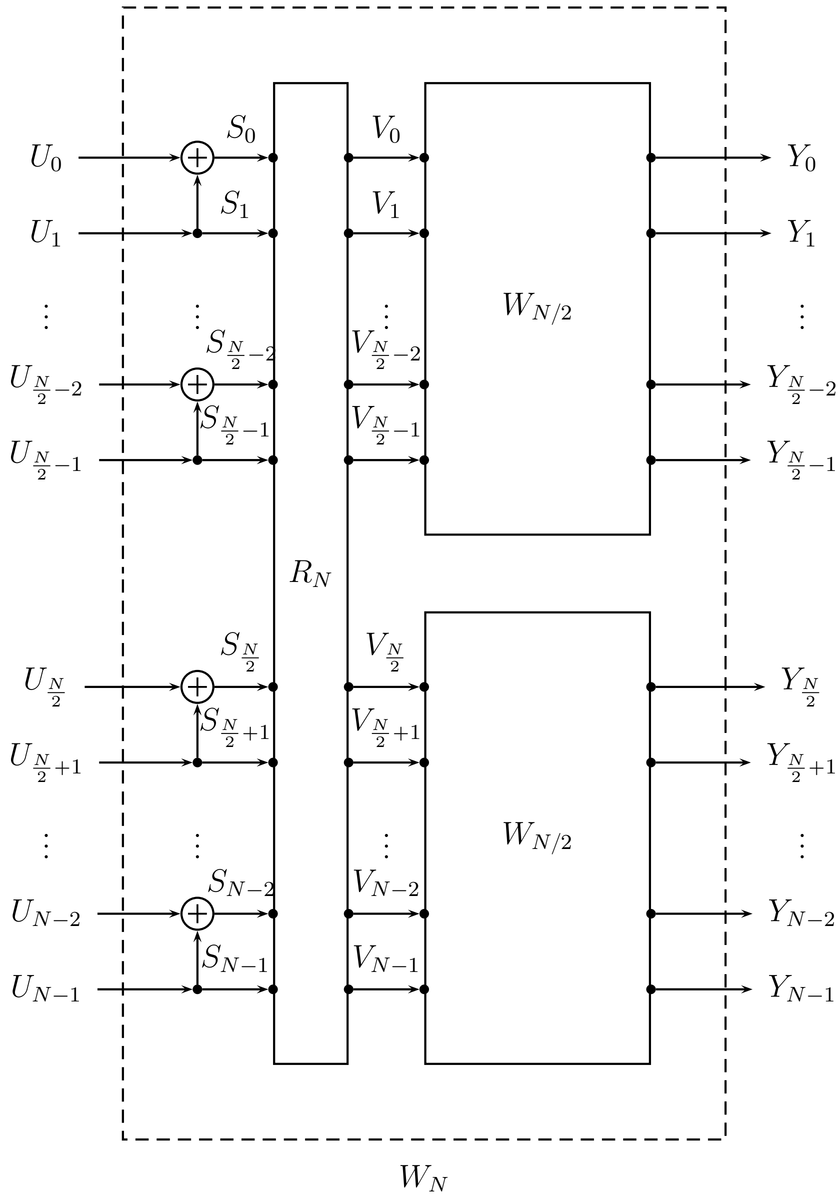


Figure 2.5: recursive code construction.

Hence, we have

$$x_0^{N-1} = u_0^{N-1} B_N F^{\otimes n} \tag{2.21}$$

The relationship between channels W_N and W^N is a generalization of eq. 2.12:

$$W_N(y_0^{N-1}|u_0^{N-1}) = W^N(y_0^{N-1}|x_0^{N-1}) = W^N(y_0^{N-1}|u_0^{N-1}G_N) \tag{2.22}$$

An important remark is necessary: from equation 2.21 one can see that the encoding relation can be thought as performed on a vector $\tilde{u}_0^{N-1} = u_0^{N-1}B_N$, which is a bit-reversal of some other vector u_0^{N-1}. Therefore, it is clear that since it is the application of a permutation, bit-reversal is irrelevant from the point of view of performance evaluation, and hence it can be neglected. This is also true for simulations where, since the code is linear, in order to save computations the all-0s codeword is considered, and the encoding operations are skipped.
If we just apply matrix $F^{\otimes n}$, what we obtain, for $n = 3$, is shown in Fig. 2.6.

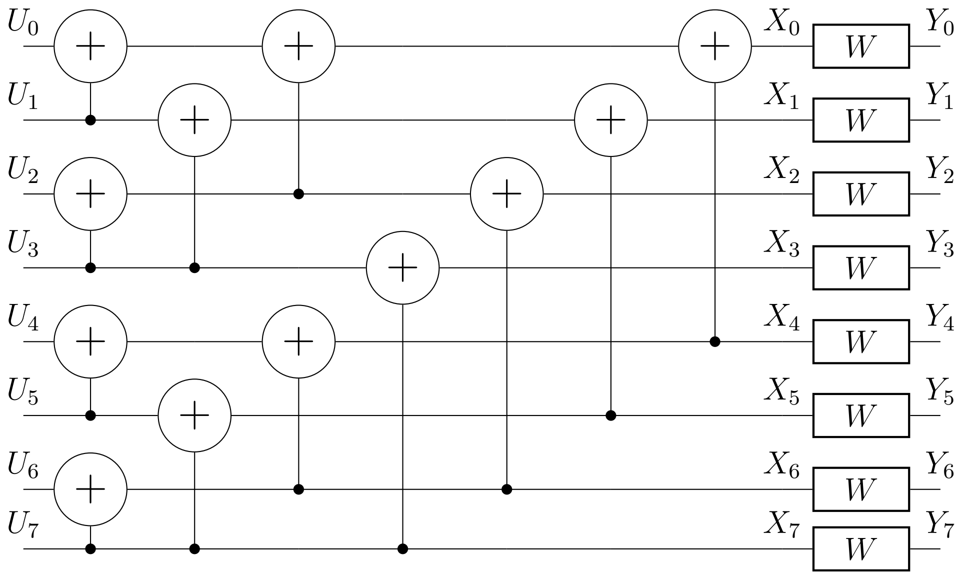


Figure 2.6: N=8.

Polarized synthetic channels $W_N^{(i)} : \mathcal{X} \to \mathcal{Y}^N \times \mathcal{X}^{i-1}$ for $0 \le i \le N-1$ are obtained from W_N as

$$W_N^{(i)}(y_0^{N-1}, u_0^{i-1}|u_i) \triangleq \sum_{u_{i+1}^{N-1} \in \mathcal{X}^{N-i}} \frac{1}{2^{N-1}} W_N(y_0^{N-1}|u_0^{N-1}) \tag{2.23}$$

One can also compute $W_N^{(2i)}$ and $W_N^{(2i+1)}$ by generalizing eqs. 2.17 to $W_{N/2}^{(i)}$, which correspond to the $-$ and $+$ operations. This makes clearer the recursive nature of channel polarization. The result is eqs. 2.24.

$$\begin{aligned}
&W_N^{(2i)}(y_0^{N-1}, u_0^{2i-1}|u_{2i}) \\
=&\frac{1}{2}\sum_{u_{2i+1}} W_{N/2}^{(i)}(y_0^{N/2}, u_{0,o}^{2i-1}|u_{2i+1}) W_{N/2}^{(i)}(y_{N/2}^{N-1}, u_{0,e}^{2i-1} \oplus u_{0,o}^{2i-1}|u_{2i} \oplus u_{2i+1}) \\
&W_N^{(2i+1)}(y_0^{N-1}, u_0^{2i}|u_{2i+1}) \\
=&\frac{1}{2} W_{N/2}^{(i)}(y_0^{N/2-1}, u_{0,o}^{2i-1}|u_{2i+1}) W_{N/2}^{(i)}(y_{N/2}^{N-1}, u_{0,e}^{2i-1} \oplus u_{0,o}^{2i-1}|u_{2i} \oplus u_{2i+1})
\end{aligned} \tag{2.24}$$

for $i = 0, ..., N/2 - 1$.
The properties of symmetric capacity apply to the generic recursive step, namely

$$\begin{aligned}
I(W_N^{(2i)}) \leq I(W_{N/2}^{(i)}) \leq I(W_N^{(2i+1)}) \\
I(W_N^{(2i)}) + I(W_N^{(2i+1)}) = 2I(W_{N/2}^{(i)})
\end{aligned} \tag{2.25}$$

Therefore also conservation of total information applies:

$$\sum_{i=0}^{N-1} I(W_N^{(i)}) = NI(W) \tag{2.26}$$

The fundamental property of polar codes is the following.

Theorem 2.3 *For any B-DMC W, channels $\{W_N^{(i)}\}_{i=0,\ldots,N-1}$ polarize in the sense that, $\forall \delta \in]0, 1[$ fixed,*

$$\begin{aligned}
\lim_{N\to\infty} \frac{|\mathcal{G}_\delta|}{N} = I(W) \\
\lim_{N\to\infty} \frac{|\mathcal{B}_\delta|}{N} = 1 - I(W)
\end{aligned} \tag{2.27}$$

where

$$\begin{aligned}
&\mathcal{G}_\delta \triangleq \{W_N^{(i)} | I(W_N^{(i)}) > 1 - \delta\} \text{ is the set of “good” channels} \\
&\mathcal{B}_\delta \triangleq \{W_N^{(i)} | I(W_N^{(i)}) < \delta\} \text{ is the set of “bad” channels}
\end{aligned} \tag{2.28}$$

Clearly, $I(W) = 0$ denotes a useless (pure noise) channel, and and $I(W) = 1$ means that the channel is perfect (no errors are introduced by the channel).

2.2.2 Encoding

A crucial quantity for our analysis will be the so-called *Bhattacharyya parameter* of channel W, which will be denoted by $Z(W)$.

Definition 2.7 *The* Bhattacharyya parameter *of channel* W *is defined as*

$$Z(W) \triangleq \sum_{y \in \mathcal{Y}} \sqrt{W(y|0)W(y|1)} \tag{2.29}$$

It is $0 \leq Z(W) \leq 1$. The reason of its importance is that $Z(W_N^{(i)})$ gives an upper bound on the bit error probability with SC (i.e., given that all previous bits are known) at bit i and on the ML decision error probability for uncoded bits.
In the particular case of a BEC it gives *exactly* the probability of error with SC.
Therefore, the Bhattacharyya parameter is used as a measure of the reliability of the channel. In particular, $Z(W) = 1 \Leftrightarrow I(W) = 0$, that is the channel is useless (pure noise), and $Z(W) = 0 \Leftrightarrow I(W) = 1$, i.e., the channel is perfect (no errors are introduced by the channel).
Furthermore, the Bhattacharyya parameter behaves, under polarization transformations (i.e., operations $+$ and $-$), in a similar way as the symmetric capacity. In particular,

$$Z(W^+) = Z(W)^2 \tag{2.30a}$$

$$Z(W^-) \leq 2Z(W) - Z(W)^2 \tag{2.30b}$$

$$Z(W^+) \leq Z(W) \leq Z(W^-) \tag{2.30c}$$

and in general

$$Z(W_N^{(2i+1)}) = Z(W_{N/2}^{(i)})^2 \tag{2.31a}$$

$$Z(W_N^{(2i)}) \leq 2Z(W_{N/2}^{(i)}) - Z(W_{N/2}^{(i)})^2 \tag{2.31b}$$

$$\sum_{i=0}^{N-1} Z(W_N^{(i)}) \leq NZ(W) \tag{2.31c}$$

In the sense of eq.2.31c we can say that polarization improves reliability.
Then, the encoding procedure is as follows: given a block size $N = 2^n$ for some n

and a rate R, the RN (clearly this quantity must be made integer, e.g., by rounding it to nearest integer, or by taking the floor or the ceiling) channels with lowest Bhattacharyya parameter are selected. These are the channels that will carry information, whereas the others will be *frozen*, i.e., no information bit is sent through them (for the sake of clarity, it can be assumed that frozen bits will be set to 0).
Theorem 2.3 implies that for $N \to \infty$, $I(W)N$ channels will be perfect and $(1 - I(W))N$ will be useless, therefore the code achieves capacity.

2.2.3 BEC

For the binary erasure channel some interesting simplifications occur.
Firs of all, inequality in eq. 2.30b becomes an equality, which allows to recursively and exactly compute all Bhattacharyya parameters of the virtual channels. Moreover, if W is a BEC(ε),

$$Z(W) = \varepsilon \tag{2.32}$$

and therefore

$$I(W) = 1 - \varepsilon = 1 - Z(W) \tag{2.33}$$

W^+ and W^- will also be BECs with parameters

$$\begin{aligned} Z(W^+) &= \varepsilon^+ = Z(W)^2 = \varepsilon^2 \\ Z(W^-) &= \varepsilon^- = 2Z(W) - Z(W)^2 = 2\varepsilon - \varepsilon^2 \end{aligned} \tag{2.34}$$

We also remark that equality in eq. 2.30b implies

$$Z(W^+) + Z(W^-) = 2Z(W) \tag{2.35}$$

This extends to any recursion step:

$$Z(W_N^{(2i)}) = 2Z(W_{N/2}^{(i)}) - Z(W_{N/2}^{(i)})^2 \tag{2.36a}$$

$$Z(W_N^{(2i+1)}) = Z(W_{N/2}^{(i)})^2 \tag{2.36b}$$

which implies

$$\sum_{i=0}^{N-1} Z(W_N^{(i)}) = NZ(W) = N\varepsilon \tag{2.37}$$

The fact that for the BEC $Z(W)$ and $I(W)$ can be determined exactly one from the other, and that these quantities can be exactly determined for all virtual channels, is the reason why the BEC is often taken as a model to study polar codes.

In Fig.2.7 the fraction of channels that have symmetric capacity $I(W_N^{(i)}) \in [\delta, 1-\delta]$ (middle channels) or $I(W_N^{(i)}) > 1-\delta$ (good channels) is shown. The bad channels are clearly the remaining ones. The polarization is clearly visible, and, in addition, we get that the speed of polarization is exponentially fast, which is proven in [2].

In Fig. 2.8 the values of $I(W_N^{(i)})$ for $i = 0, ..., 2^n - 1$ at various n are depicted. The polarization process is clearly visible.

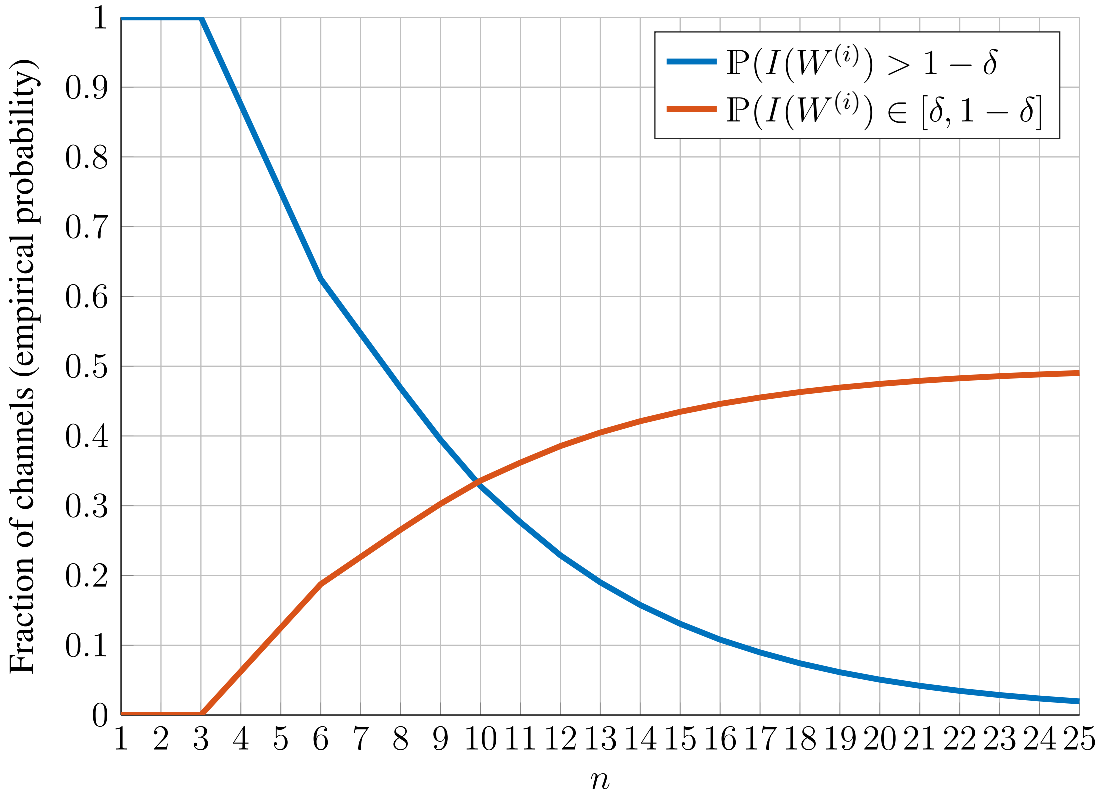
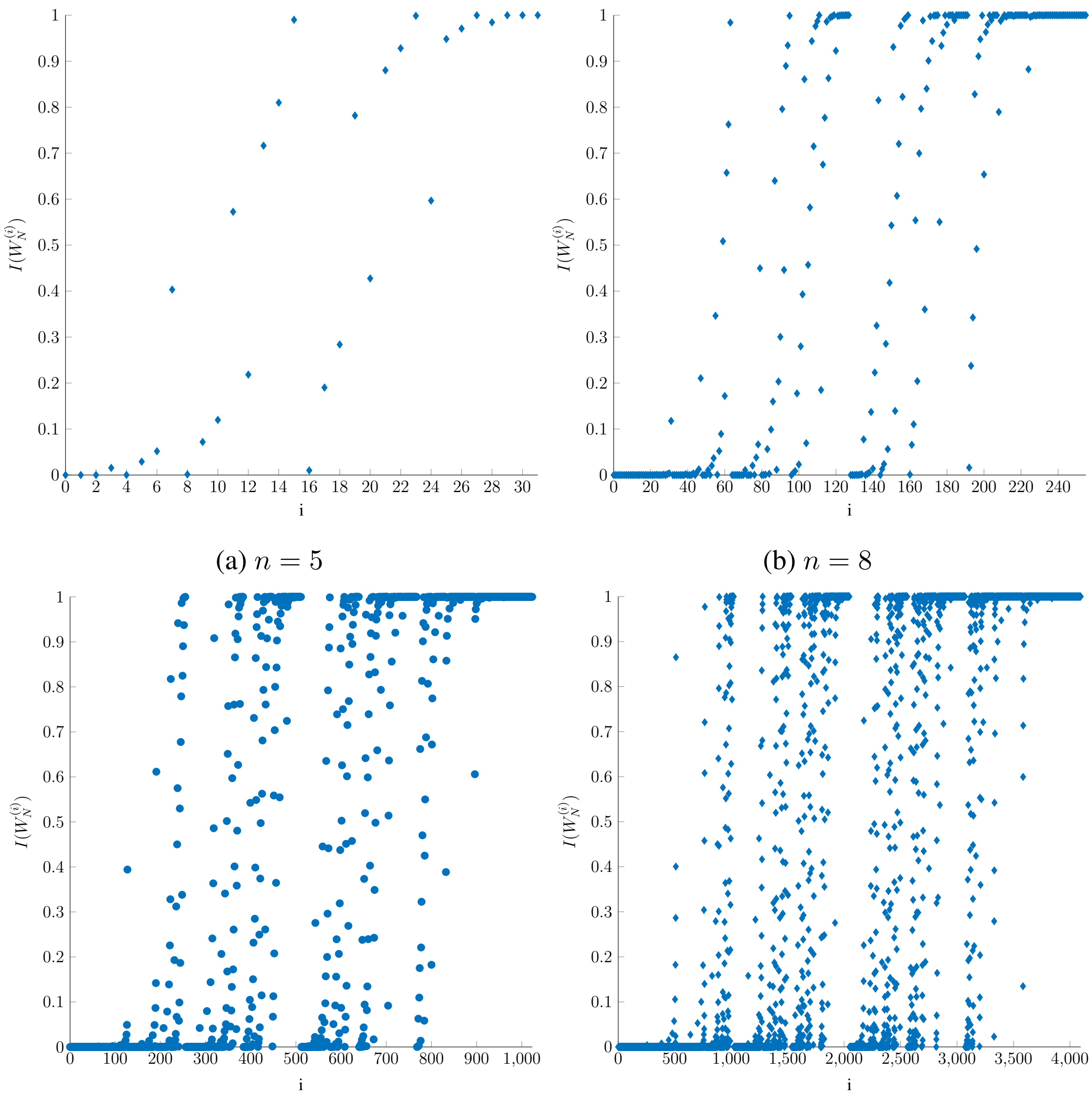


Figure 2.7: fraction of good and middle channels for $\delta = 10^{-3}$.



(a) $n = 5$ (b) $n = 8$

(c) $n = 10$ (d) $n = 12$

Figure 2.8: distribution of $I(W_N^{(i)})$ for various n.

An important remark can be made: it is not the absolute number of middle channels that decreases. In fact, it increases sublinearly in N since the fraction must tend to 0, but still increases to ∞.

2.2.4 Decoding

The decoding scheme that allows to achieve capacity with low complexity, namely $\Theta(N \log N)$, is successive cancellation (SC). It consists in successively decoding all bits of the packets starting from the first to the last one, and using for the decoding of a bit also the results obtained by decoding all the previous ones. The decoding procedure is based on the LRs (likelihood ratio) of the channels associated to the received bits.

Definition 2.8 *The LR of the i-th synthetic channel $W_n^{(i)}$ is defined as*

$$\tilde{L}_N^{(i)}(y_0^{N-1}, \hat{u}_0^{i-1}) \triangleq \frac{W_n^{(i)}(y_0^{N-1}, \hat{u}_0^{i-1}|u_i = 0)}{W_n^{(i)}(y_0^{N-1}, \hat{u}_0^{i-1}|u_i = 1)} \tag{2.38}$$

Definition 2.9 *The LLR (log-likelihood ratio) of the i-th synthetic channel $W_n^{(i)}$ is defined as*

$$L_n^{(i)} \triangleq \ln\left(\tilde{L}_N^{(i)}(y_0^{N-1}, \hat{u}_0^{i-1})\right) \tag{2.39}$$

where $n = log_2 N$, i is the index of the considered channel, and $\hat{u}_0^{i-1}$ are the bits previously decoded in SC.

Then, the decision on bit i is

$$\hat{u}_i = \begin{cases} u_i & \text{if } u_i \text{ is frozen} \\ 0 & \text{if } L_n^{(i)} > 0 \\ 1 & \text{if } L_n^{(i)} < 0 \\ ber(\frac{1}{2}) & \text{if } L_n^{(i)} = 0 \end{cases} \tag{2.40}$$

Successive cancellation is given in Algorithm 3. Note that these LRs are associated to the synthetic channels, and therefore their computation, is not straightforward. We focus now on an algorithm that allows to compute these quantities with complexity $O(N \log N)$. The key ideas is that the LRs can be computed recursively.
In fact, we have that

$$\tilde{L}_N^{(2i)}(y_0^{N-1}, \hat{u}_0^{i}) = \frac{1 + \tilde{L}_{N/2}^{(i)}(y_0^{N/2-1}, \hat{u}_{0,e}^{2i-1} \oplus \hat{u}_{0,o}^{2i-1})\tilde{L}_{N/2}^{(i)}(y_{N/2}^{N-1}, \hat{u}_{0,o}^{2i-1})}{\tilde{L}_{N/2}^{(i)}(y_0^{N/2-1}, \hat{u}_{0,e}^{2i-1} \oplus \hat{u}_{0,o}^{2i-1}) + \tilde{L}_{N/2}^{(i)}(y_{N/2}^{N-1}, \hat{u}_{0,o}^{2i-1})} \tag{2.41a}$$

Algorithm 3 SC decoding

for all i $\in \{0, ..., N-1\}$ **do**
    compute $L_n^{(i)}$
    if i is frozen **then**
        $\hat{u}_i \leftarrow u_i$
    else
        if $L_n^{(i)} > 0$ **then**
            $\hat{u}_i \leftarrow 0$
        else if $L_n^{(i)} < 0$ **then**
            $\hat{u}_i \leftarrow 1$
        else
            $\hat{u}_i = ber(\frac{1}{2})$
return $\hat{u}_0^{N-1}$

$$\tilde{L}_N^{(2i+1)}(y_0^{N-1}, \hat{u}_0^{i}) = \tilde{L}_{N/2}^{(i)}(y_0^{N/2-1}, \hat{u}_{0,e}^{2i-1} \oplus \hat{u}_{0,o}^{2i-1})^{1-2\hat{u}_{2i}} \tilde{L}_{N/2}^{(i)}(y_{N/2}^{N-1}, \hat{u}_{0,o}^{2i-1}) \tag{2.41b}$$

A more algorithmic approach, which does not require bit reshuffle and uses LLRs, is given (see [2] and [3]) by the following equations:

$$L_s^{(2i)} = f_-(L_{s-1}^{(2i-[i \bmod 2^{s-1}])}, L_{s-1}^{(2^{s-1}+2i-[i \bmod 2^{s-1}])}) \tag{2.42a}$$

$$L_s^{(2i+1)} = f_+(L_{s-1}^{(2i-[i \bmod 2^{s-1}])}, L_{s-1}^{(2^{s-1}+2i-[i \bmod 2^{s-1}])}, u_s^{(2i)}) \tag{2.42b}$$

for $s = n, n-1, ..., 1$, and $0 \leq i \leq 2^{n-1} - 1$ and

$$L_0^{(i)} \triangleq \ln \frac{W(y_i|0)}{W(y_i|1)} \tag{2.43}$$

$0 \leq i \leq 2^n - 1$ are the channel LLRs, i.e., the ones we can directly compute using the observation of the received symbols and the definition of the original channel W.
Functions f_- and f_+ are defined as follows:

$$f_-(x, y) \triangleq \ln \frac{e^{x+y} + 1}{e^x + e^y} \tag{2.44}$$

$$f_+(x, y, u) \triangleq (-1)^u x + y \tag{2.45}$$

and starting from $u_n^{(i)} \triangleq \hat{u}_i$, we compute quantities

$$u_{s-1}^{(2i-[i \bmod 2^{s-1}])} \triangleq u_s^{(2i)} \oplus u_s^{(2i+1)} \tag{2.46}$$

$$u_{s-1}^{(2^{s-1}+2i-[i \bmod 2^{s-1}])} \triangleq u_s^{(2i+1)} \tag{2.47}$$

In Fig. 2.9, taken from [10], a graphical representation of the recursions is shown.

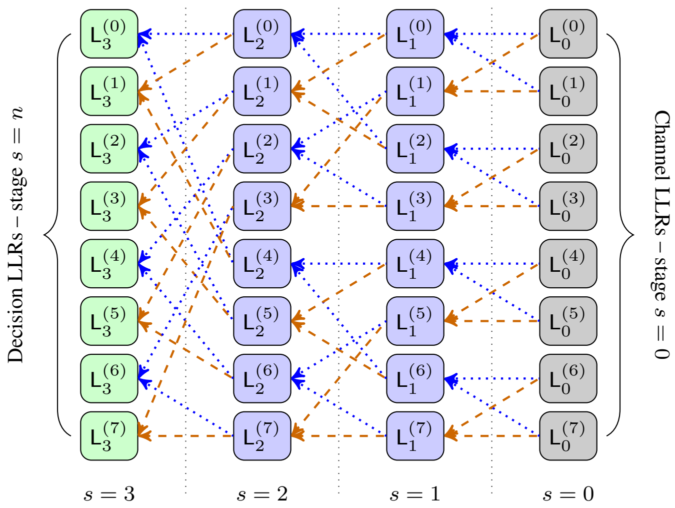


Figure 2.9: The butterfly computational structure of the SC decoder for $n = 3$; blue and orange arrows show f_- and f_+ updates respectively.

We remark that in the particular case of a BEC, since $L_n^{(i)} \in \{0, \pm\infty\}\ \forall i = 0, ..., 2^n - 1$, the SC can fail for the first time at bit i (i.e., we can have a decoding error in i given that all previous bits have been correctly decoded) if and only if $L_n^{(i)} = 0$. This means that if $L_n^{(i)} \neq 0\ \forall i = 0, ..., 2^n - 1$ for sure there will not be any decoding error.
This is clearly not true in general, since we may have some bits for which $L_n^{(i)} > 0$ finite but $\hat{u}_i = 1$, which implies a decoding error.
We note that in Algorithm 3 first we compute $L_n^{(i)}$ and then we evaluate if bit i is frozen or not. Therefore, it may seem that, in case bit i is frozen, $L_n^{(i)}$ is computed and then thrown away, wasting the computational effort employed to determine $L_n^{(i)}$. However, it turns out (and it is also pretty intuitive) that, given the set of frozen bits, the computational effort necessary to determine those information bits for which the computation of $L_n^{(i)}$ could be avoided is the same (or even bigger) as the effort used to compute directly all $L_n^{(i)}$ and throw away those corresponding to information bits.

2.2.5 Performance

A packet is erroneous if at least one of the bits is erroneously decoded. Therefore the packet error event $\mathcal{E}$ is included in the union of events $\mathcal{E}_i$, which is the event corresponding to a decoding error at bit i. Hence,

$$\mathcal{E} \subset \bigcup_{i \in \mathcal{A}} \mathcal{E}_i \tag{2.48}$$

where $\mathcal{A}$ is the set of nonfrozen bits (and consequently $\mathcal{A}^c$ is the set of frozen bits), since a decoding error can only happen at a nonfrozen bit.
The Bhattacharyya parameter gives an upper bound on the decoding error probability at bit i, i.e.,

$$\mathbb{P}(\mathcal{E}_i) \leq Z(W_N^{(i)}) \tag{2.49}$$

Using the union bound, we obtain

$$P_e = \mathbb{P}(\mathcal{E}) \leq \mathbb{P}(\bigcup_{i \in \mathcal{A}} \mathcal{E}_i) \leq \sum_{i \in \mathcal{A}} \mathbb{P}(\mathcal{E}_i) \leq \sum_{i \in \mathcal{A}} Z(W_N^{(i)}) \tag{2.50}$$

Clearly, if a decoding error for a bit occurs, then the whole packet is erroneous. Therefore,

$$\mathcal{E}_i \subset \mathcal{E} \tag{2.51}$$

which implies

$$\mathbb{P}(\mathcal{E}_i) \leq \mathbb{P}(\mathcal{E}) \tag{2.52}$$

and

$$\max_{i \in \mathcal{A}} \mathbb{P}(\mathcal{E}_i) \leq \mathbb{P}(\mathcal{E}) \tag{2.53}$$

For the BEC,

$$\mathbb{P}(\mathcal{E}_i) = Z(W_N^{(i)}) \tag{2.54}$$

if we pessimistically consider than a decoding error happens when an erasure happens at information bit i. In practice, as stated in 8.7, in case of erasure, one can "flip a coin". However, the considerations are analogous. Then, using eq. 2.53 we have

$$\max_{i \in \mathcal{A}} Z(W_N^{(i)}) = \max_{i \in \mathcal{A}} \mathbb{P}(\mathcal{E}_i) \leq \mathbb{P}(\mathcal{E}) \tag{2.55}$$

and therefore

$$\max_{i\in\mathcal{A}} Z(W_N^{(i)}) \leq P_e \leq \sum_{i\in\mathcal{A}} Z(W_N^{(i)}) \tag{2.56}$$

However, a (looser) lower bound is available in general for any B-DMC:

$$\max_{i\in\mathcal{A}} \frac{1}{2}\left(1-\sqrt{1-Z(W_N^{(i)})^2}\right) \leq P_e \tag{2.57}$$

and therefore for any B-DMC we have

$$\max_{i\in\mathcal{A}} \frac{1}{2}\left(1-\sqrt{1-Z(W_N^{(i)})^2}\right) \leq P_e \leq \sum_{i\in\mathcal{A}} Z(W_N^{(i)}) \tag{2.58}$$

From this inequalities it can be derived the following theorem.

Theorem 2.4 *Consider a symmetric B-DMC W and a fixed rate $R < I(W)$. Then*

$$P_e(N, R, \mathcal{A}, \mathbf{u}_{\mathcal{A}^c}) = o(2^{-N^\beta}) \tag{2.59}$$

for any $\beta < \frac{1}{2}$, where set $\mathcal{A}$ contains the best channels for rate R, and frozen bits $\mathbf{u}_{\mathcal{A}^c}$ are arbitrarily fixed.

Therefore the error probability scales roughly as $2^{-\sqrt{N}}$.
A more refined result has been derived in [11]:

$$P_e = 2^{-2^{\frac{n}{2}+\sqrt{n}Q^{-1}\left(\frac{R}{I(W)}\right)+o(\sqrt{n})}} \tag{2.60}$$

where $n = \log_2 N$ and

$$Q(t) \triangleq \frac{1}{\sqrt{2\pi}}\int_t^{+\infty} e^{-\frac{z^2}{2}}dz \tag{2.61}$$

These results are valid for a fixed rate $R < I(W)$, that is, in what is called the *error exponent* regime.
It is also interesting to see what happens if the error probability P_e is fixed and the rate varies. This is called the *scaling exponent* regime, and we have the following theorem [11] [12].

Theorem 2.5 *Let W be a D-BMC. If we require*

$$\sum_{i \in \mathcal{A}} Z(W_N^{(i)}) \leq P_e \tag{2.62}$$

for a fixed $P_e > 0$*, then*

$$N = \Theta\left(\frac{1}{(I(W) - R)^{\mu}}\right) \tag{2.63}$$

for any rate $R < I(w)$*, and where* $3.579 \leq \mu \leq 4.714$ *for any symmetric B-DMC, and in particular* $\mu = 3.6325$ *for the BEC.* μ *is called* scaling exponent *of the code.*

Definition 2.10 *In general we say that a scaling law holds for the error probability* $P_e(N, R, W)$ *of a capacity-achieving code if there exists a function, called* mother curve*, and a constant* $\mu > 0$*, called* scaling exponent*, such that*

$$\lim_{N \to \infty \,:\, N^{1/\mu}(I(W) - R) = z} P_e(N, R, W) = f(z) \tag{2.64}$$

In Fig. 2.10 we show the results of simulations of the error probability for a BAWGNC of rate $R = \frac{1}{2}$ for various block lengths. Since the channel is Gaussian, its estimation is necessary (see section 7.1.2). In order to evaluate the error probability of the code itself, and minimize the error due to channel estimation,the channel is estimated at each SNR $\Gamma = \rho \frac{E_b}{N_0} = \frac{E_b}{N_0} = \frac{1}{\sigma_w^2}$. To evaluate the performance, we compute the capacity of the channel using eq. A.3, and we obtain Table 2.1. We see that the results are quite poor, since, in order to achieve $P_e = 10^{-7}$ for $n = 12$, we must have $\frac{E_b}{N_0} \approx 3.25 \Rightarrow I(W) = 0.7397$. Therefore our rate is still far from the channel capacity. It is also confirmed by comparison with bounds given in [13].

Other performance measurements can be found in [2] and [9]. This is the main drawback of polar codes: in spite of being provably capacity-achieving with low complexity, the finite-length performance with SC decoding are quite poor. For $n \sim 15$ they are still outperformed by LDPC and turbo codes. In order to really become competitive in terms of error probability, one must increase n to values of $n \sim 20$ or $n \sim 30$. But at such high values of n (which *per se* are impractical for many applications, such as mobile transmissions, since they imply packet sizes of the order of the Mbits) the encoding and

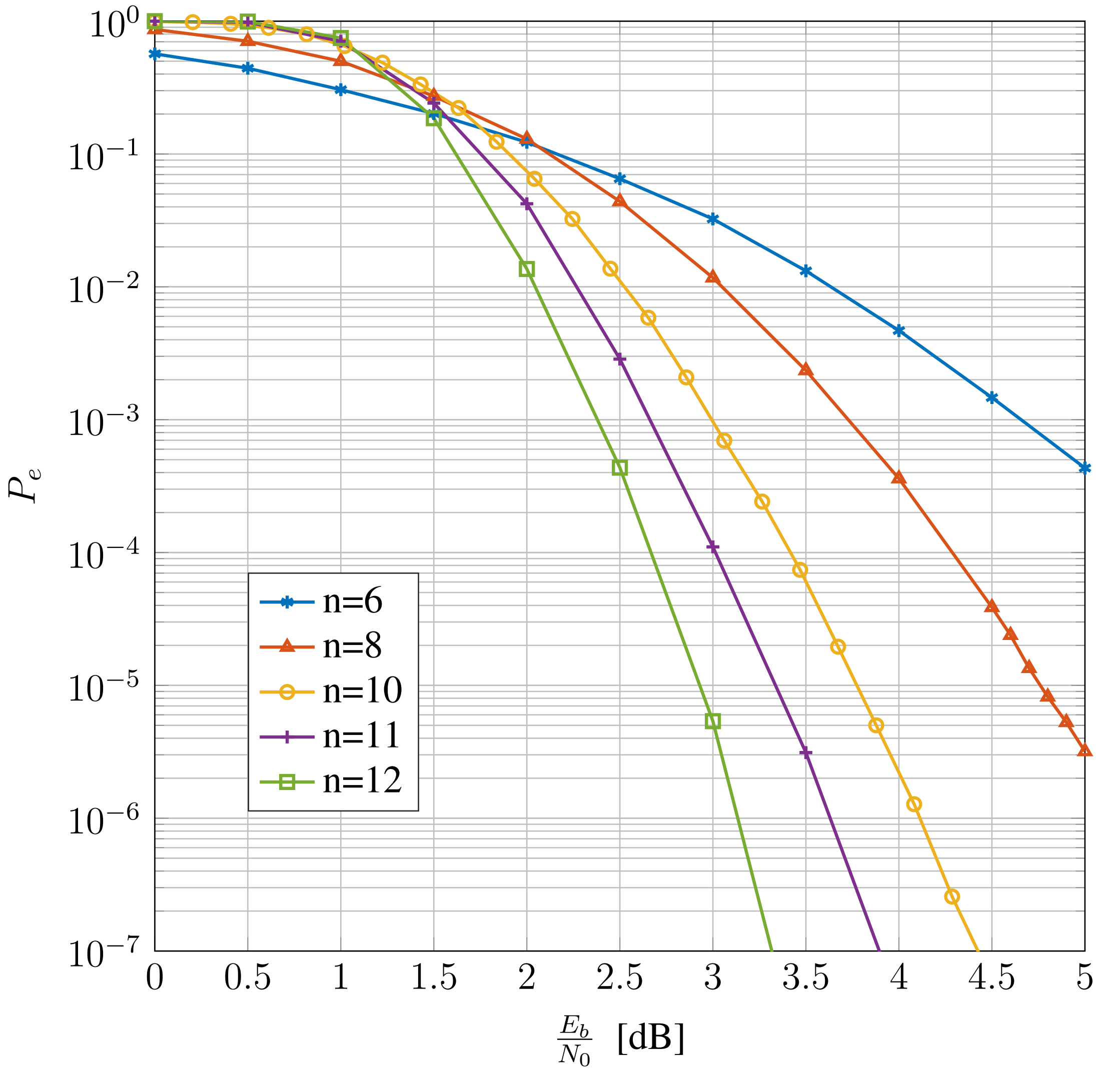


Figure 2.10: channel estimated at each SNR.

decoding processes are very time-consuming.

From these considerations, the necessity of additional techniques to improve the finite-length performance of polar codes emerges.

Table 2.1: capacities for BAWGNC.

$\frac{E_b}{N_0}$ [dB]	σ_w^2	$I(W)$ [bit/ch. use]
0.00	1.0000	0.4860
0.25	0.9441	0.5048
0.50	0.8913	0.5239
0.75	0.8414	0.5432
1.00	0.7943	0.5628
1.25	0.7499	0.5825
1.50	0.7079	0.6023
1.75	0.6683	0.6222
2.00	0.6310	0.6422
2.25	0.5957	0.6620
2.50	0.5623	0.6817
2.75	0.5309	0.7013
3.00	0.5012	0.7206
3.25	0.4732	0.7397
3.50	0.4467	0.7584
3.75	0.4217	0.7766
4.00	0.3981	0.7943
4.25	0.3758	0.8115
4.50	0.3548	0.8281
4.75	0.3350	0.8440
5.00	0.3162	0.8592
5.25	0.2985	0.8736

Chapter 3

Feedback Schemes

3.1 Introduction

In this work we will consider mainly the BEC with erasure probability $\varepsilon = \frac{1}{2}$. The code and the rate R (i.e., the number and indices of the frozen channels) are fixed for the whole transmission, and the packet length $N = 2^n$ is also fixed. The number of information bits, denoted by K, is given by some integer approximation of RN.
In order to simplify the system we also consider the case of full feedback, that is, the sender knows exactly what the receiver received. In appendix B some considerations will be made on the non-full feedback case. The reason of the choice of the binary erasure channel is that it allows some simplifications. First of all, the Bhattacharyya parameters can be computed exactly, and therefore we are able to eliminate the errors introduced by the channel estimation. Moreover, for the BEC successive cancellation is simpler, since the LLRs take value in $\{\pm\infty, 0\}$, and therefore the receiver knows for sure the first bit at which the SC decoder fails, which is in general not the case.
Note that the term *error* in this context indicates an error on the *information* bits, not on the received symbols. Therefore it is a synonym of genie help and genie aid (see section 3.2.1).

3.2 Schemes

3.2.1 General Setting

We introduce now the different schemes we developed in order to improve the finite-length performance of polar codes. These schemes are presented in an increasing level of complexity and efficiency (in the sense of how many bits are retransmitted for a given error pattern in order to decode a certain packet). Clearly, if a packet has no errors, then it is immediately decoded, and no retransmission scheme is needed.

A common way of seeing this system is to think of *genie-aided SC decoding*. We consider a BEC. Each time the receiver is unable to decode a bit (since it is a BEC, this is known *a priori*), it invokes the help of a genie, which provides a *genie help* (or *genie aid*), consisting in the correct bit that should be put at the position where the SC decoder got stuck. Hence, the SC decoder is able to correctly proceed, until it halts again because it is unable to decode. It will then invoke the next genie help, and so on until the end of the packet, that will therefore be entirely decoded.

Algorithm 4 shows the genie-aided SC decoder, where g_i is the genie aid for bit i and is found by the receiver in the following packet.

From a practical point of view, what happens is that the sender, which knows exactly what the receiver receives, tries to decode the received packet. Each time the SC decoder fails, the sender, which obviously knows the correct bits (since it is the packet it just sent), uses the corresponding correct bit to allow the SC decoder to go on in the decoding process, and records it, until the end of the packet. Note that it is thanks to the full-feedback that we are able to put all the genie helps regarding packet i into packet $i + 1$. In fact, if only the received packet is available, by using SC decoder we are able to know only the position of the first genie aid, since nothing can be said about the subsequent bits. This is precisely why we speak of *successive* cancellation: in order to decode a bit, we must have already decoded all the previous ones, and if we are unable to decode a bit, we cannot decode the successive ones. We could certainly guess, but since SC uses the already decoded bits to decode the new ones, the result is that decoding errors propagate. This means that if an error occurs at some bit, the decoding error probability for the subsequent bits will significantly increase. At the following error, it

Algorithm 4 Genie-aided SC decoding

for all i $\in \{0, ..., N-1\}$ **do**
  compute $L_n^{(i)}$
  if i is frozen **then**
    $\hat{u}_i \leftarrow u_i$
  else
    if $L_n^{(i)} > 0$ **then**
      $\hat{u}_i \leftarrow 0$
    else if $L_n^{(i)} < 0$ **then**
      $\hat{u}_i \leftarrow 1$
    else
      $\hat{u}_i \leftarrow g_i = u_i$
return $\hat{u}_0^{N-1}$

will increase again, in a sort of "cascade effect". Moreover, the bit most likely to be wrong are at the beginning. Therefore decoding by guessing is not a practically viable option.

In general, all genie aids used for the decoding of packet i will be put by the sender (which knows what the receiver received because of the full feedback hypothesis) at the beginning of packet $i+1$, in the same order as they were invoked in packet i (i.e., the first genie help used to decode packet i will be the first retransmitted bit of packet $i+1$). Then, genie aids for packet $i+1$ will be put at the beginning of packet $i+2$, and so on. The reason why we choose to put them at the beginning is to take advantage of the successive cancellation decoding, which allows to use the information we decode until the first error occurs, without having to wait for the whole packet to be decoded. However, on the other hand, it is also true that in general the first bits are the ones with highest probability to be incorrectly decoded.

When the receiver decodes a bit that corresponds to a genie help, we say that that genie help (or error) has been *resolved*. Then, the receiver resolves the j-th genie help of packet i, with $j = 0, ..., M-1$ and $M \leq K$ total number of genie helps, using the j-th unfrozen bit of packet $i+1$.

All the proposed schemes decode the received packets by preserving the arrival order, that is, if packet s arrives before packet t, then it will also be completely decoded before packet t, or at most at the same time.

3.2.2 Metrics of Interest

These schemes, for the BEC, manage to get packet error probability $P_e = 0$. On the other hand we allow for some average delay D (which in the limit may even be ∞), that is measured as the number of additional packets that must be received in order to successfully decode a given packet (that is, if a packet is received and successfully decoded at reception, it has delay $D = 0$). We also allow for rate reduction because of the retransmitted bits, which do not carry any useful information. This naturally introduces the notion of effective rate R_{eff}. The aim is to develop a scheme that does not make D excessively big or R_{eff} excessively small (for the application of interest). These are therefore the two main design parameters, and the metrics we are interested to study. If we denote by K the number of informative bits, then the average effective rate is

$$R_{eff} = \frac{K - M}{N} = R - \frac{M}{N} \tag{3.1}$$

where M is the average number of retransmitted bit (which is equal to the average number of genie helps) per packet, and $R = \frac{K}{N}$ is the "true" nominal rate.
We remark that, strictly speaking, in the computation of the effective rate one should also consider the fact that, for a nonzero average delay, there will always be some uncoded packets still pending in the decoding queue once the transmission has finished. A way to solve this problem is to pad the transmission by adding empty packets (e.g., packets of all 0's) after the last useful packet (packet that carries information) until the decoding queue is emptied and all useful packets have been decoded. Clearly, this system causes an overall rate reduction. In fact, the new effective rate becomes

$$R'_{eff} = \frac{q(K - M)}{(q + cD)N} \tag{3.2}$$

where q is the number of packets carrying information bits and cD is the number of padding packets, which is linear in the delay, according to some constant $c > 0$ which accounts for the delay variance. The numerator is the same since the number of information bits does not change. By taking the inverse, we get

$$\frac{1}{R'_{eff}} = \frac{1}{R_{eff}} + \frac{cDN}{q(K-M)} \tag{3.3}$$

where the last term accounts for the rate reduction due to padding. However, we see that for for $q \to \infty$, $\frac{cDN}{q(K-M)} \to 0$ and hence $R'_{eff} \Rightarrow R_{eff}$. Therefore, we will assume that our transmission will consist of a number of information packets big enough to make this rate reduction negligible.

Measurability of the Delay

We define e_i the random vector associated to the number and position of errors in packet i (e.g., a vector of 0's and 1's according to the position of the errors in the packet), and we consider the random process $\{e_j\}_{j \geq i}$. Clearly, this process depends only on the channel, which is supposed memoryless and always identical, and therefore e_i are i.i.d. random vectors. Moreover it is clear that the delay D_i associated to packet i depends solely on this error process from packet i on. Hence, we can write

$$D_i = f(e_i, e_{i+1}, ...) \tag{3.4}$$

and

$$D_{i+1} = f(e_{i+1}, e_{i+2}, ...) \tag{3.5}$$

for some function f. This clearly shows that D_i and D_{i+1} are not independent, but since e_i are i.i.d., this implies that D_i and D_{i+1} are identically distributed, and in particular they have same mean and variance. Then, under the assumption that $Cov(D_i, D_j) \to 0$ as $|j - i| \to \infty$ (which is reasonable since intuitively the delays associated to packets distant in time are less dependent), the weak law of large numbers holds (see [14]), which justifies the correctness of the empirical mean.

For what it concerns the delay associated to information bits, we consider that bits are decoded only when the whole packet is decoded. Then, since each information bit is associated to one and only one packet, we empirically evaluate the quantity

$$\frac{\sum_i m_i D_i}{\sum_i m_i} \tag{3.6}$$

where m_i is the number of information bits associated to packet i (in this case we consider that a packet is decoded as a whole, and therefore the delay of a bit is the delay of the packet to which it belongs). But if we expand each term $m_i D_i = \underbrace{D_i + ... + D_i}_{m_i \text{ times}}$ we still get a sequence that satisfies the previous hypothesis, and therefore we still get convergence in probability to the same average $\mathbb{E}[D]$.

If instead we consider that decoding and information retrieval can be performed bit-wise, then $\mathbb{E}[D]$ represents an upper bound to the average delay before decoding each bit, since it is the maximum delay associated to bits of the same packet. Mathematically, if we call $h(m)$ the function that associates to bit m its delay before decoding, then we are partitioning all information bits into intervals (corresponding to packets) and we are considering function $g(m) \triangleq \max\{h(k) : k \in \mathcal{I}_m\}$, where $\mathcal{I}_m$ is the interval (packet) to which bit m belongs. $g(m)$ is the delay associated to decoding of packet m and therefore, as stated before, the average of g is $\mathbb{E}[D]$, and clearly by definition $g \geq h$. Hence, the average of h is smaller or equal than $\mathbb{E}[D]$.

Measurability of the Effective Rate

R_{eff} is determined by the genie distribution of each packet, which is in turn determined by the erasure pattern on the received bits. The erasure patterns, that is, the joint probability of erasures at different bits, for different packets are independent (since the channel is memoryless), and identically distributed. Therefore the number of genie helps in different packets are i.i.d. random variables with finite mean and variance, and the laws of large numbers straightforwardly applies.

3.2.3 Scheme I

The first and simplest scheme consists in putting into a buffer (decoding queue) all packets that have at least one error, and decode all of them as soon as we receive a packet without errors, i.e., a packet that we are able to fully decode at reception. In fact, in such a case, the genie resolution propagates from a packet to the previous one, and so on, and all packets are decoded completely.

3.2.4 Scheme II

A more complex system consists in successfully decoding packet i if all the bits of packet $i+1$ that correspond to retransmitted bits of packet i are successfully decoded. Otherwise, we put packet i in a buffer, and wait until for a packet $j > i+1$ the first error occurs after the retransmitted bits. In this case, we are able to completely decode packet $j-1$, but this allows to decode packet $j-2$ and so on until packet i.
For the algorithm, we need a variable, `previous_genie_helps`, that keeps track of how many bits of the current packet are bits retransmitted, corresponding to the genie helps of the previous packet.

Algorithm 5 scheme II

```
for all current packet i do
    t ← number of bits successfully decoded
    if no genie helps needed to decode packet i then
        decode packet i and all packets in the buffer
        empty buffer and compute metrics for decoded packets
    else if previous_genie_helps ≤ t then
        decode all packets in the buffer
        compute metrics
        put packet i in the buffer
    else
        previous_genie_helps ← total number of genie helps necessary to completely decode packet i
        put packet i in the buffer
```

3.2.5 Scheme III

By carefully considering the previous feedback scheme, we see that there is an inefficiency. In fact, in order to decode packet i, we wait until we are able to completely decode a packet, but this is not necessary, since in order to decode packet i we only need to decode, in packet j, only those bits that are linked to errors in packet i. As an example, consider the case in which there are, let us say, 10 genie helps in packet i. Then, in packet $i+1$ the first 10 bits will be retrasmitted bits referring to packet i. Now, let us suppose that in packet $i+1$ 2 errors (genie helps) occur: one involving one of the 10 retransmitted bits and the other not involving these bits. Then, in packet $i+2$ we will have the first 2 bits that are retrasmitted bits of packet $i+1$. With the previous system, we decode packet i and $i+1$, at the same time, if and only if possible errors on packet $i+2$ involve only bits from the third one on. Otherwise, we are obliged to wait at least for the reception of packet $i+3$. But suppose the second bit is erroneous, while the first one is not, that is, we are able to decode only the first bit. This first bit, however, allows to correct the first error of packet $i+1$, which was the only error involving the 10 bits of packet $i+1$ referring to packet i. Hence, by using the first bit of packet $i+2$ we are able to successfully decode the first 10 bits of packet $i+1$, which makes possible the successful decoding of packet i, without waiting for packet $i+3$. This is the rationale behind this last feedback scheme.

In Algorithm 6 we use a stack $\mathscr{S}$ to store data for each packet, namely its progressive number (to compute the delay) d, the position where the SC decoder got stuck (position of the missing genie help) p and a counter which counts the number of genie helps used for that packet g.

Algorithm 6 is as seen by the receiver. It is inefficient, but has the advantage of being clear and intuitive.

For the simulations, a much more efficient and faster algorithm is employed, which takes advantage of the full feedback (therefore for each packet we immediately know the positions of all genie aids). The idea is to keep track of how many genie helps at any instant still remain to be resolved in each packet before completely decoding it. This metric is updated at each new packet received for all packets still in the buffer. For this purpose we can use an array `bits_needed(i)`, which gives how many genie helps at each instant still need to be resolved to completely decode packet i, built using information returned by the SC decoder.

Algorithm 6 scheme III Receiver

$i = 0$
while 1 **do**
    $j = 0$
    $g = 0$
    $d = i$
    while $j \leq 2^n - 1$ **do**
        try decode bit j of packet i
        if decoding is successful **then**
            DECODE($\hat{u}_j$,i)
            $j \leftarrow j + 1$
        else
            put packet i into stack $\mathscr{S}$: $p \leftarrow j$, $g \leftarrow g + 1$
            $j \leftarrow 2^n + 1$
    if $j = 2^n$ **then**
        store g and $delay = i - d$
    $i \leftarrow i + 1$

Optimality of Scheme III

One can think of some simple cases where schemes I and II fail (that is, they yield an infinite delay), whereas scheme III succeeds. Consider the following example: all packets need exactly two genie helps at the second and third information bits, that is, for each packet we are able to immediately decode only one bit. Schemes I and II never decode any packet, whereas scheme III successfully decode all of them with delay $D = 2$.

In general, scheme III satisfies the following property.

Proposition 3.1 *Let us assume that an infinite number of packets is transmitted. Then, necessary condition for a packet i to never be decoded is that from some packet $j \geq i$ on, all packets have no bits successfully decoded, i.e., the first genie help is needed at the first information bit.*

Algorithm 7 DECODE procedure for Algorithm 6

```
procedure DECODE(û,t)
    if 𝒮 = ∅ then return
    else
        take top packet from 𝒮: this is current packet
        l ← g
        use û to resolve bit û_l
        decode(û_l)
        l ← l + 1
        while l ≤ 2^n do
            try decode bit l of current packet
            if decoding is successful then
                decode(û_l)
                l ← l + 1
            else
                put current packet into stack 𝒮: p ← l, g ← g + 1
                l ← 2^n + 1
        if j = 2^n then
            store metrics g and delay = i − d
```

Proof 3.1 *Let us suppose that we are able to decode at least one bit for some packet $j \geq i$. Then, referring to Algorithm 8, in line 7 we have $t \geq 1$. Therefore, either packet $j-1$ is completely decoded, or t genie helps are resolved. Each genie help corresponds to one bit still not decoded, therefore resolving t genie helps in packet $j-1$ means that at least $t \geq 1$ additional bits are successfully decoded in packet $j-1$, and therefore in line 15* `bits_needed(j)` *strictly decreases. Moreover, in line 16 we have $t \geq 1$. We can repeat the reasoning until we arrive to packet i, for which too we have that* `bits_needed(i)` *strictly decreases.*
Now, by contradiction, since the number of transmitted packets is infinite, and there exists no packet $j \geq i$ such that the condition of the theorem holds, there is an infinite number of packets for which at least one bit is successfully decoded. Hence, we can

Algorithm 8 scheme III simulation

for all current packet i **do**
  if packet i completely decoded **then**
    empty buffer and compute metrics for decoded packets
  else
    put packet i at the end of the buffer
    `bits_needed(i)` $\leftarrow$ genie invocations necessary to completely decode packet i
    $t \leftarrow$ bits successfully decoded in packet i
    for all packets j in the buffer, starting from i-1 to the oldest one **do**
      if $t \geq$ `bits_needed(j)` **then**
        all packets of index $\leq j$ are successfully decoded
        compute metrics for those in the buffer
        remove them from the buffer
      else if $t > 0$ **then**
        t genie helps are resolved in packet j
        `bits_needed(j)` $\leftarrow$ `bits_needed(j)` $-t$ ▷ this allows to continue with SC decoder in packet j and decode more bits
        $t \leftarrow$ number of *additional* bits decoded in packet j
      else
        do nothing ($t = 0$)

repeat the first part of the proof an infinite number of times, and therefore at some point it will necessarily be `bits_needed(i)`$= 0$*, which implies packet* i *successfully decoded.*

However, the condition of proposition 3.1 implies that from packet j on nothing is decoded and therefore it is clear that there exists no feedback scheme able to retrieve some information from such a scenario.
This is precisely the sense in which we can say that scheme III is optimal.
Because of this optimality, our analysis will be mainly focused on scheme III. However it is also important to remark that optimality comes at the cost of a much greater

algorithmic complexity of scheme III with respect to schemes I and II.

3.3 Implementation

The simulations have been carried out using programs in `C` and `C++`. In particular `C` has been used in order to derive the maximum optimization and simulation speed, since many simulations require a significant computational effort.
Since the code is linear, performance and metrics do not change if we transmit all-0s packet (i.e., $u_i = 0 \; \forall i$) instead of randomly generated bits. Clearly the all-0s information packets is encoded in the all-0s codeword, and therefore there is no necessity to simulate also the encoding operations, but just the SC decoding under transmission of the all-0s codeword.
The metrics of our interest have been estimated via a suitable number of Monte Carlo trials, of the order of $\approx 10^6$ packets. For computational reasons a limit on the maximum delay has been set (its value varies according to the simulation scenario). This results in an underestimation of some delays (since those trials that would have resulted in a delay bigger that the threshold have not been considered in the empirical mean). This also implies an outage probability of the delay, that gives an empirical indication of the probability of a packet being decoded with a delay bigger than the threshold. Therefore, the expected delay must be evaluated by carefully considering also the outage probability, since the bigger this probability is, the more significant the bias for underestimation is.
The theoretical analysis has been carried out mainly using `Matlab`.

Chapter 4

Performance Analysis of Feedback Schemes

4.1 Introduction

A very quick calculation allows to conclude that scheme I is clearly outperformed by schemes II and III, since it only decodes when no genie helps are needed, which is practically impossible for rates that approach the channel capacity. Therefore the meaningful comparison is certainly between these two latter schemes.

4.2 Delay

Figures 4.1, 4.2 and 4.3 show the delay comparison between schemes II and III for various packet lengths. As expected, scheme III yields a significant performance improvement over scheme II. Moreover, the performance increases as N increases, thanks to polarization, since for, the same nominal rate we obtain a lower delay, which confirms the effectiveness of polarization also for the delay.

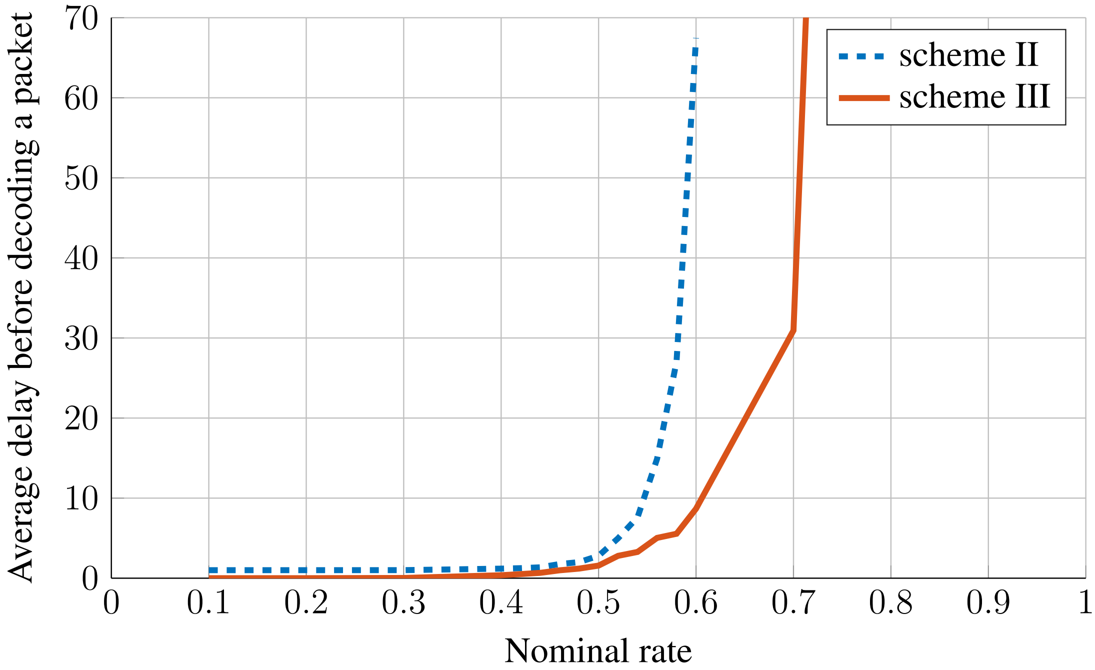


Figure 4.1: performance comparison of schemes II and III for $N = 128$.

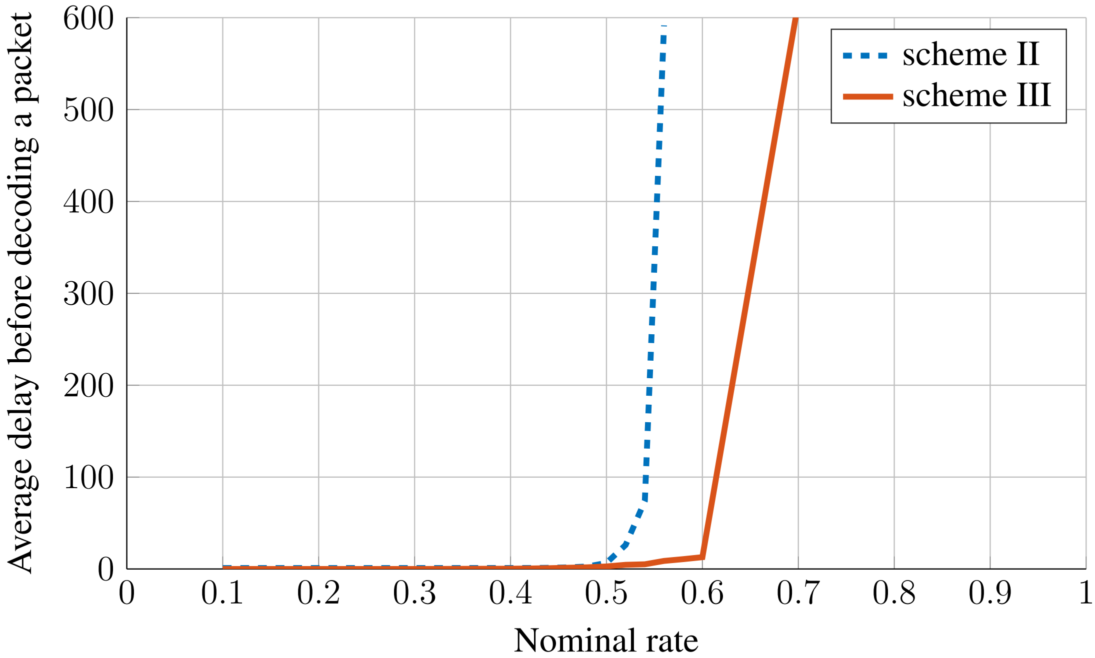


Figure 4.2: performance comparison of schemes II and III for $N = 512$.

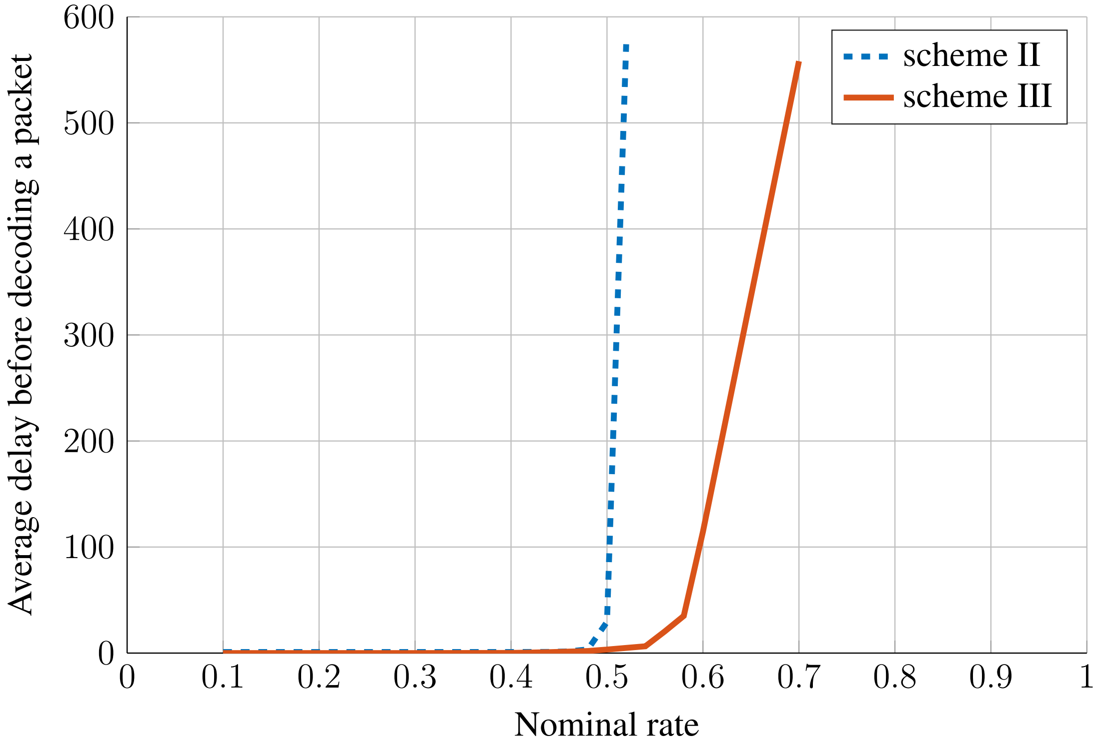


Figure 4.3: performance comparison of schemes II and III for $N = 2048$.

4.2.1 Impact of Polarization on the Delay

We grouped together data for scheme III in Fig.4.4 to underline the effect of polarization: lower N allows to get finite delay also for $R > C$ (even $R = 0.7$). This is due to the fact that for low N polarization is still "weak", that is, there are few very bad and very good channels. This means that we can take as nonfrozen a number of channels bigger than the one dictated by capacity ($C = 1 - \varepsilon = 0.5$), and still get acceptable channels.

On the other hand, for N big, polarization is stronger, and therefore as soon as the rate is bigger than capacity, very bad channels (i.e., $Z^{(i)} \approx 1$) are unfrozen, which increases the delay. Moreover, because of the structure of polarization, bad channels are mainly concentrated at the beginning (i.e., they have low indices), where the SC starts decoding. This means that if we unfreeze a very bad channel, it is likely that it will be in one of the first positions, and therefore the number of bits successfully decoded for each packet upon reception is very small, which greatly increases the delay.

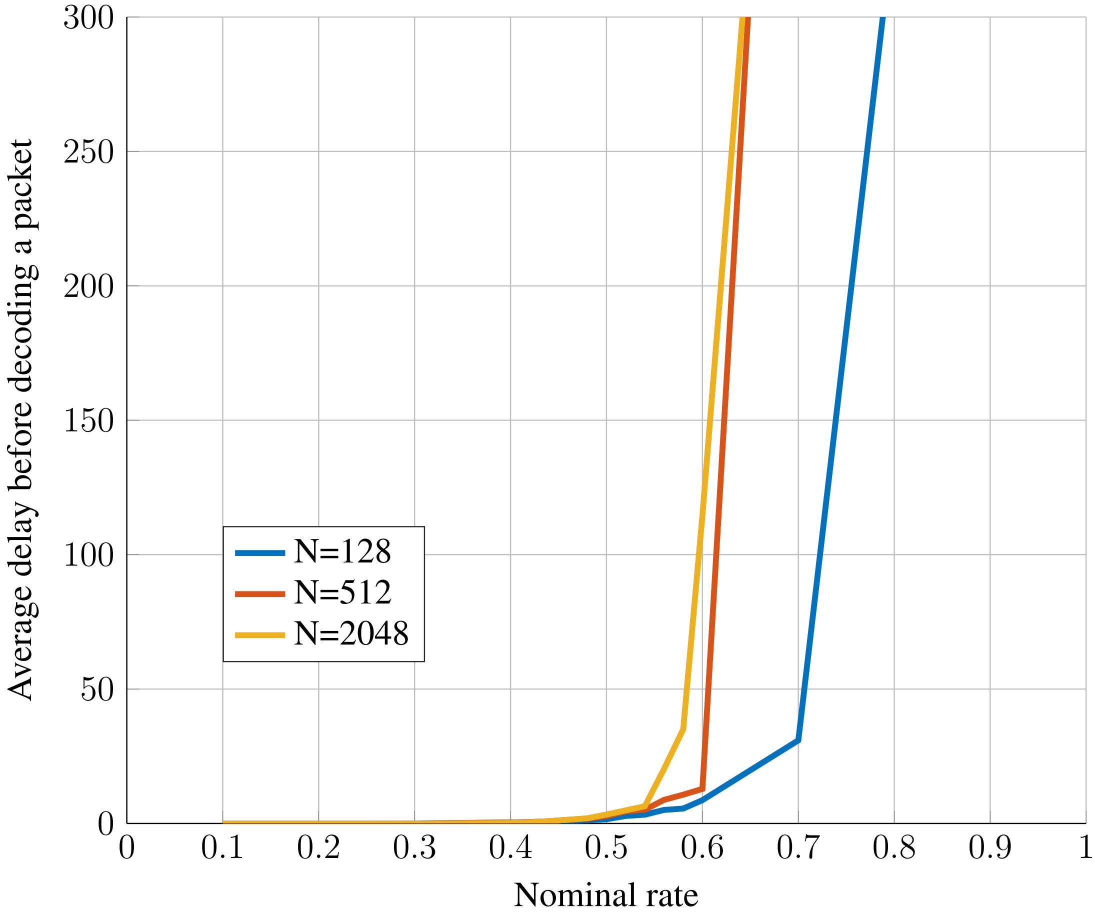


Figure 4.4: polarization for scheme III.

4.2.2 Outage Probability

As already remarked, it is meaningful to take into consideration the outage probability, that is the probability that a packet has a delay bigger than some fixed threshold. This is what is shown in Fig. 4.5.

As expected, for a given rate the outage probability of delay is lower for scheme III with respect to scheme II.

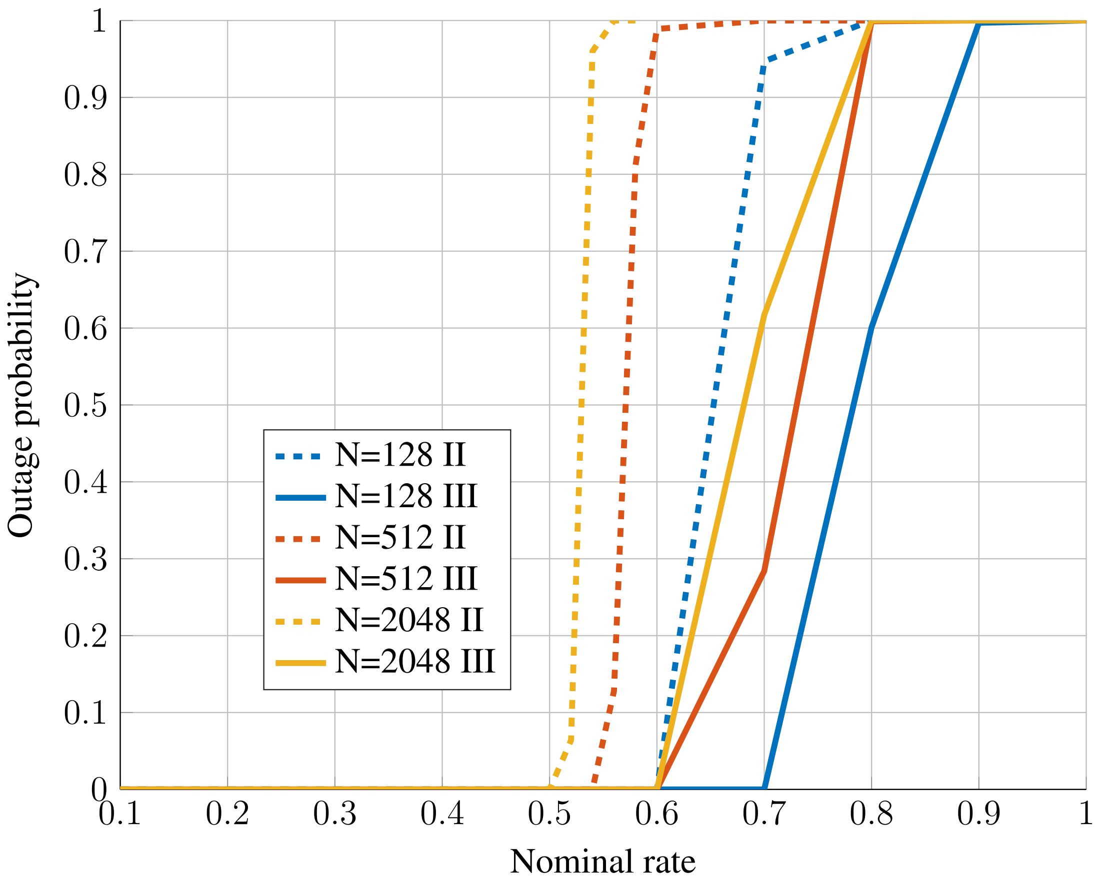


Figure 4.5: outage probability of delay for schemes II and III, threshold at 2000.

4.3 Effective Rate

In Fig. 4.6 the effective rate is shown. It is clearly the same for schemes I, II and III since it depends on the bit retransmission policy, which is the same for the two schemes (which differ only in how these bits are used). As expected, the effective rate tends to the channel capacity $C = 0.5$ if the nominal rate increases. For example, at rate $R = 1$ we transmit on all the channels (no frozen bits), which means that on average half of the bits will need a genie help, and therefore half of the bits of each packet will be retransmitted bits. However, this is not a way to achieve capacity, since the price we have to pay for $R_{eff} = C$ is a delay $D \to \infty$. Again, we see the advantage of polarization, as the effective rate for a fixed nominal rate increases as N increases.

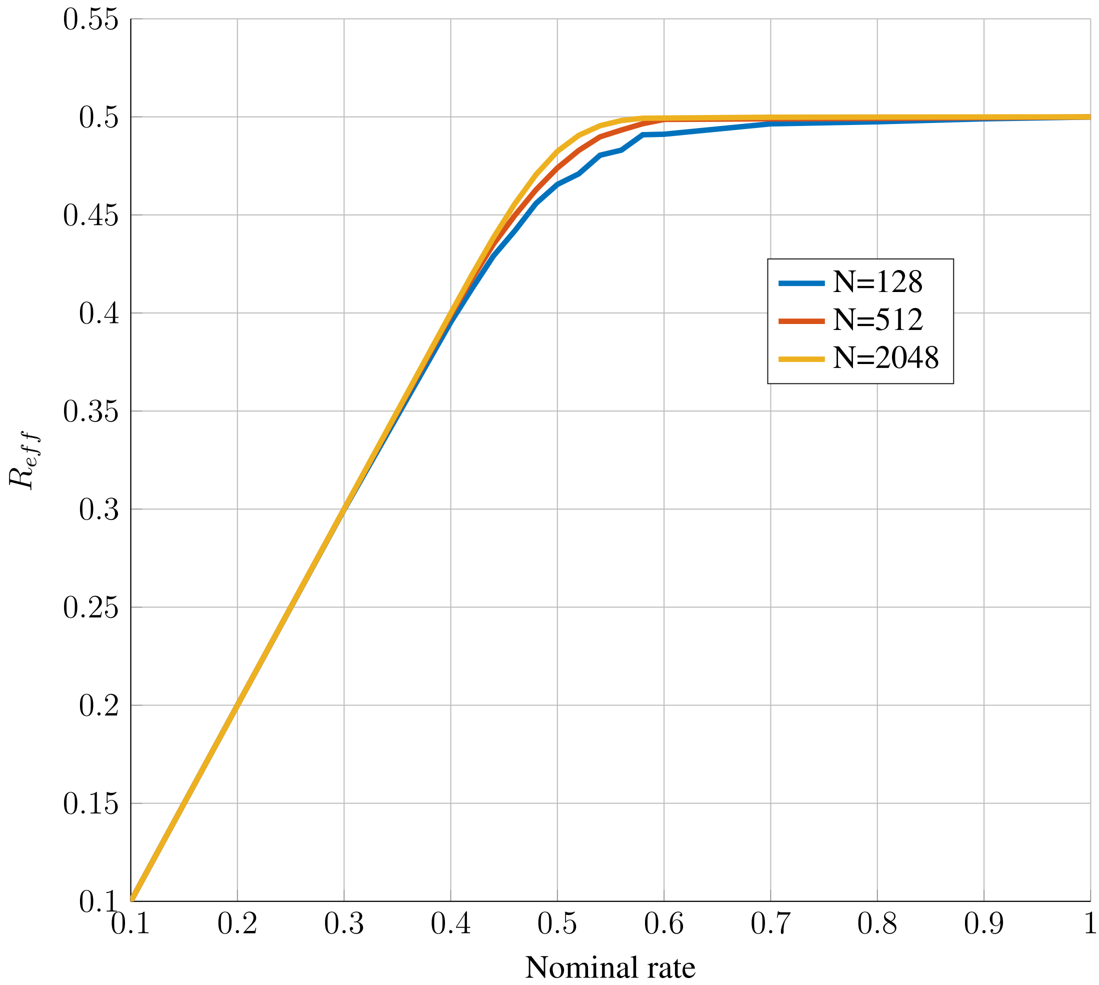


Figure 4.6: effective rate for various N.

4.3.1 Design Parameters

It is interesting for design purposes to evaluate the relationship between the delay and effective rate. The question is the following: we fix a packet size N. Then, given a maximum delay, what is the highest effective rate it can be obtained? The result is shown in Figures 4.8 and 4.7.

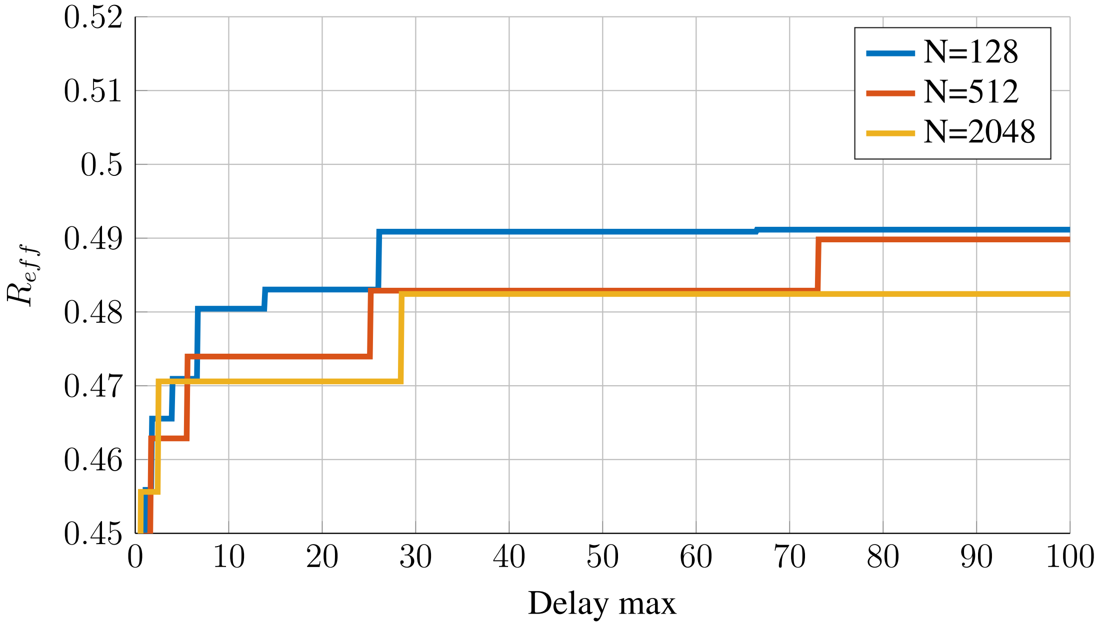


Figure 4.7: maximum R_{eff} for a given delay for scheme II.

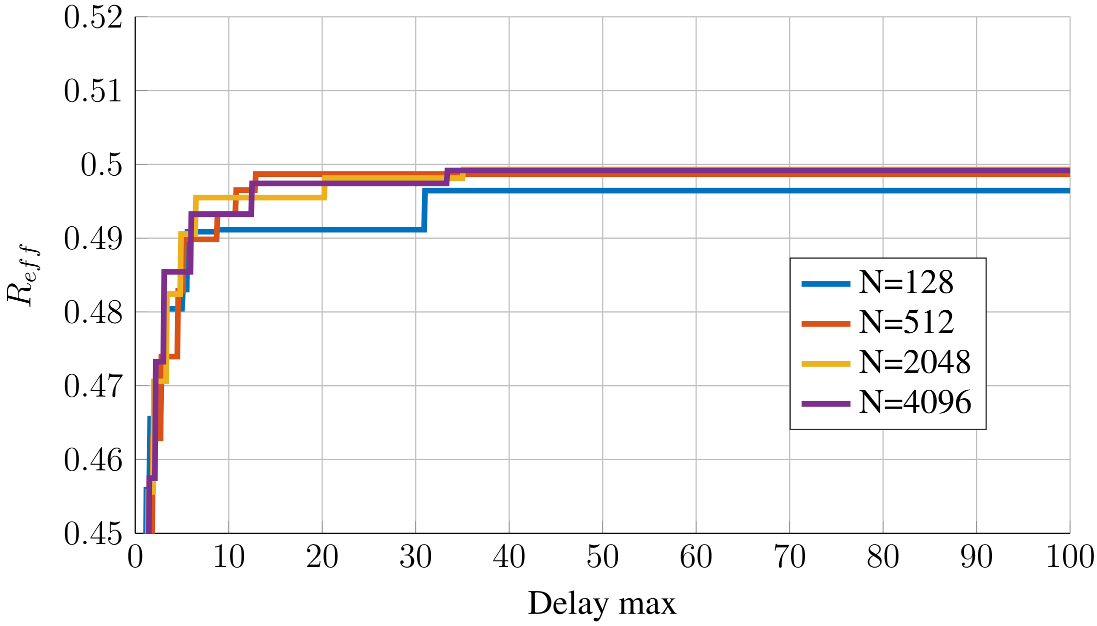


Figure 4.8: maximum R_{eff} for a given delay for scheme III.

Again, scheme III performs better than scheme II as is reaches higher effective rates at lower delays.
Perhaps surprisingly, it appears that polarization does not provide great advantage in this scenario. However this is easily understood if we consider that, as remarked before, a lower polarization allows, for a given delay (see for example Fig. 4.4), to reach a higher nominal rate, which in turns allows to have a higher effective rate (Fig. 4.6).
Nevertheless, in Fig. 4.8 we see that if we want to achieve an effective rate very close to capacity, then by increasing N we are able to do so with a maximum delay smaller with respect to the case with N small. This is not shown in 4.7, due to the delay threshold we have set in the simulation, but if we allowed for a potentially infinite delay, we would see the same behavior.

Chapter 5

Theoretical Modeling of Feedback Schemes

5.1 Introduction

We now focus on the theoretical modeling of feedback schemes. Since there is an increase in efficiency of the schemes, the analysis will be developed more thoroughly for scheme III. In fact, from a practical point of view, only the last one should be considered. For the models, we assume that errors in the synthetic channels are independent, which, experimentally, gives a pretty tight upper bound on the delay (see Fig. 5.6).
We use shorthand notations

$$\begin{aligned} Z^{(i)} &\triangleq Z(W_N^{(i)}) \\ \overline{Z^{(i)}} &\triangleq 1 - Z^{(i)} \end{aligned} \tag{5.1}$$

5.2 Distribution of the Number of Genie Helps

A question that naturally arises is what is the distribution of the number of genie helps in a packet and if it can be modeled in some way. Let us denote by X_i the indicator random variable

$$X_i = \begin{cases} 1 & \text{if a genie help is needed at bit } i \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

for $i = 0, ..., N-1$.
We remind that

$$\{X_i = 1\} \equiv \{\text{error at bit } i\} \bigcap \left(\bigcap_{j=0}^{i-1} \{\text{bit } j \text{ correctly decoded}\} \right) \tag{5.3}$$

since we are dealing with SC cancellation, and therefore we try to decode bit i if and only if all previously bits have been already decoded. Moreover, since we are dealing with genie-aided SC, all previous bits have been correctly decoded.
Therefore, for the BEC,

$$\mathbb{P}[X_i = 1] = Z^{(i)} = \mathbb{E}[X_i] \tag{5.4}$$

Clearly, eq. 5.4 holds for nonfrozen bits, whereas for the frozen bits we can assume that $X_i = 0$ w.p. 1 since for sure a genie help will not be needed to decode a frozen bit.
The number of genie aids in a packet is given by $S = \sum_{i=0}^{N-1} X_i$, and therefore

$$\mathbb{E}[S] = \sum_{i \in \mathcal{A}} Z^{(i)} \tag{5.5}$$

where $\mathcal{A}$ is the set of nonfrozen indices.
For what it concerns the second order statistics, we have that

$$\mathrm{Var}(S) = \sum_{i,j \in \mathcal{A}} \mathrm{Cov}(X_i, X_j) = \sum_{i \in \mathcal{A}}^{K-1} \mathrm{Var}(X_i) + 2 \sum_{\substack{i,j \in \mathcal{A} \\ i<j}} \mathrm{Cov}(X_i, X_j) \tag{5.6}$$

Now, for a frozen bit $\mathrm{Var}(X_i) = 0$, whereas for a nonfrozen bit

$$Var(X_i) = \mathbb{E}[X_i^2] - (\mathbb{E}[X_i])^2 = Z^{(i)} - (Z^{(i)})^2 \tag{5.7}$$

To compute the correlation matrix, it is more suitable to denote the channels (and the bits) using the polarization sequence.
Let $\mathbf{s}, \mathbf{t} \in \{+, -\}^m$ and

$$C_m^{(\mathbf{s},\mathbf{t})} \triangleq \mathrm{Cov}(\tilde{X}_m^{(\mathbf{s})}, \tilde{X}_m^{(\mathbf{t})}) \tag{5.8}$$

where $\tilde{X}_m^{(\mathbf{s})}$ means that we are referring to the bit corresponding to sequence $\mathbf{s} \in \{+, -\}^m$ in the block of size 2^m. Therefore our aim is to compute this quantity for $X_i = \tilde{X}_n^{(\mathbf{s}(i))}$, where $\mathbf{s}(i)$ is the polarization sequence corresponding to index i (incidentally, we remark that the main advantage of using the polarization sequences instead of

the indices is that sequences are independent on the indexing and ordering of the channels).
Then correlation matrix is computed using a single step recursion [15]:

$$C_n^{(\mathbf{s}-,\mathbf{t}-)} = 2(1 - Z_{n-1}^{(\mathbf{s})})(1 - Z_{n-1}^{(\mathbf{t})})C_{n-1}^{(\mathbf{s},\mathbf{t})} + {C_{n-1}^{(\mathbf{s},\mathbf{t})}}^2 \tag{5.9a}$$

$$C_n^{(\mathbf{s}-,\mathbf{t}+)} = 2(1 - Z_{n-1}^{(\mathbf{s})})Z_{n-1}^{(\mathbf{t})}C_{n-1}^{(\mathbf{s},\mathbf{t})} - {C_{n-1}^{(\mathbf{s},\mathbf{t})}}^2 \tag{5.9b}$$

$$C_n^{(\mathbf{s}+,\mathbf{t}-)} = 2Z_{n-1}^{(\mathbf{s})}(1 - Z_{n-1}^{(\mathbf{t})})C_{n-1}^{(\mathbf{s},\mathbf{t})} - {C_{n-1}^{(\mathbf{s},\mathbf{t})}}^2 \tag{5.9c}$$

$$C_n^{(\mathbf{s}+,\mathbf{t}+)} = 2Z_{n-1}^{(\mathbf{s})}Z_{n-1}^{(\mathbf{t})}C_{n-1}^{(\mathbf{s},\mathbf{t})} + {C_{n-1}^{(\mathbf{s},\mathbf{t})}}^2 \tag{5.9d}$$

with $C_0 = \varepsilon(1 - \varepsilon)$.
If we index the bits from 0 to $2^n - 1$ and we assign to the polarization in the $-$ direction the index transformation $i \to 2i$ and to the one in the $+$ direction the index transformation $i \to 2i + 1$, we obtain

$$C_n^{(2i,2j)} = 2(1 - Z_{n-1}^{(i)})(1 - Z_{n-1}^{(j)})C_{n-1}^{(i,j)} + {C_{n-1}^{(i,j)}}^2 \tag{5.10a}$$

$$C_n^{(2i,2j+1)} = 2(1 - Z_{n-1}^{(i)})Z_{n-1}^{(j)}C_{n-1}^{(i,j)} - {C_{n-1}^{(i,j)}}^2 \tag{5.10b}$$

$$C_n^{(2i+1,2j)} = 2Z_{n-1}^{(i)}(1 - Z_{n-1}^{(j)})C_{n-1}^{(i,j)} - {C_{n-1}^{(i,j)}}^2 \tag{5.10c}$$

$$C_n^{(2i+1,2j+1)} = 2Z_{n-1}^{(i)}Z_{n-1}^{(j)}C_{n-1}^{(i,j)} + {C_{n-1}^{(i,j)}}^2 \tag{5.10d}$$

A first approximation can be obtained by using a normal distribution, with mean and variance as above. We can also evaluate the impact of the covariance. However, we can already guess that for low rates, that yield a low number of genie helps, the Gaussian distribution will not fit, since it can also take negative values, whereas the number of genie helps can only be positive. The simplest distribution that takes positive integer values is the Poisson distribution. One can also consider a binomial distribution, or, even better, a Poisson-binomial distribution, described by eq. 5.14.
In Fig. 5.1 we show the experimental distribution and its approximations for a rate $R = 0.4$. These figures refer to simulation with $n = 12$ and 10^4 Monte Carlo trials.

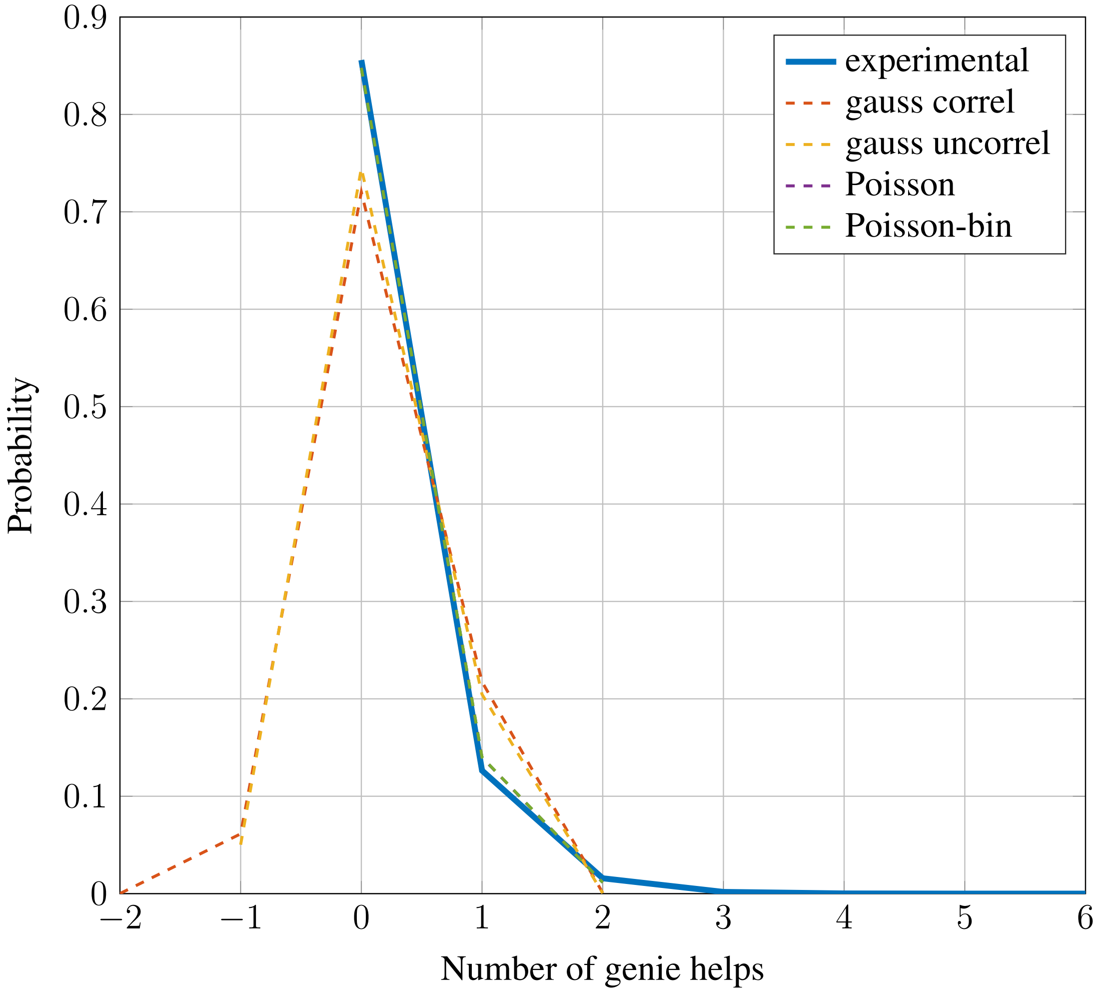


Figure 5.1: genie helps distribution for $R = 0.40$ and $n = 12$.

We see that the best approximation is given by the Poisson and Poisson-binomial distributions, as expected since they take only positive integer values.

In Figures 5.2 and 5.3 we see that the Poisson and Poisson-binomial approximations progressively worsen, while the normal approximation becomes more accurate. This is not unexpected, since with the Gaussian distribution we are able to adjust both the mean and the variance, whereas in the other two mean and variance depend one from the other, since there is only one parameter. Incidentally, we also observe that taking into account the correlation is crucial: the approximation given by the normal distribution assuming the random variables uncorrelated is far worse.

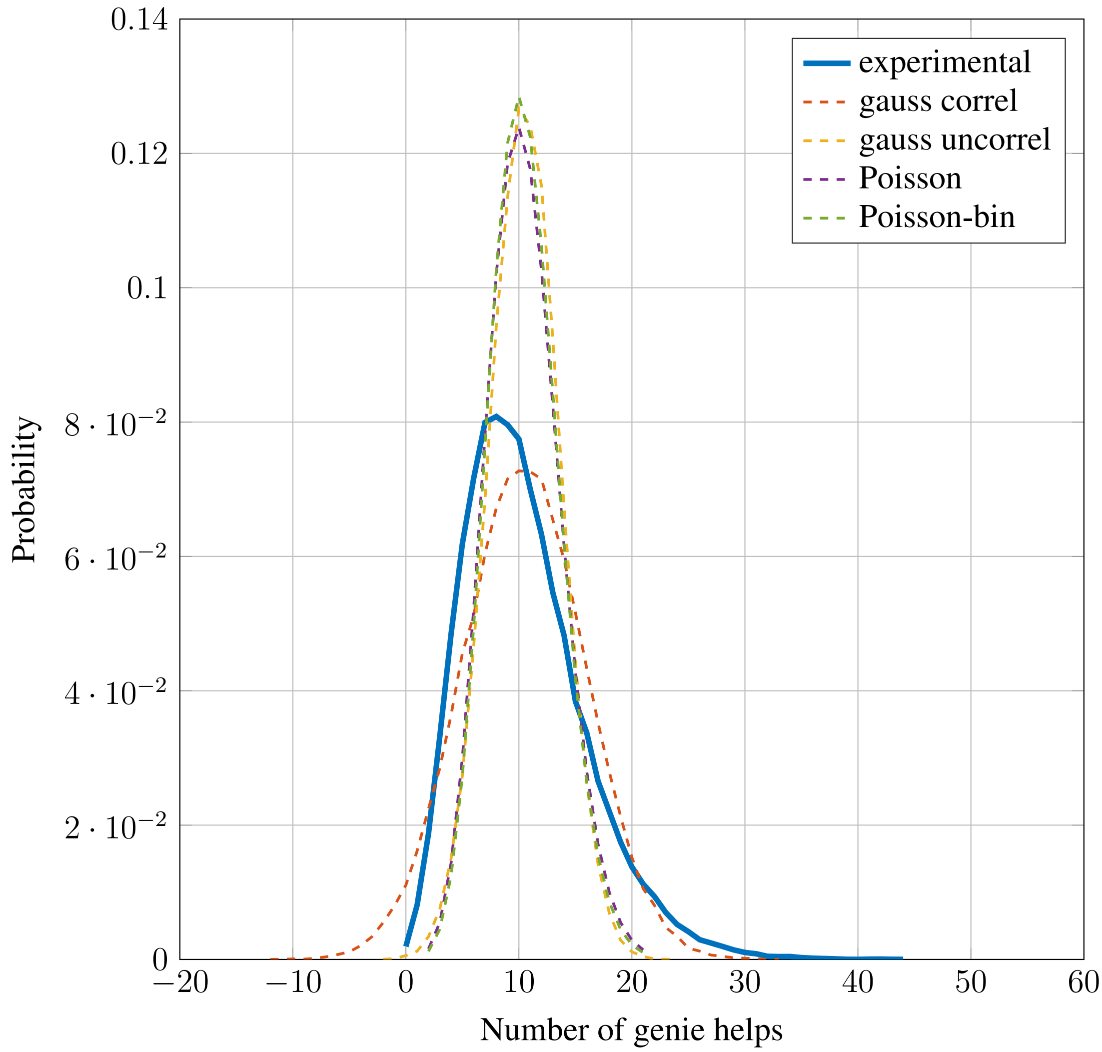


Figure 5.2: genie helps distribution for $R = 0.46$ and $n = 12$.

5.3 Models

From now on, in order to simplify the notation, it is more convenient to neglect frozen channels, and therefore index i will refer to nonfronzen channels only. Hence, for example, $Z^{(i)}$ is the Bhattacharyya parameter of the i-th non-frozen channel. The number of nonfrozen channels is denoted by K, which is some integer approximation of RN.

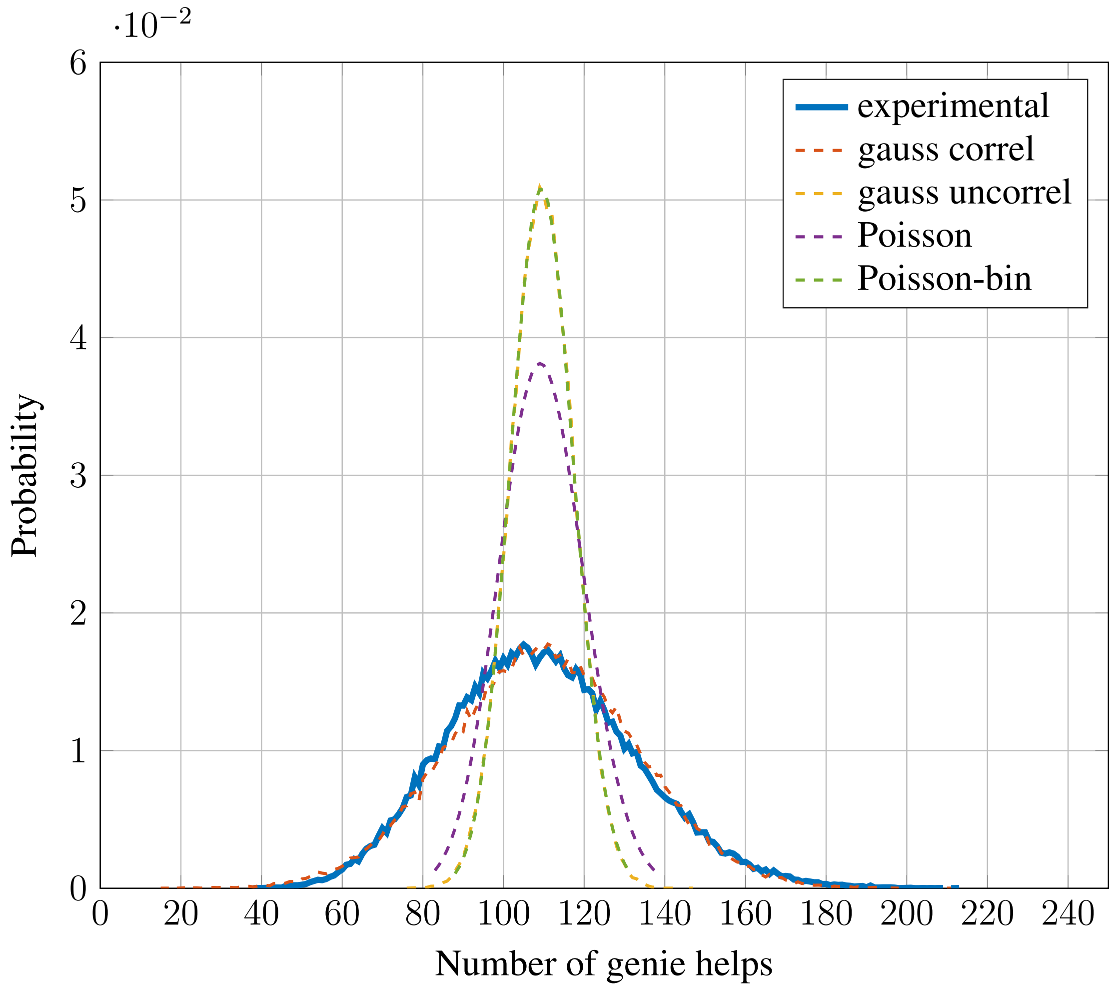


Figure 5.3: genie helps distribution for $R = 0.52$ and $n = 12$.

5.3.1 Scheme I

In this very simple scheme, the delay to transmit a packet is easily modeled via a geometric random variable whose parameter is the probability $1 - P_e$ of a packet to be correctly decoded without genie helps. Then, the delay is given by

$$D = \frac{P_e}{1 - P_e} \tag{5.11}$$

since we consider a delay 0 for a packet that is decoded immediately at reception (i.e., we consider delay as the number of additional packets we need to send in order to decode the packet of interest).

5.3.2 Scheme II

We already stated that for the BEC, $Z^{(i)}$ gives the probability of needing a genie help at (nonfrozen) bit i. Errors in SC, and therefore genie helps, are not independent. In fact, intuitively, having a genie help at bit i means that channel $W_N^{(i)}$ is not very good, and by construction it is more likely that another non-optimal channel will be in the nearby. However, in order to keep our models reasonably simple, we will consider them as independent.
Therefore, the events of having genie helps at given bits will be considered as independent Bernoulli random variable of parameters $Z^{(i)}$ (this means that they are independent, but not identically distributed).
We consider a packet p, and denote by X_0^p the number of genie helps of packet p. We also assume $X_0^p = i > 0$, otherwise we straightforwardly have zero delay.
Then, X_m^p, $\forall m > 0$ is the number of errors in packet $p + m$ in case at least one genie help is needed in one of its retransmitted bits, otherwise, if packet $p + m$ has no errors involving one of the restransmitted bits, we have $X_m^p = 0$.
X_m^p is a random variable $\forall m \geq 0$, and $p \geq 0$ fixed, and $\{X_n^p\}_{n\geq 0}$ is a Markov chain $\forall p \geq 0$ fixed, since the knowledge of the current state is clearly sufficient to stochastically describe all future states.
In order to derive a model for the delay, we specify the Markov chain as in Fig. 5.4. We denoted by $\psi_0^{i-1}(j)$ the probability of having j total errors in a packet, of which at least one involving one of the i retransmitted bits, and $\psi_0^{i-1}(0)$ is the probability of having no errors in the first i bits.
In order to derive an expression for $\psi_0^{i-1}(j)$, let us denote by $\varphi_s^i(k)$ the probability of having exactly k (independent) errors in channels $s, ..., i$ for a given packet.
$\varphi_s^i(k)$ can be computed as follows:

$$\varphi_s^i(k) = \sum_{\substack{(e_s,\dots,e_i) \\ \sum_{j=s}^i e_j = k}} \prod_{j=s}^{i} |e_j - (1 - Z^{(j)})| = Z^{(s)} \varphi_{s+1}^i(k) + (1 - Z^{(s)}) \varphi_{s+1}^i(k-1) \quad (5.12)$$

with $e_j \in \{0,1\}$ and boundary conditions:

$$\begin{cases} \varphi_i^k(k-i+1) = \prod_{j=i}^k Z^{(j)} \; \forall 0 \leq i \leq k \\ \varphi_i^k(0) = \prod_{j=i}^k (1-Z^{(j)}) \; \forall 0 \leq i \leq k \\ \varphi_i^k(l) = 0 \text{ for } l > k-i+1, \; \forall 0 \leq i \leq k \end{cases} \tag{5.13}$$

since independent errors assumption implies that bit i is correctly decoded with probability $1-Z^{(i)}$.

This implies that the number of errors X in bits $s,...,j$ is distributed as a *Poisson binomial distribution*, with parameters $Z^{(s)},...,Z^{(j)}$ which can be efficiently computed as [16]:

$$\mathbb{P}(X=k) = \frac{1}{j-s+2} \sum_{l=0}^{j-s+1} C^{-lk} \prod_{m=1}^{j-s+1} (1+(C^l-1)Z^{(m)}) \tag{5.14}$$

with

$$C = \exp\left(\frac{2i\pi}{j-s+2}\right) \tag{5.15}$$

and $i = \sqrt{-1}$.
Then,

$$\psi_0^{i-1}(j) = \begin{cases} \sum_{m=1}^{i} \varphi_0^{i-1}(m)\varphi_i^{K-1}(j-m) & \text{if } j \geq i \\ \sum_{m=1}^{j} \varphi_0^{i-1}(m)\varphi_i^{K-1}(j-m) & \text{if } j < i \end{cases} \tag{5.16}$$

that is,

$$\psi_0^{i-1}(j) = \sum_{m=1}^{\min(i,j)} \varphi_0^{i-1}(m)\varphi_i^{K-1}(j-m) \tag{5.17}$$

and

$$\psi_0^{i-1}(0) = \varphi_0^{i-1}(0) \tag{5.18}$$

Then, the probability transition matrix of the Markov chain is simply given by $P_{i,j} = \psi_0^{i-1}(j) \;\; 0 < i \leq K, j \geq 0$, and $P_{0,0} = 1$.
We define the random variable D_p representing the delay before decoding associated to packet p as follows:

$$D_p \triangleq \min\{n \geq 0 : X_n = 0\} \tag{5.19}$$

where state 0 is absorbing (i.e. D_p is the absorption time associated to Markov chain $\{X_n^p\}_{n\geq 0}$). In order to compute that, we fix an initial state $X_0^p = k$, which represents

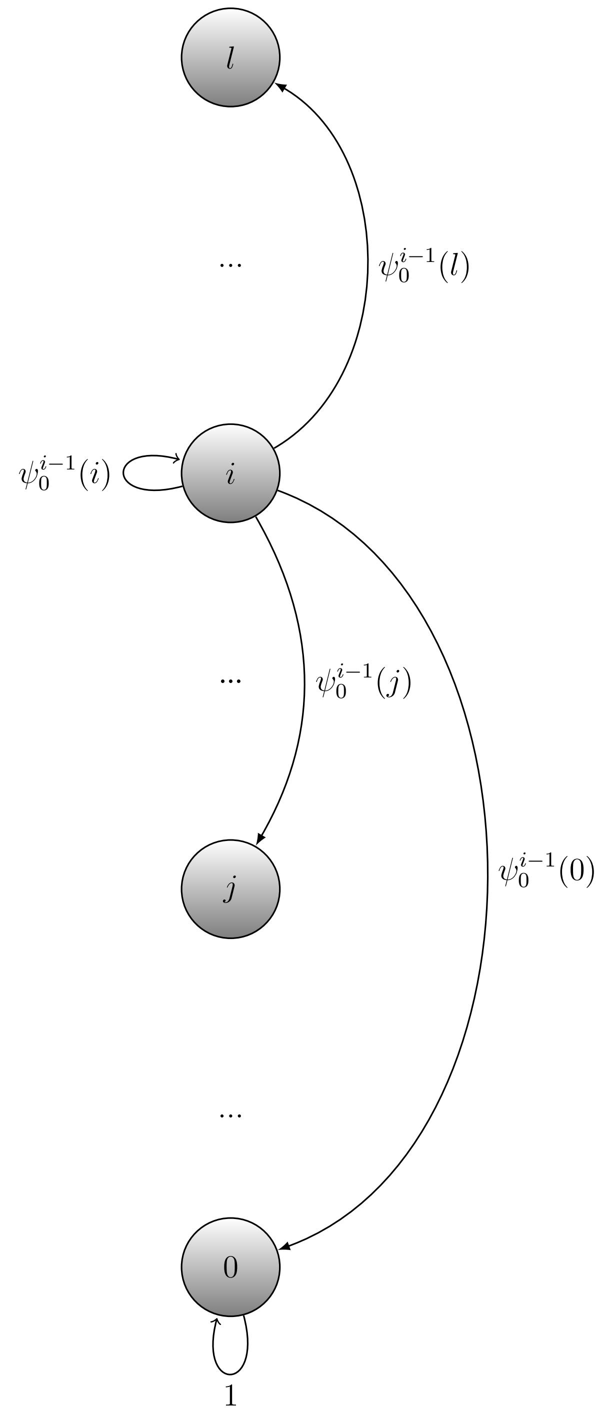


Figure 5.4: Markov chain for scheme II.

the number of errors in packet p upon reception. By solving the Markov chain we get $\mathbb{E}[D_p|X_0^p = k]\ \forall 0 \leq k \leq K$. In fact by denoting $\nu_k = \mathbb{E}[D_p|X_0^p = k]$ by first step analysis we have [17]

$$\underline{\nu} = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \cdots \\ \nu_K \end{bmatrix} = \underline{1} + \underline{\underline{\tilde{P}}} \cdot \underline{\nu} \tag{5.20}$$

with $\nu_0 = 0$ by definition, $\underline{1}$ the K-dimensional column vector of all 1's and $\underline{\underline{\tilde{P}}}$ is the transition matrix $\underline{\underline{P}}$ without the first column and row (i.e., the ones corresponding to absorbing state 0). This yields:

$$\underline{\nu} = \underline{1} \cdot (\underline{\underline{I}} - \underline{\underline{\tilde{P}}})^{-1} \tag{5.21}$$

Then, $\mathbb{E}[D_p] = \mathbb{E}[\mathbb{E}[D_p|X_0^i = k]] = \sum_{k=0}^{K} \mathbb{E}[D_p|X_0^i = k]p(k)$ where $p(k)$ is the pdf of the number of errors in the packet (in the non-frozen channels). As proven in subsection 3.2.2, we have $\mathbb{E}[D] = \mathbb{E}[D_p]$. We finally remark that the inversion of $\underline{\underline{I}} - \underline{\underline{\tilde{P}}}$ is $O(K^3) = O(N^3)$ since R is constant, that is, it is an operation computationally expensive.

5.3.3 Scheme III

Given a packet, in order to decode it we only need to know, for each of the following packets, how many errors happen in the region of intersection of retransmitted bits. We call this region *decoding region for packet 0 at packet n*, and we denote its size by R_0^n. By design it is always at the beginning of each packet and it contains the bits of the first packet that are still missing. Without loss of generality (since the system is time-invariant and therefore the underlying Markov chain in homogeneous, and metrics do not change as stated in subsection 3.2.2) we denote by 0 the packet of which we want to know the delay before decoding, and by X_0 the number of errors that occur in this packet. Hence, if in packet 0 i errors occur, we have $R_0^1 = i$ since the first i bits of packet 1 will contain the retransmission of the i erroneous bits of packet 0. For packet 2, R_0^2 will be the number of errors of packet 1 that occur in its first i bits, and so on.

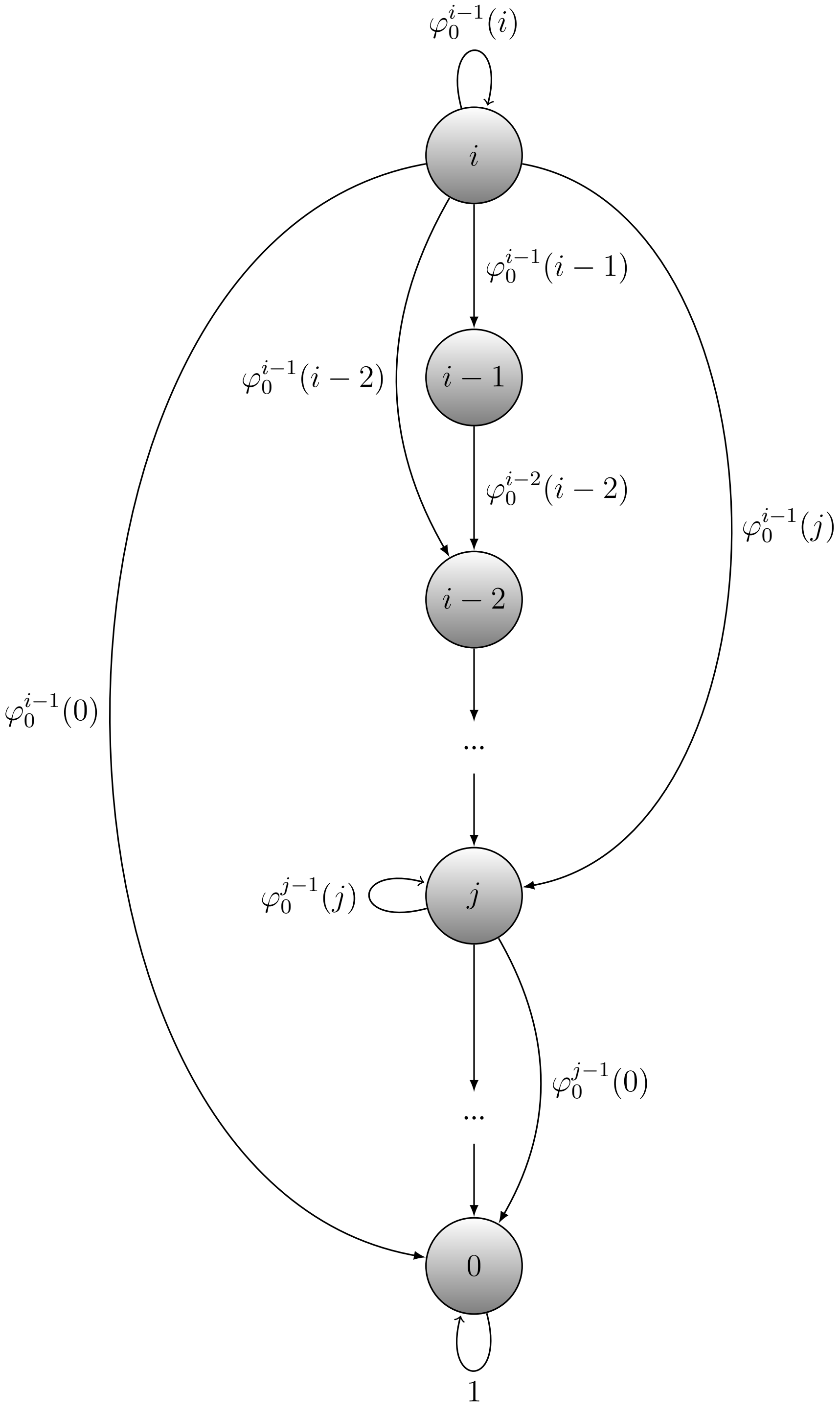


Figure 5.5: Markov chain for scheme III.

Then, $X_n = R_0^n$, $n \geq 1$, will denote the number of bits, which are at the beginning, that must be decoded in packet n in order to be able to completely decode packet 0. This implies that X_n is also equal to the number of errors in packet 0 that still have to be corrected. Note that since R_0^n represents an intersection between errors and a number of bits, it is a non-increasing function in n. The Markov chain we obtain is shown in Fig.5.5, with i the number of errors in packet 0 and $\varphi_0^i(j)$ is the probability of having exactly j (independent) errors in channels 0,...,i for a given packet.
In fact, if at some point the decoding region has size R_0^n and $j \leq R_0^n$ errors occur in the first R_0^n bits of packet n, then for the following packet we will have $R_0^{n+1} = j$. Packet 0 is decoded as soon as $R_0^m = 0$ for some m. For the delay, the same considerations of the previous section hold.
For the Markov chain we obtain that $P_{i,j} = \varphi_0^{i-1}(j)$, which is the probability of having j errors in the decoding region of packet n given that its size is i.
Again, the solution of the Markov chain gives $\mathbb{E}[D_i|X_0^i = k]$, average delay for a given number of initial errors (initial state of the chain). Then, $\mathbb{E}[D] = \mathbb{E}[\mathbb{E}[D|X_0^i = k]] = \sum_{k=0}^{K} \mathbb{E}[D|X_0^i = k]p(k)$ where $p(k)$ is the pdf of the number of errors in the packet (in the non-frozen channels). If we assume the errors to be independent, then they have a Poisson-binomial distribution, and we can take advantage of the quantities already computed:

$$\mathbb{E}[D^{\perp\!\!\!\perp}] = \sum_{k=0}^{K-1} \mathbb{E}[D|X_0^i = k]\varphi_0^{K-1}(k) \tag{5.22}$$

Accuracy of the Theoretical Model for Scheme III

In Figures 5.6 and 5.7, as a preliminary to the analysis of our theoretical model, we evaluate the impact of our crucial assumption of considering the genie helps as independent, while in reality they are correlated as we verified in section 5.2. The simulation with independent genie helps is simply obtained by realizations of Bernoulli random variables of parameters the Bhattacharyya parameters of the corresponding bits.
We see that the approximation is very good: the two curves are very close to each other. Therefore this assumption is experimentally validated.
Fig. 5.8 shows the approximation given by solving the Markov chain. In order to evaluate the accuracy of the Markov model only (whose solution gives the expected delay

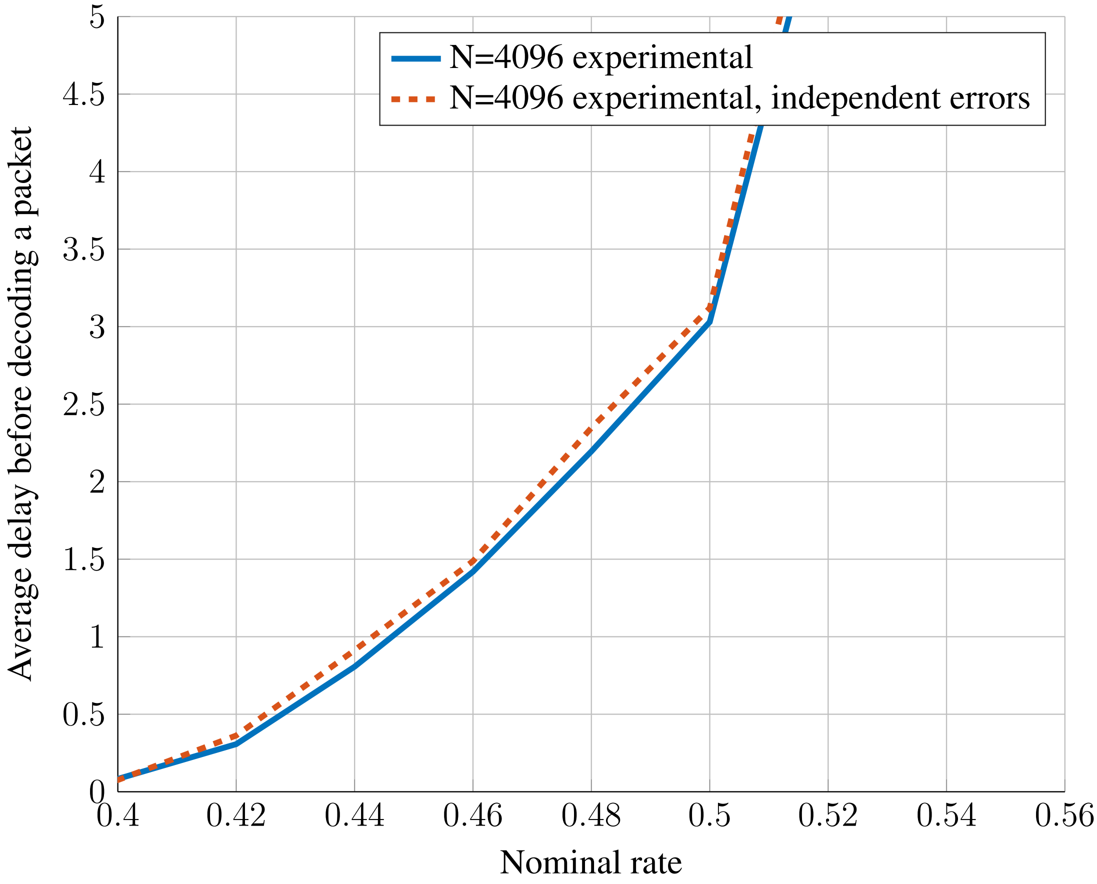


Figure 5.6: comparison between correlated and independent genie helps.

for a given initial number i of errors), we derived the expected delay using as the distribution of the initial errors an experimental one, instead of applying one of the models of section 5.2.

We see that the Markov model results to be very accurate.

In order to estimate the impact of the initial distribution (true correlated or approximated independent), we refer to Fig. 5.9. We see that the approximation is very good, despite the fact that, as we saw is section 5.2, the Poisson-binomial distribution is not *per se* a very good approximation of the true genie distribution. Evidently, the Markov chain has a mitigating effect on the initial distribution, that is, the expected delay varies slowly when varying the number of initial errors.

However, the computation of the probability transition matrix is very demanding (even

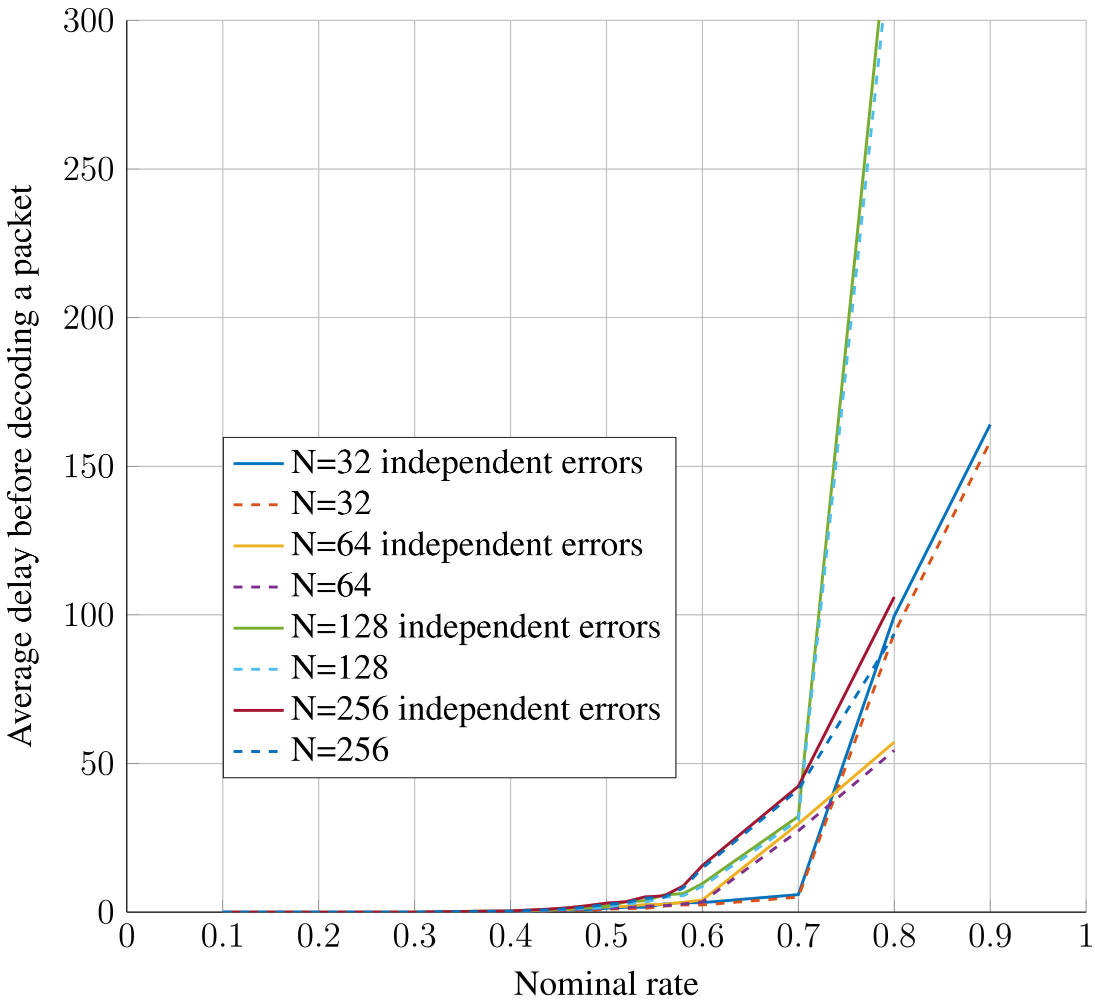


Figure 5.7: comparison between correlated independent genie helps for scheme III and various N.

though it gives, unlike a simulation, exact results), but for scheme III, unlike scheme II, the matrix is lower triangular, and therefore its inversion has complexity $O(N^2)$ instead of the usual $O(N^3)$.
We remark that, in both correlated and independent case, we can compute $\varphi_0^{2^n-1}(k)$ (in case of rate $R = 1$) as

$$\varphi_0^{2^n-1}(k) = \sum_{l=\max\{0,k-2^{n-1}\}}^{\min\{k,2^{n-1}\}} \varphi_0^{2^{n-1}-1}(l)\varphi_0^{2^{n-1}-1}(k-l) \tag{5.23}$$

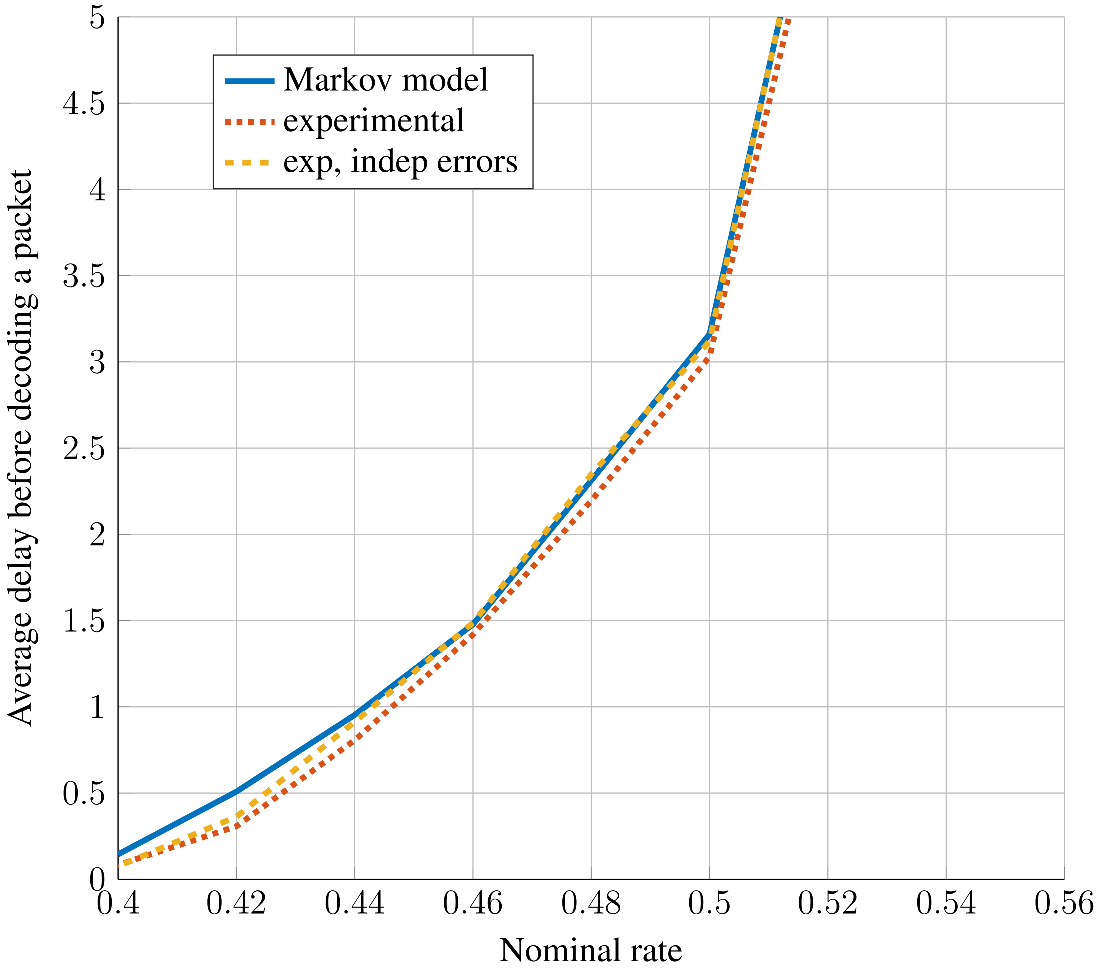


Figure 5.8: comparison between MC and simulated scheme with $N = 4096$.

The idea is to combine the genie aid coming from each of the two sub-blocks of size 2^{n-1} (which are independent) of the construction of the polar code.
By induction, and using the Vandermonde convolution, we have that for a $BEC(\varepsilon)$

$$\varphi_0^{2^n-1}(k) = \binom{2^n}{k} \varepsilon^k (1-\varepsilon)^{2^n-k} \tag{5.24}$$

that is, the number of errors on the whole packet with rate $R = 1$ is distributed as a binomial random variable of parameter ε, which is also the average of all the 2^n Bhattacharyya parameters. This suggests us that a possible approximation of the number of errors in a given section, for any rate, may be a binomial with parameter the average

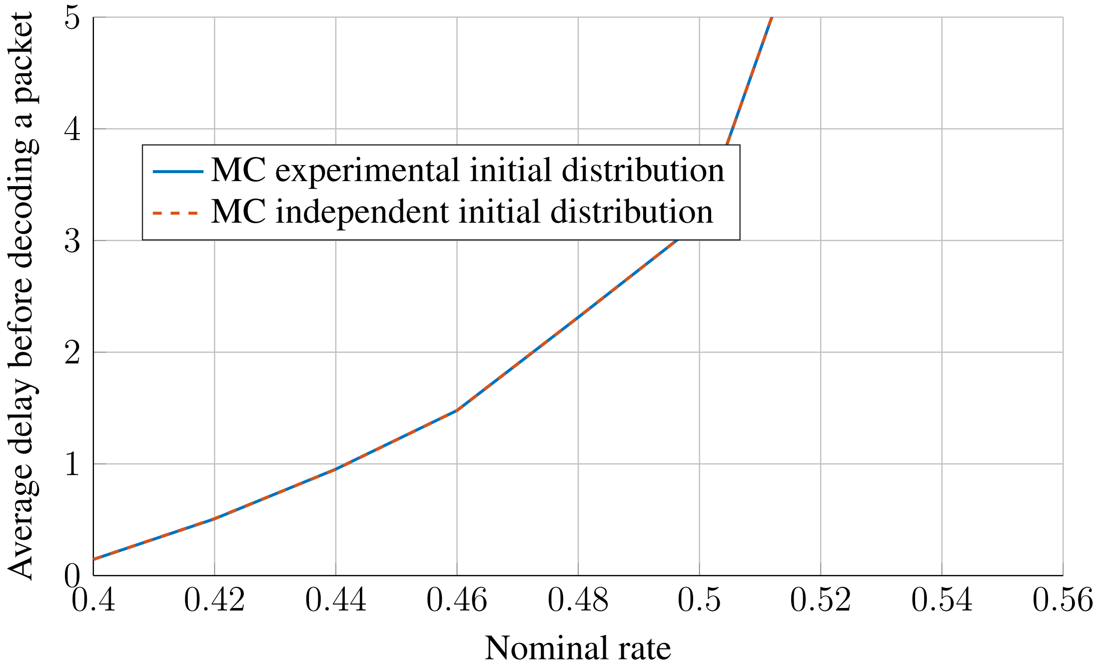


Figure 5.9: impact of the usage of independent or experimental genie distributions with $N = 4096$.

of the Bhattacharyya parameters in the given section, i.e.

$$\varphi_0^{i-1}(k) = \binom{i}{k} \hat{Z}^k (1 - \hat{Z})^{i-k} \tag{5.25}$$

with

$$\hat{Z} = \frac{1}{i} \sum_{j=0}^{i-1} Z^{(j)} \tag{5.26}$$

which experimentally gives a pretty accurate approximation of the delay (see 8.1).

5.4 Effective Rate

The estimation of the effective rate is straightforward and exact. In fact, by applying eq. 5.5 to eq. 3.1 with $M = \mathbb{E}[S]$ we get

$$R_{eff} = R - \frac{\sum_{j=0}^{K-1} Z^{(j)}}{N} \tag{5.27}$$

Chapter 6

List Decoding

6.1 Introduction

List decoding [18] [19] [20] is a technique introduced to improve the finite-length performance of polar codes. It is mainly applied to BEC, but it can also be extended to other types of channels. The key idea is to exploit the information given by frozen bits to reduce the number of genie aids in successive cancellation.

List decoding is basically and extension of successive cancellation that consists in splitting the decision process (thus creating a sort of tree, or list of decision patterns) each time we are unable to decide based on the LLR (i.e., whenever we have at bit i $L_n^{(i)} = 0$ for the BEC). Now, it may happen that while going on along a given decision pattern, at some point we decode for a frozen bit, and the decision we would take based on the LLR is in conflict with the value of the frozen bit, that is known. In that case, since it is impossible to have a conflict with a frozen bit, that branch is surely wrong and can be deleted.

Therefore, the use of list decoding instead of SC reduces the total number of genie helps. A trade-off between this reduction and decoding complexity emerges: by increasing the list size we reduce the number of genie helps, but on the other hand we increase the decoding complexity by a factor equal to the size of the list.

If the size of the list is finite, then when reaching it, we call for genie help. In this case it is clearly better to call it at the position of the first split (that is, the root of the decision

tree), in order to free the biggest number of lists (all which had as guess the opposite of the genie help).
Our simulations clearly use the first system associated to scheme III, which is the one with the best performance. Moreover, additional genie helps are used to resolve (if needed) the decision tree that remains when all bits have been decoded.
We finally state an important property of list decoding, without proving it: the space of all decision sequences (that is, the set of vectors we had if we followed all the paths of the decision tree, from root to leaves) is a vector subspace of $\mathbb{GF}(2)^N$ (or more precisely, an affine space given by a vector space translated by the true codeword. However, since we consider the all-0's codeword, it is effectively a vector space).
This has the remarkable consequence that the bits at which the splitting happen are the same for all the lists, or in other words if a branch splits at bit i, then all branches split at bit i. This implies that the decision tree is a complete tree whose nodes have all the same height (therefore it is sensible to choose as the list size L a power of 2).

6.2 Experimental Results

In Fig.6.1 we show the delay performance of scheme III, whereas in Fig. 6.2 we give a detail of the region of interest. We see that the performance improvement is appreciable for $L = 32$, but it quickly becomes completely negligible as we increase L. Moreover, it gets smaller as we increase N. Therefore we conclude that for high N the performance improvement does not justify the additional decoding complexity required.
The other metric of interest, R_{eff}, shows the same behavior, as we see in Fig.6.3, where for clarity we omitted the legend (which is the same of Fig. 6.1).

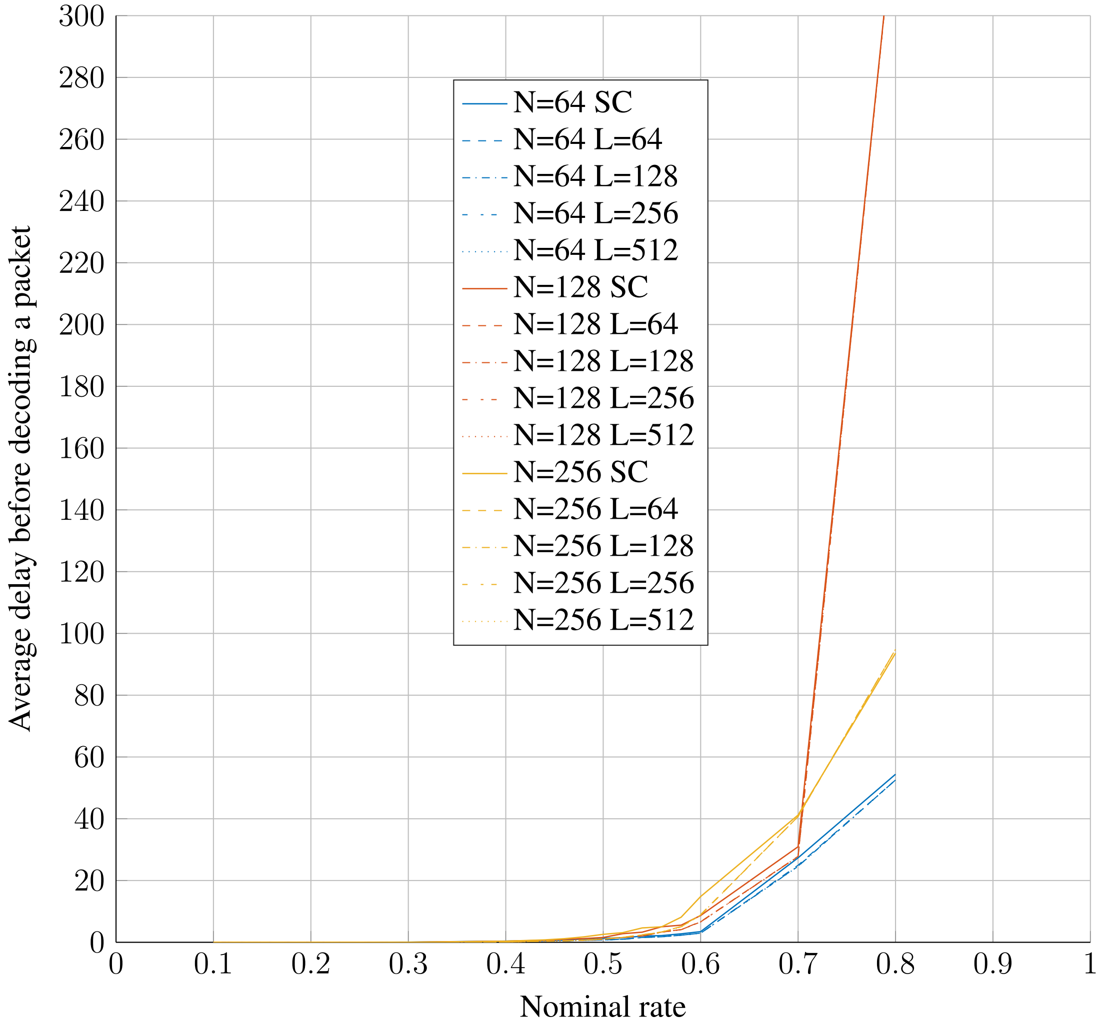


Figure 6.1: delay with list decoding for scheme III.

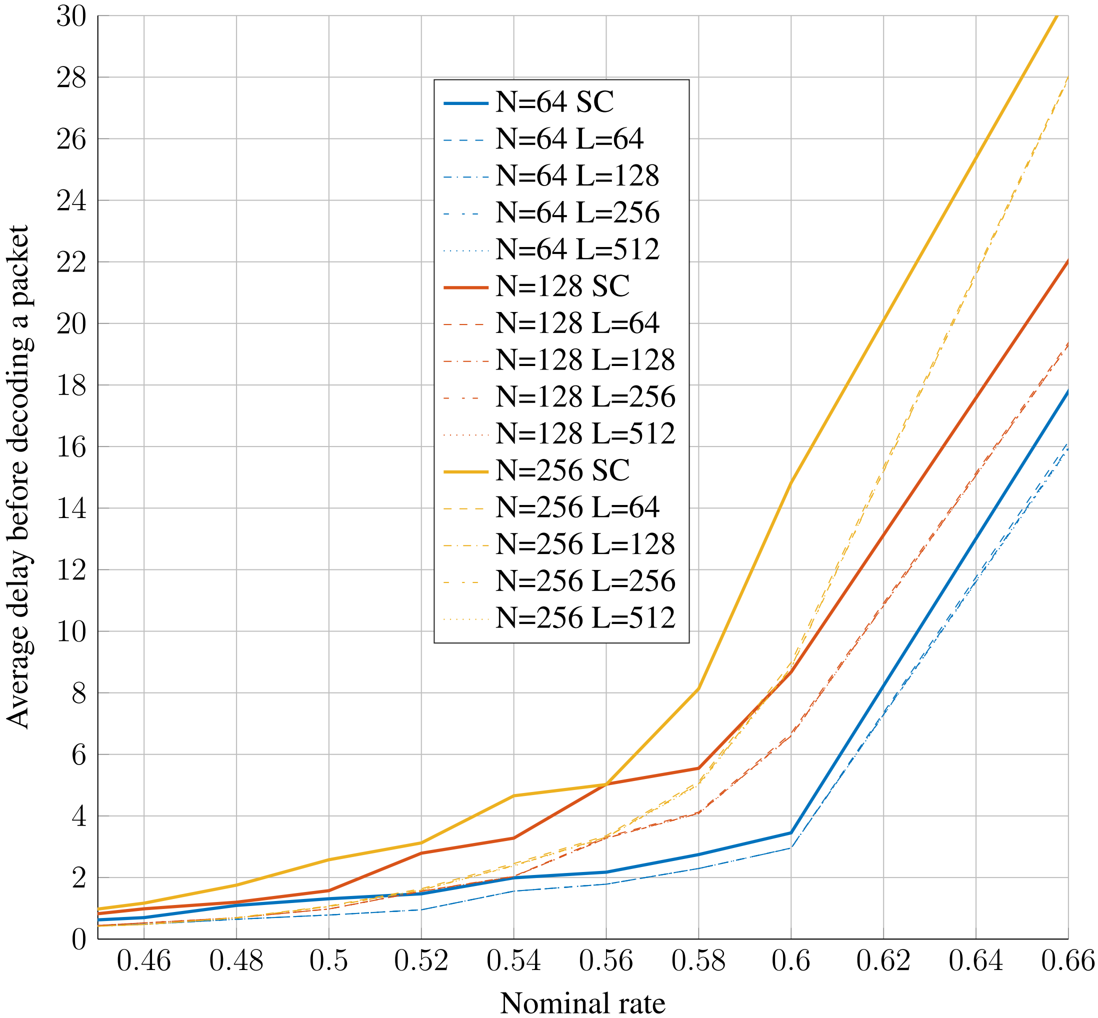


Figure 6.2: detail of delay with list decoding for scheme III.

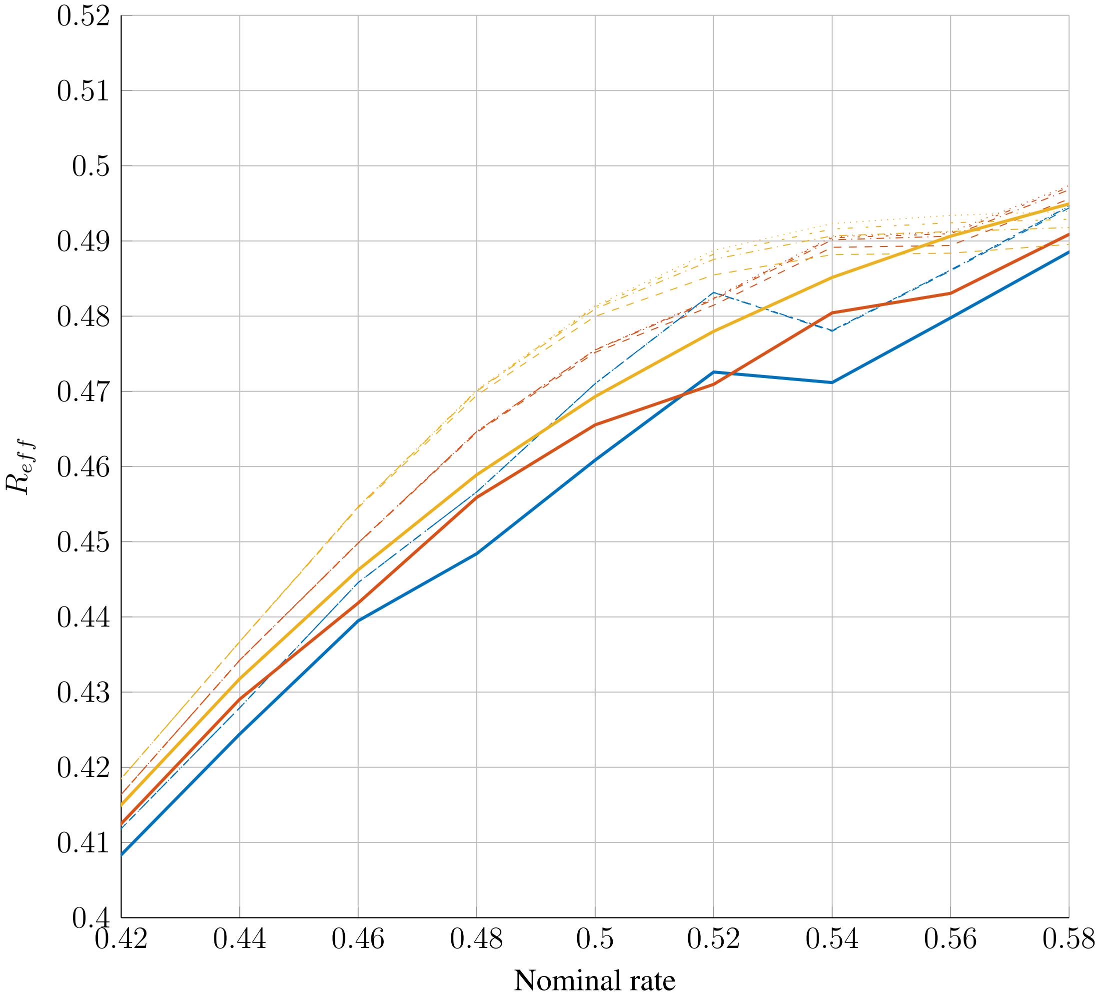


Figure 6.3: detail of R_{eff} with list decoding for scheme III.

Chapter 7

Gaussian Channel

7.1 Introduction

We now apply scheme III for a Gaussian channel. We keep the hypothesis of full feedback. However now, unlike the BEC case, the receiver has no knowledge of whether it is making a mistake or not, since $L_n^{(i)}$ now can take all real values. In fact, while in the BEC case we knew exactly at which bits the errors would have occurred, that is those with $L_n^{(i)} = 0$, and therefore we could invoke the strictly necessary number of genie helps to ensure $P_e = 0$, now the LLR $L_n^{(i)}$ gives only a probability for a bit to be 0 or 1. Still, using the LLR we have a clue about the bits at which the SC decoder is most likely to commit an error. In fact the LLR gives not only the decision criterion of eq. 8.7, but also the confidence in taking that decision. The bigger the absolute value of the LLR, the greater is the confidence in taking the decision.

Therefore a reasonable thing to do is to consider that receiver and sender agree on a threshold, and if a LLR is below that threshold in absolute value, then the receiver asks for a genie help for that bit. The reason behind this system is that in such a case there is too little confidence to make a valid choice for that bit. The scheme then becomes Algorithm 9.

This system has two important consequences. Firstly, now we also have an outage probability of decoding failure (or residual error probability) even in case of genie-helped decoding, that is, the probability that a bit is incorrectly decoded even if the correspond-

Algorithm 9 Genie-aided SC decoding for the Gaussian channel

for all i $\in \{0, ..., N-1\}$ **do**
  compute $L_n^{(i)}$
  if i is frozen **then**
    $\hat{u}_i \leftarrow u_i$
  else
    if $L_n^{(i)} > \tau$ **then**
      $\hat{u}_i \leftarrow 0$
    else if $L_n^{(i)} < -\tau$ **then**
      $\hat{u}_i \leftarrow 1$
    else
      $\hat{u}_i \leftarrow g_i = u_i$
return $\hat{u}_0^{N-1}$

ing LLR is above the threshold.

Secondly, a trade-off between the outage probability of failure and the effective rate emerges. In fact, the residual error probability probability decreases as we increase the threshold, but on the other hand by increasing the threshold we also increase the number of genie aids, and therefore we reduce the effective rate since more bits are retransmitted.

7.1.1 Density Evolution

Given that there is no failure, the scheme and the model are exactly the same of the BEC case, where the only difference is that instead of $Z^{(i)}$ we have the probability that the log-likelihood ratio for the bit of index i falls inside the threshold.

For a BAWGNC(σ_w) (see appendix A.2) the log-likelihood ratio of received bit i is (using the notation introduced in eq. 2.43 and mapping $0 \leftrightarrow 1$ and $1 \leftrightarrow -1$):

$$L_0^{(i)} = \frac{2}{\sigma_w^2} y_i \tag{7.1}$$

Therefore, since Y_i is a random variable (of which y_i is a realization), $L_0^{(i)}$ is also a random variable and has a certain distribution.
This distribution is called the *L-density* of $L_0^{(i)}$, and is computed as the density of the LLR given the transmitted bit or symbol (usually it is assumed that bit 0 is transmitted).
For example, in the case of a BEC, we have

$$L_0^{(i)} = \begin{cases} 0 & \text{w.p. } \varepsilon \\ +\infty & \text{w.p. } (1-\varepsilon) \end{cases} \tag{7.2}$$

For the BAWGNC(σ_w), if we assume symbol $+1$ was transmitted, then

$$y_i = 1 + w_i \tag{7.3}$$

which implies

$$L_0^{(i)} = \frac{2}{\sigma_w^2} + \frac{2}{\sigma_w^2} w_i \tag{7.4}$$

and since $w_i \sim \mathcal{N}(0, \sigma_w^2)$ we have

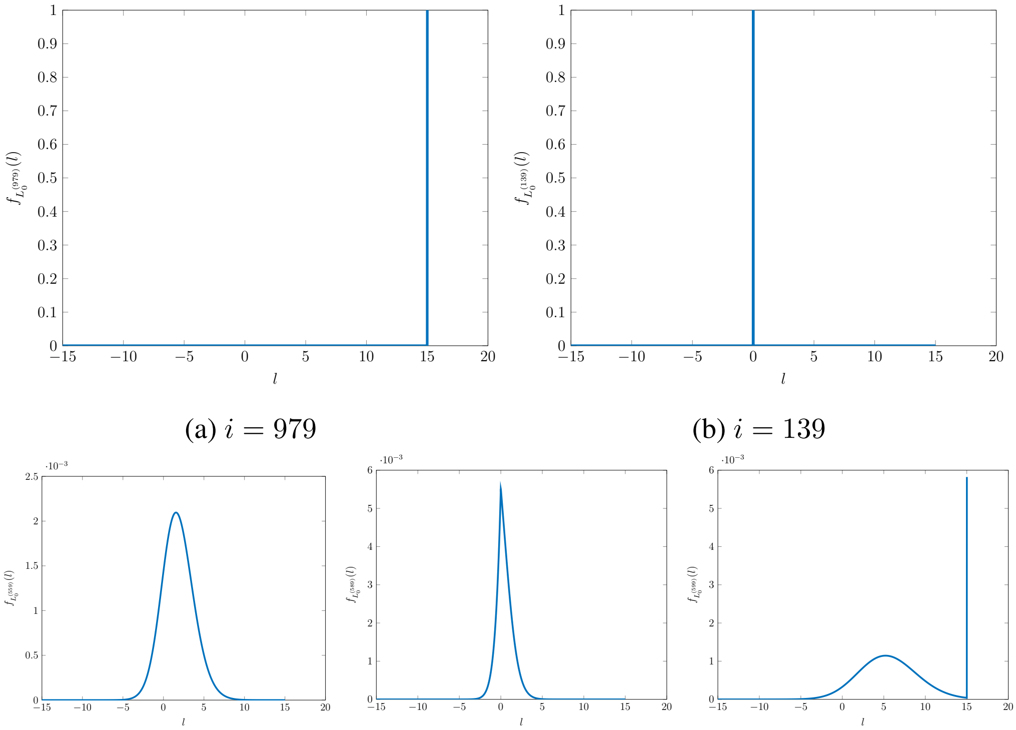
$$L_0^{(i)} \sim \mathcal{N}\left(\frac{2}{\sigma_w^2}, \frac{4}{\sigma_w^2}\right) \tag{7.5}$$

that is

$$f_{L_0^{(i)}}(l) = \sqrt{\frac{\sigma_w^2}{8\pi}} e^{-(y-\frac{2}{\sigma_w^2})\frac{\sigma_w^2}{8}} \tag{7.6}$$

Therefore, now we must know the distribution probability of the LLRs for every synthetic channels (i.e., the L-density).
The L-densities of the synthetic channels are given by the usual convolution $\circledast$ of the L-densities for the $+$ combination and by the convolution in the G-domain $\boxast$ for the $-$ combination, as stated in [21]. We see in Fig. 7.1 that, as expected, L-densities polarize: good and bad channels's densities become deltas centered in 0 (bad channels) and $+\infty$ (good channels, with all-0s transmitted codeword). Obviously, for numerical reasons, the $+\infty$ has been approximated with a large value, such that the probability of wrong decoding for such a value of LLR is practically 0. For our purposes, 15 is a suitable value to represent $+\infty$.
Then, to carry out the theoretical analysis, we can apply the model of section 5.3.3 with

$$Z^{(i)} = \int_{-\tau}^{\tau} f_{L_n^{(i)}}(u)du \tag{7.7}$$



(a) $i = 979$ (b) $i = 139$

(c) $i = 559$ (d) $i = 589$ (e) $i = 599$

Figure 7.1: L-density polarization for $n = 10$.

where τ is the LLR threshold within which the decoder asks for the genie help. However an important modification is needed, which derives from the fact that we have a residual failure probability: we need to introduce another absorbing state, corresponding to decoding error. Then the model we obtain is shown if Fig. 7.2. Intuitively, $\varphi_0^{i-1}(0)$ is the probability that for all bits in $0, ..., i-1$ the LLR realization falls beyond the threshold, but on the "right" side (that is, the side corresponding to the right decision). Conversely, $\Phi(i)$, given by 7.8, is the probability that at least in one bit of $0, ..., i-1$ the LLR realization is on the "wrong" side, which results in the decoder making a "confident" wrong decision.

$$\Phi(i) = 1 - \sum_{k=0}^{i} \varphi_0^{i-1}(k) \tag{7.8}$$

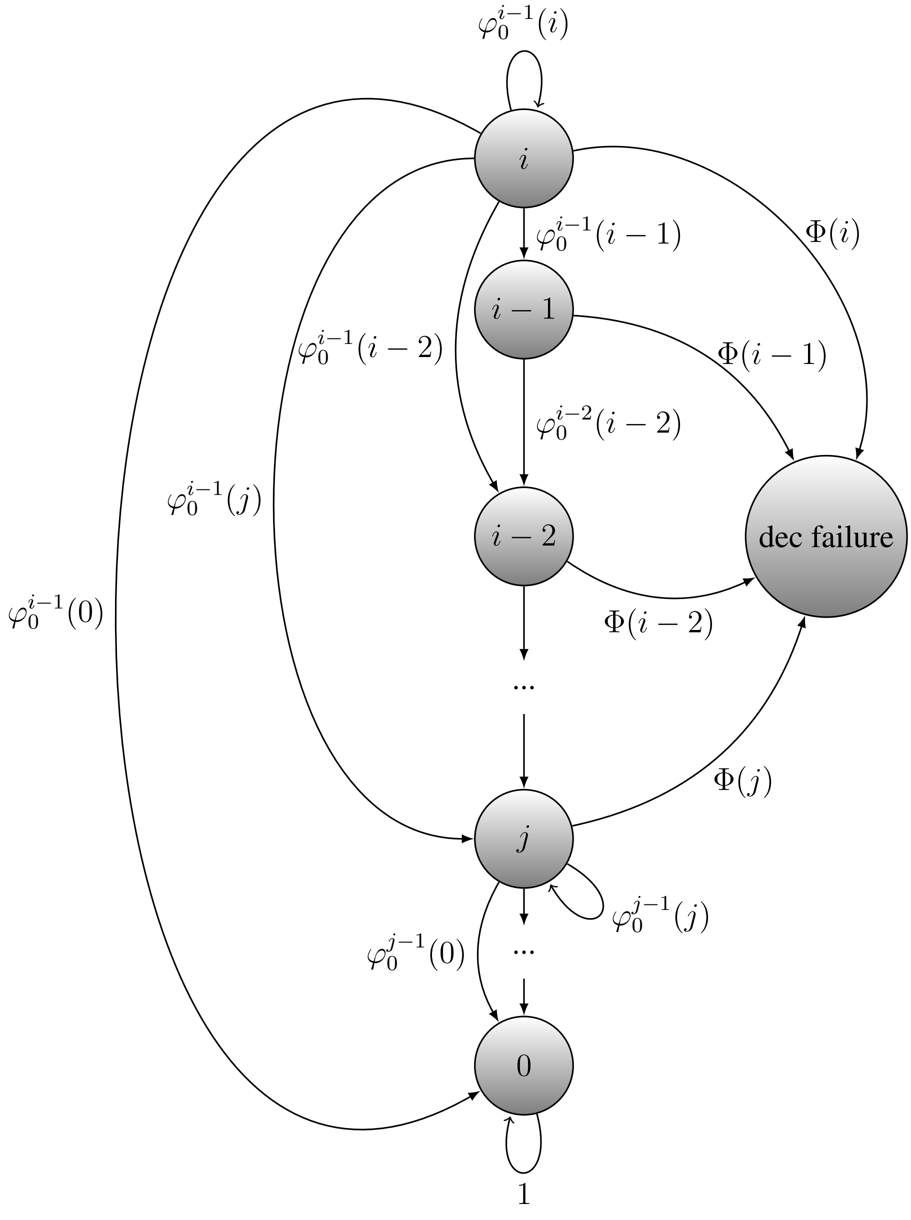


Figure 7.2: theoretical model for scheme III over BAWGNC.

7.1.2 Channel Estimation

Another important difference of the BAWGNC with respect to the BEC is that now the parameters of the synthetic channels, namely the Bhattacharyya parameters, cannot be computed exactly, but they have to be estimated.
For this purpose we use the following result:

$$Z(W_n^{(i)}) = \mathbb{E}\left[\sqrt{\frac{W_n^{(i)}(y_0^{N-1}, u_0^{i-1}|u_i \oplus 1)}{W_n^{(i)}(y_0^{N-1}, u_0^{i-1}|u_i)}}\right] \tag{7.9}$$

If we take into account the fact that we transmit the all-0s codeword, and compare the term inside the square root in eq. 7.9 with eq. 2.38, we see that they are the same except for the subsitution $\hat{u} \to u$. Hence, in 7.9 we use the likelihood ratios given that all previous bits are correct. This is easily obtained by setting all bits as frozen and by applying Algorithm 3, since, as stated in it, first we compute $L_n^{(i)}$ and then we evaluate if the bit i is frozen or not.
Then, the estimation of the expectation is carried out via Monte Carlo simulation of M packets (with M big enough to provide the desired accuracy on the estimation) of the SC decoder with all bits frozen, and then

$$Z(W_n^{(i)}) \approx \widehat{Z(W_n^{(i)})} = \frac{1}{M}\sum_{j=1}^{M} e^{-\frac{L_{n,j}^{(i)}}{2}} \tag{7.10}$$

where we used the LLRs instead of the LRs and $L_{n,j}^{(i)}$ is the LLR of the i-th bit of the j-th packet with all bits forzen.
Then, the frozen bits in the encoder are chosen according to the rate and the values of $\widehat{Z(W_n^{(i)})}$.
Clearly, since we have an estimation instead of exact values, we get an additional source of errors. This is due to the fact that, as a result of the estimation, we may not take all and only the best channels, but we may have a frozen channel that is better than an unfrozen one. This is even more important if we use the estimation of a channel for another channel (e.g., we estimate the Bhattacharyya parameters, and therefore the frozen channels, for a BAWGNC(σ_w) and we keep the same for a BAWGNC(σ'_w) with $\sigma_w \neq \sigma'_w$).
In fact, in general, if we consider a family of channels indexed by a parameter (e.g.,

$\{\text{BAWGNC}(\sigma_w),\ \sigma_w \in \mathbb{R}^+\ \}$ or $\{\text{BEC}(\epsilon),\ \epsilon \in [0,1]\ \}$), then the order (according to the Bhattacharyya parameters) of the channels is not stable with respect to the parameter of the channel. For example if we take two channels BAWGNC(σ_w) and BAWGNC(σ'_w), then

$$Z(W_n^{(i)}(\sigma_w)) < Z(W_n^{(j)}(\sigma_w)) \not\Rightarrow Z(W_n^{(i)}(\sigma'_w)) < Z(W_n^{(j)}(\sigma'_w)) \tag{7.11}$$

Therefore, in order to get the best performance, the estimation procedure should be repeated for every channel (i.e., for every value of the parameter of the channel).
However, in practice, what is usually done is to consider an interval of values of the parameter, take a value representative of that interval, estimate the channel only for that value and keep the estimation for all the values of the interval. This is because, empirically, it is verified that the order of the Bhattacharyya parameters does not change abruptly for small variations of the parameter, and therefore, if the interval is small enough, the error we commit on the channel choice is not big (i.e., we do not take very bad channels instead of very good ones).
Fig.7.3 shows the impact of channel estimation on P_e.

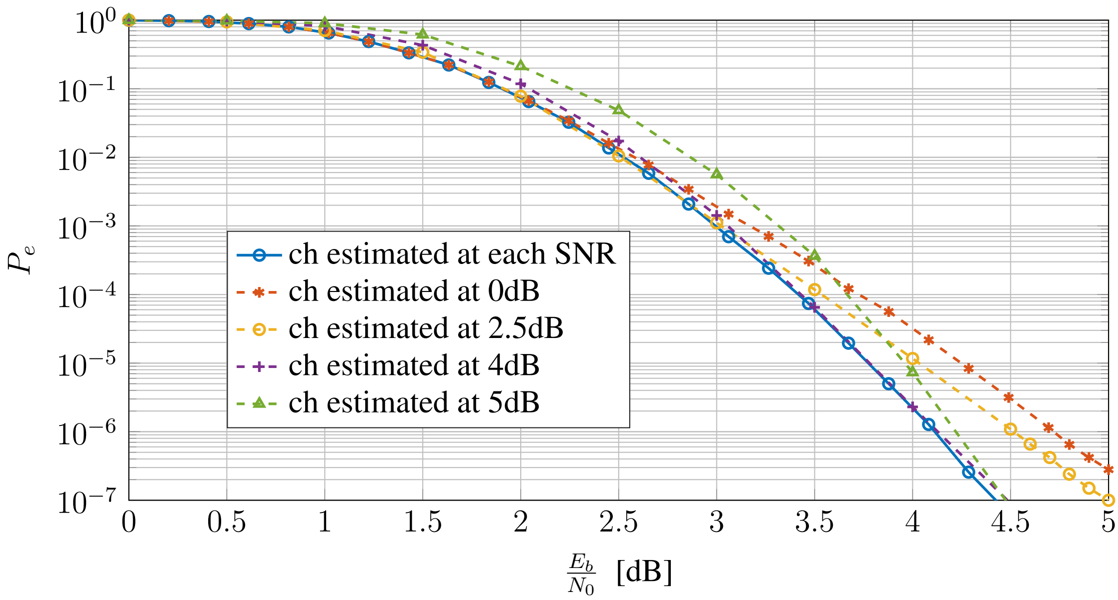


Figure 7.3: impact of channel estimation with $N = 1024$.

7.1.3 Experimental Results

We estimate the channel using $M = 10^7$. We fix the noise variance $\sigma_w^2 = 1$ so that we do not have to estimate the channel multiple times, and we vary the nominal rate R. Fig. 7.4 shows the simulated results. A first remark that can be made is that the behavior of the delay seems qualitatively the same for all the thresholds. We can suppose that the "staircase" behavior is due to the effect of thresholds: genie invocations are grouped according to the thresholds, rather than being selected individually.

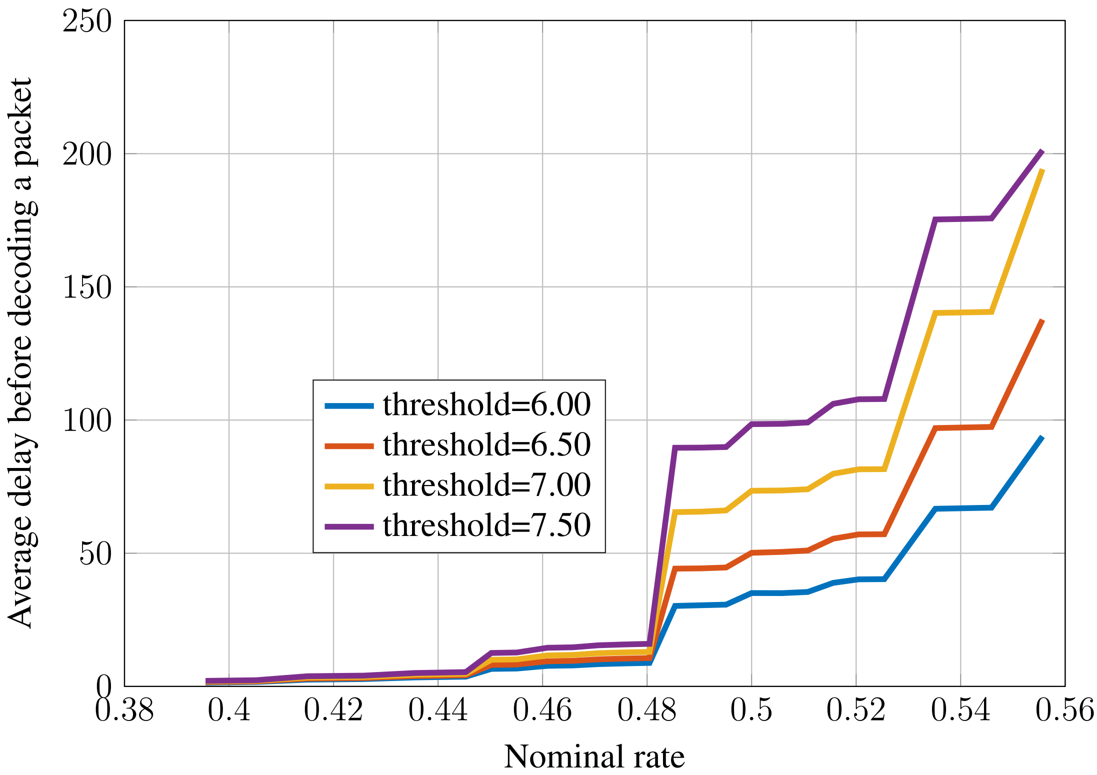


Figure 7.4: delay given by simulation of scheme III with $N = 1024$.

For computational reasons we do not solve the Markov chain for all rates, but we focus on the region around $R = 0.5$, where an abrupt increase of the delay occurs.

The comparison between the Markov model and the simulations for the BAWGN channel is shown in Fig. 7.5. Again, the two curves have exactly the same qualitatively behavior. From a quantitative point of view, the simulation has an underestimation bias. This is probably due to the limit imposed on the maximum delay. In fact Fig.7.6 shows

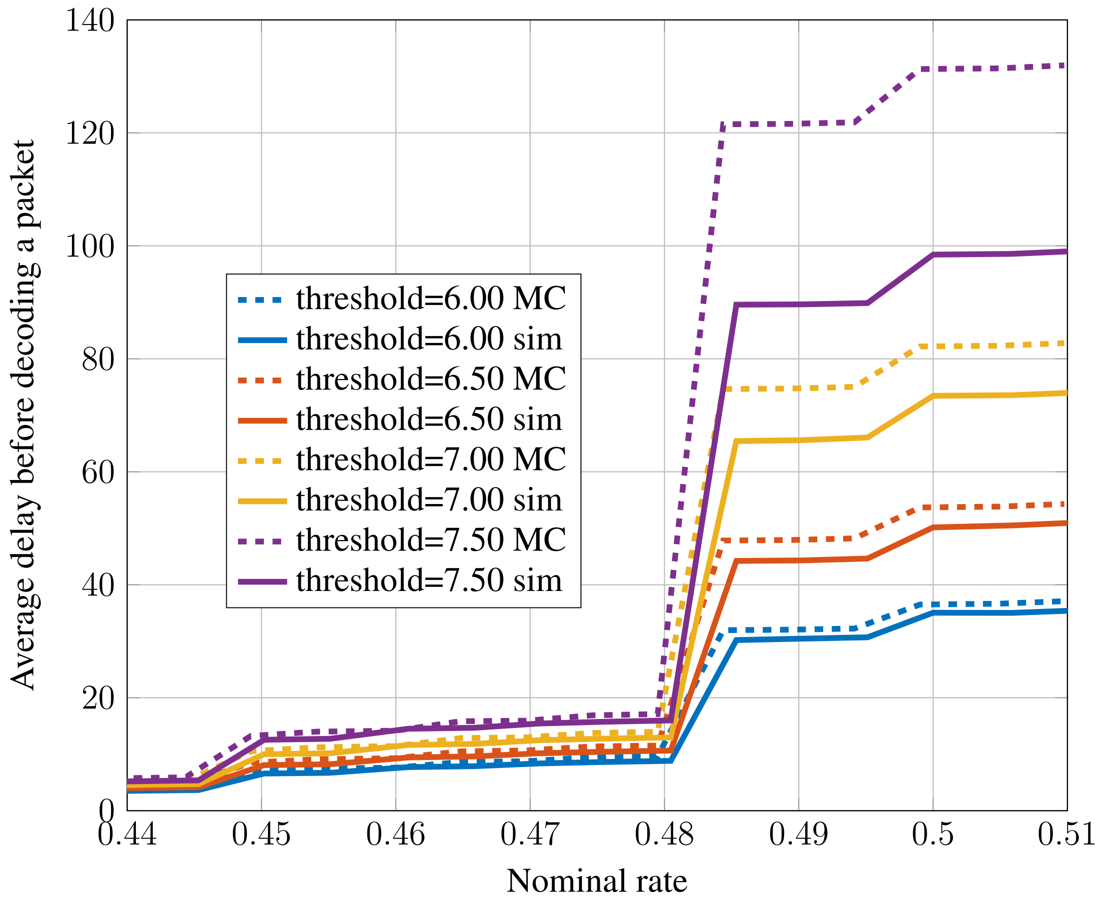


Figure 7.5: comparison of simulation and Markov model for scheme III with $N = 1024$.

the outage probability for the delay, and we see that it increases as the threshold increases, which explains why the bias increases with the threshold. The fact that there is a bias for $\tau = 6.0$ while $P_{out} = 0$ is probably due to the number of Monte Carlo trials, which is not very high for this kind of simulation.

Finally, Fig. 7.7 shows the outage probability (i.e., the probability of wrong decoding) for various thresholds.

We do not have experimental measures for this quantity, since in our implementation, in order to evaluate the delay given that no decoding errors occur, we simply discard packets that present a decoding error, and the system acts as if they never existed. However, if we assume that the model is accurate enough and we compare it to Fig. 2.10 (using

Table 2.1 to keep the same gap to capacity) we have an error probability about two orders of magnitude smaller than the system without feedback schemes. This confirms the validity of the scheme in improving the finite-length performance of polar codes.

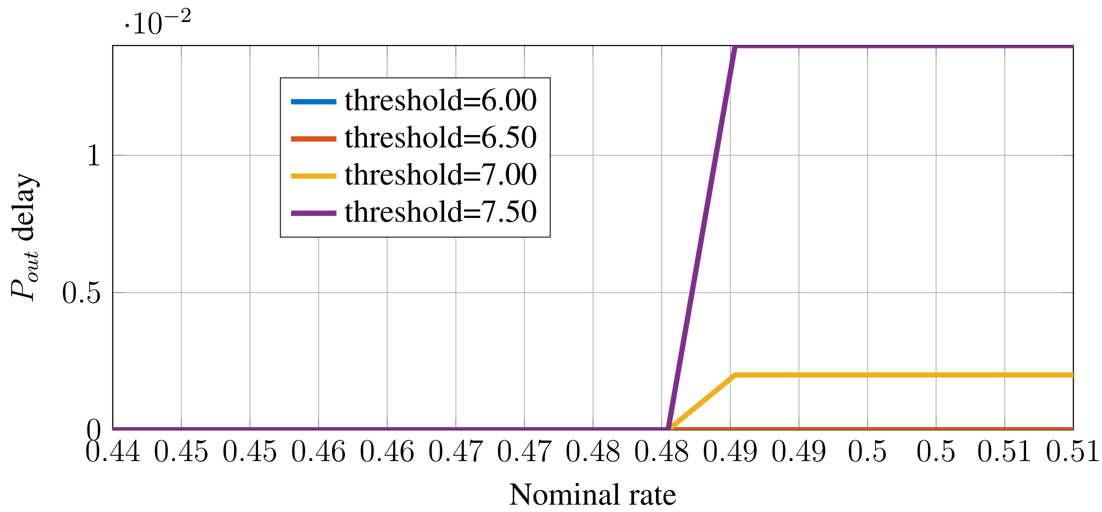


Figure 7.6: simulated outage probability for delay with $N = 1024$, threshold at 500.

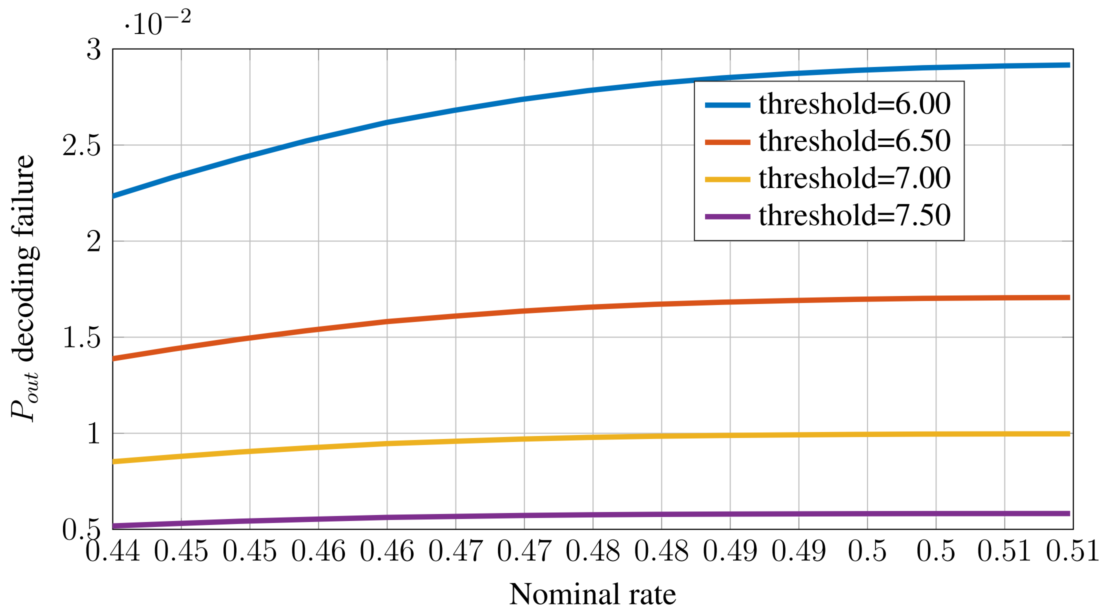


Figure 7.7: outage probability for decoding error given by MC with $N = 1024$.

Chapter 8

Bounds and Asymptotic Results

8.1 Bounds for the Delay

We want now to derive some simple bounds for scheme III applied to the BEC. This bounds will be asymptotic and pretty rough, but the will allow to avoid solving the exact Markov chain. For every state, we consider the expected value of associated transition probabilities, i.e.

$$\mathbb{E}_i = \sum_{j=0}^{i} jP_{i,j} \ \ \forall i > 0 \tag{8.1}$$

Intuitively, the expected arrival state starting from state *i* is the average number of errors in bits $0, ..., i-1$, which is given by $\sum_{j=0}^{i-1} Z^{(j)}$. Therefore, in order to simplify the analysis of the mean absorption time, we simplify the Markov chain. Since the average number of errors in bits $0, ..., i-1$, for every state i, is a crucial quantity, we would like to keep it fixed, but on the other hand we allow differences in the variance, i.e.,

$$Var_i = \sum_{j=0}^{i} j^2 P_{i,j} - \mathbb{E}_i^2 \ \ \forall i > 0 \tag{8.2}$$

Therefore, we consider another chain, with same average:

$$\mathbb{E}_i = \sum_{j=0}^{i} jP_{i,j} = \sum_{j=0}^{i} jP'_{i,j} = \mathbb{E}'_i \ \ \ \forall i > 0 \tag{8.3}$$

but different variance:

$$Var_i = \sum_{j=0}^{i} j^2 P_{i,j} - \mathbb{E}_i^2 \neq \sum_{j=0}^{i} j^2 P'_{i,j} - \mathbb{E}'^2_i = Var'_i \ \ \ \forall i > 0 \tag{8.4}$$

Experimentally, we verify (as we see in Fig. 8.1 and in detail in Fig. 8.2 with various models) that the delay increases as the variance decreases. Therefore, an upper bound can be found by considering the simplest Markov chain with the lowest variance.

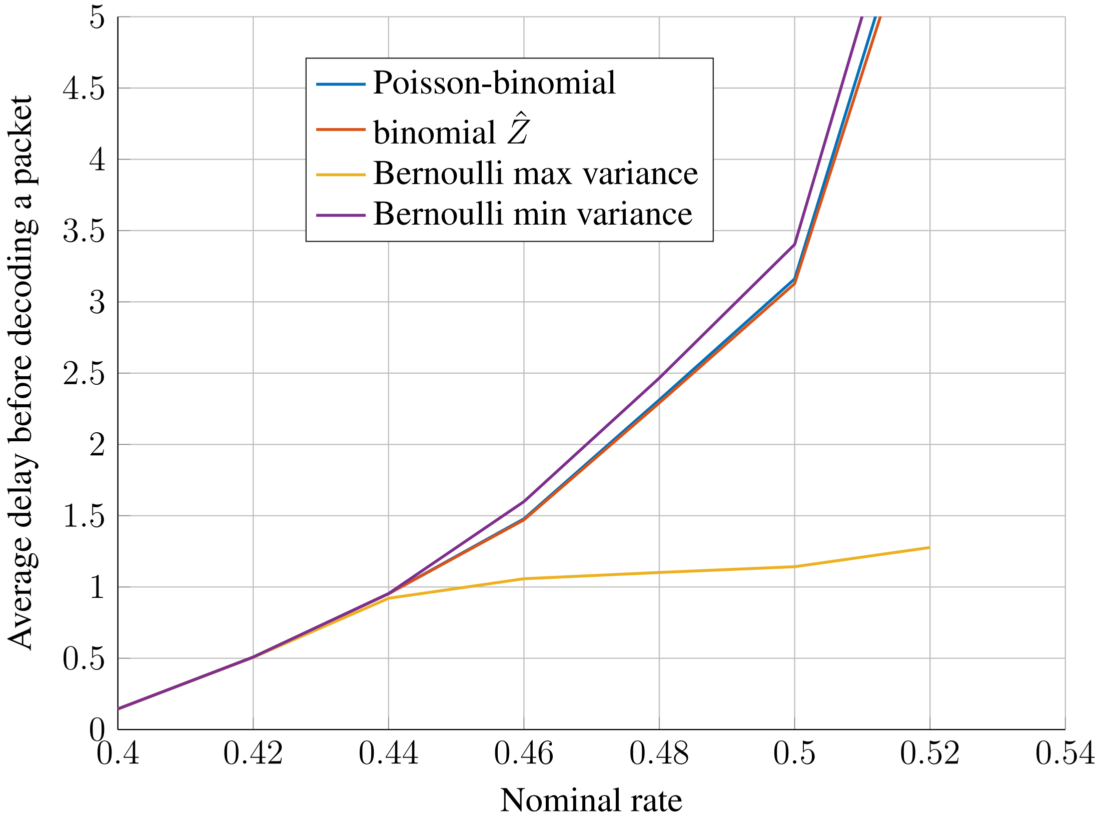


Figure 8.1: delay for various Markov models, $N = 4096$.

The models we considered in Figures 8.1 and 8.2 are:

1. the usual Poisson-binomial model presented in section 5.3.3
2. the binomial approximation of the Poisson-binomial, as in eq. 5.25, which is (see [22]) the distribution with maximum variance among all the Poisson-binomial distributions of fixed expected value
3. a Bernoulli approximation that maximizes the variance, that is the model of Fig. 8.3, and it is clearly the distribution with maximum variance among all possible distributions

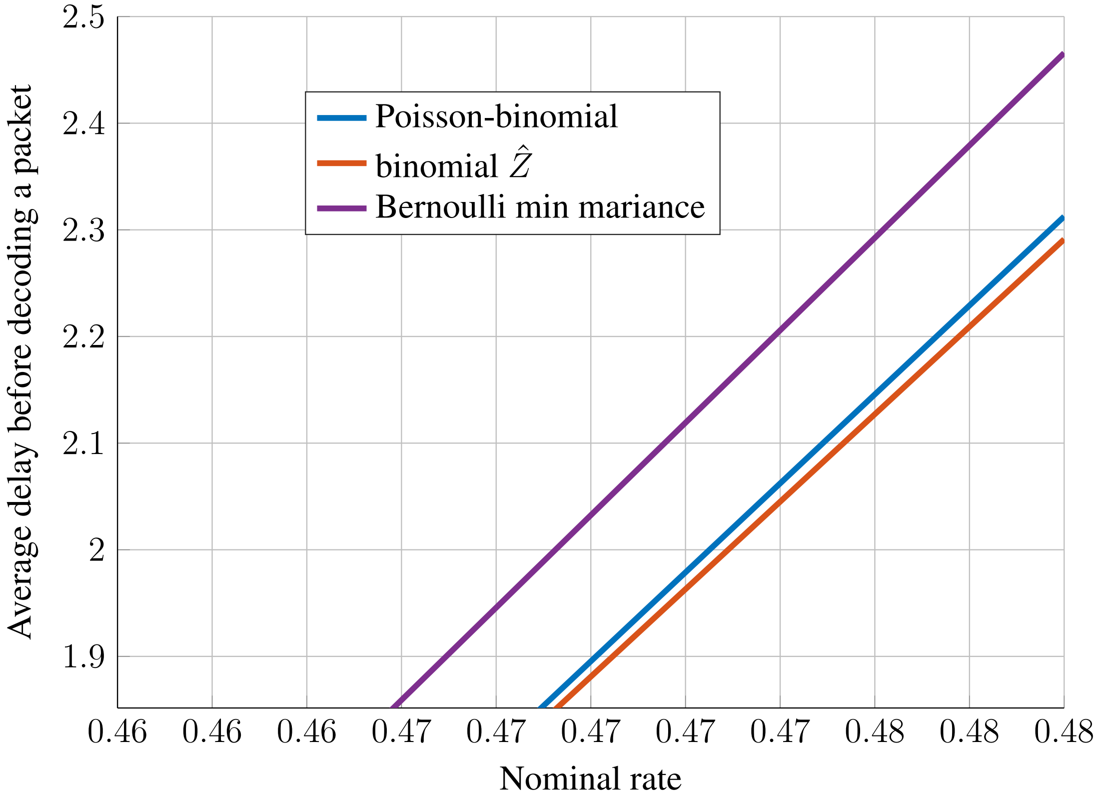


Figure 8.2: delay for various Markov models, detail, $N = 4096$.

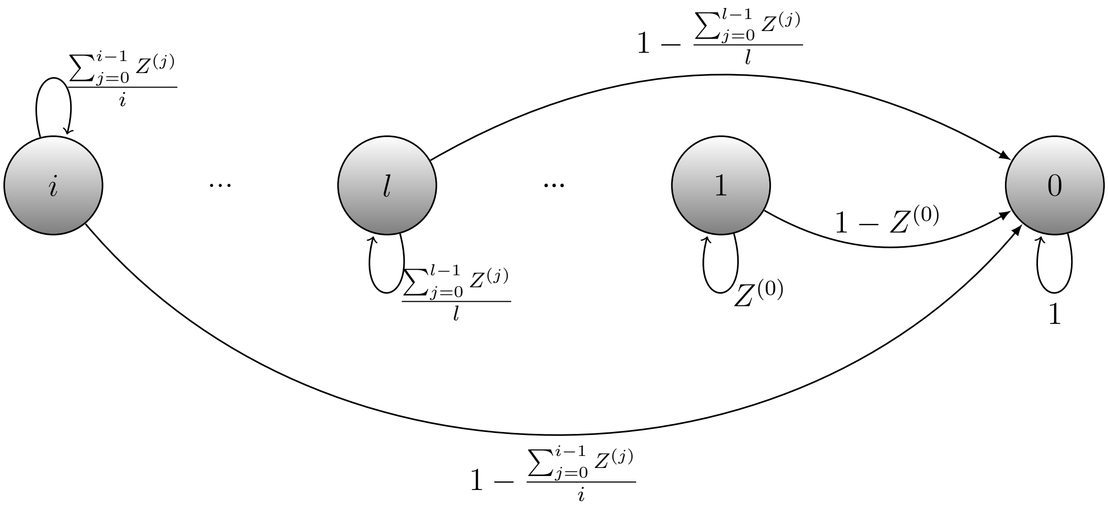


Figure 8.3: Bernoulli model with maximum variance.

4. a Bernoulli approximation that minimizes the variance, that is the model of Fig. 8.4, and it is the distribution that minimizes the variance by keeping exactly the same expected value. For this chain we have to determine arrival state l and

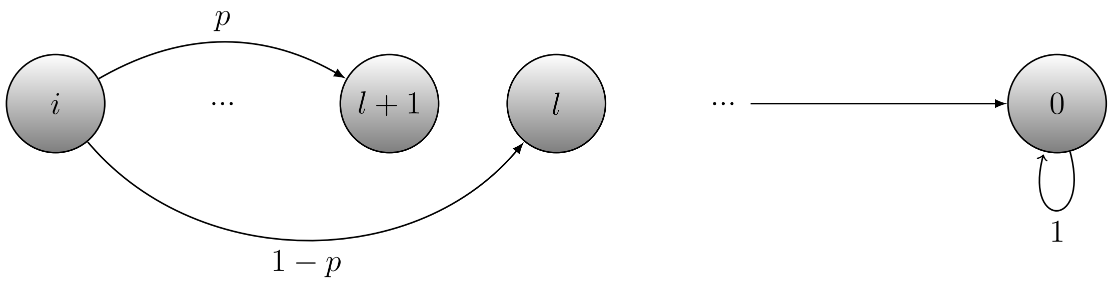


Figure 8.4: Bernoulli model with minimum variance.

probability p associated to transition $i \to l+1$. We have

$$\sum_{j=0}^{i-1} Z^{(j)} = p(l+1) + (1-p)l = p + l \tag{8.5}$$

since $0 \leq p \leq 1$ the natural choice is to take l as the integer and fractional parts of $\sum_{j=0}^{i-1} Z^{(j)}$ respectively. Therefore we have

$$\begin{aligned} l &= \lfloor \sum_{j=0}^{i-1} Z^{(j)} \rfloor \\ p &= \sum_{j=0}^{i-1} Z^{(j)} - \lfloor \sum_{j=0}^{i-1} Z^{(j)} \rfloor \end{aligned} \tag{8.6}$$

All the bounds we develop are for a given initial state i, that is an initial number of errors. Clearly, to derive the expected delay, we simply average them over the distribution of the number genie aids.

8.1.1 Upper Bounds

In order to derive upper bounds, we consider two regimes: a moderate errors regime, in which we have $\sum_{j=0}^{i-1} Z^{(j)} \geq 1$, for i initial state, and a rare errors regime, in which $\sum_{j=0}^{i-1} Z^{(j)} < 1$.

In the first case we will follow the reasoning of section 8.1 and build a simple chain that minimizes the variance. In the second case we will consider a simplification of the general Markov chain.

Moderate Errors Regime

We consider the chain, depicted in Fig. 8.5, that with probability 1 goes form state i to state $\sum_{j=0}^{i-1} Z^{(j)}$. If we want a discrete model, we can get the ceiling, i.e., we go from state i to state $\lceil \sum_{j=0}^{i-1} Z^{(j)} \rceil = l$ and then from state l to state $\lceil \sum_{j=0}^{l-1} Z^{(j)} \rceil = h$ and so on until we arrive to state 1. The recursion is depicted in Fig. 8.6. Note that these considerations assume that we start from a state i such that $\sum_{j=0}^{i-1} Z^{(j)} \geq 1$ (i.e., a regime with a moderate number of errors).

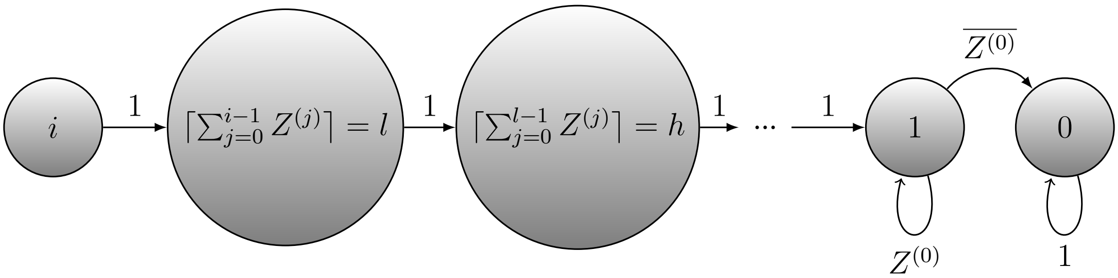


Figure 8.5: Markov model with minimum variance.

We now remove, for the sake of simplicity, the ceiling functions and we also consider a continuous interpolation of the partial sums. Therefore what we get is a process that goes from i to $\sum_{j=0}^{i-1} Z^{(j)} = x \in \mathbb{R}_+$, and then again to $\sum_{j=0}^{x-1} Z^{(j)} = y \in \mathbb{R}_+$ and so on. Solving the recursion requires the knowledge of all the partial sum of the Battacharrya parameters, which is equivalent to know all the parameters. In order to avoid this, we can consider the function which increases linearly up to the value $\lceil \sum_{j \in \mathcal{A}} Z^{(j)} \rceil$ with derivative $\max_{i \in \mathcal{A}} Z^{(i)}$ (i.e., we assume that all the first Battacharrya parameters are equal to the maximum, up to the sum), and then stays constant. It is not difficult to see that this function is greater or equal than the partial sums, and therefore the recursion gives a bigger delay. To the delay provided by the recursion, we must finally add a constant term given by the expected delay of going from state 1 to state 0.

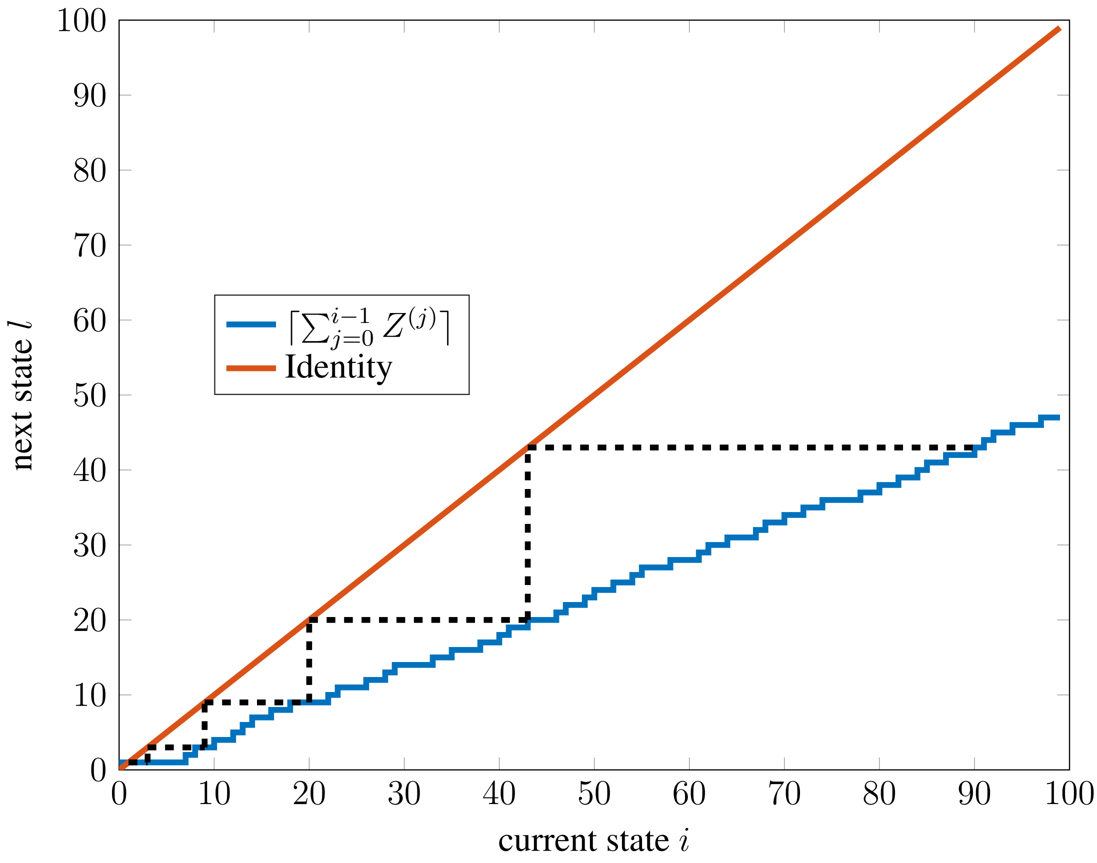


Figure 8.6: Fixed point recursion.

$$\begin{cases} M^{D(i)}i = 1 & \text{for } i < \frac{\lceil\sum_{j\in\mathcal{A}} Z^{(j)}\rceil}{M} \\ M^{D-1}\left(\lceil\sum_{j\in\mathcal{A}} Z^{(j)}\rceil\right) = 1 & \text{for } i \geq \frac{\lceil\sum_{j\in\mathcal{A}} Z^{(j)}\rceil}{M} \end{cases} \tag{8.7}$$

where $D-1$ in the second case is due to the fact that before starting with the recursion there is the first step $i \to \lceil\sum_{j\in\mathcal{A}} Z^{(j)}\rceil$ Therefore we obtain

$$M^{D(i)}i = 1 \Rightarrow D(i) = \min\left(\frac{\ln i}{\ln M^{-1}}, \frac{\ln\lceil\sum_{j\in\mathcal{A}} Z^{(j)}\rceil}{\ln M^{-1}} + 1\right) = \frac{\ln \mathscr{I}}{\ln M^{-1}} \tag{8.8}$$

where we put

$$M \triangleq \max_{i\in\mathcal{A}} Z^{(i)} \tag{8.9}$$

and

$$\mathscr{I} \triangleq \min\left(i, \frac{\lceil\sum_{j\in\mathcal{A}} Z^{(j)}\rceil}{M}\right) \tag{8.10}$$

As a first scaling result, we have that $\forall i$, if a maximum allowed delay D^{max} is fixed,

$$D(i) \leq D^{max} \Leftrightarrow M \leq e^{-\frac{\ln \mathscr{I}}{D^{max}}} \tag{8.11}$$

which, as proven in [23], requires

$$N \geq \left(\frac{ce^{-\frac{\ln \mathscr{I}}{D^{max}}}}{I(W) - R} \right)^{\mu} \tag{8.12}$$

The fact that a real interpolation of the partial sums is considered does not represent a problem, since a simple reasoning suffices to prove that twice the previous delay is an upper bound to the delay obtained with the ceiling (i.e., the one to which the above Markov chain refers). In fact, starting from state i, in 2 steps, by assuming we do not get stuck on a state (a case we exclude by hypothesis), we go to state $h < l \Rightarrow h \leq l - 1 = \lceil \sum_{j=0}^{i-1} Z^{(j)} \rceil - 1 \leq \sum_{j=0}^{i-1} Z^{(j)} \Rightarrow h \leq \sum_{j=0}^{i-1} Z^{(j)}$. Now, for the continuous case, consider the following process: with one step we go to the next state, and with the following one we stay in that state before proceeding to the next state in the third step and so on. Therefore, after two steps, we are in state $\sum_{j=0}^{i-1} Z^{(j)}$, which is greater that the former case. By induction we get the general conclusion.

Rare errors regime

In this section we consider an upper bound for the case where $\sum_{j=0}^{i-1} Z^{(j)} < 1$. In this case, in fact, the analysis provided above does not apply. Here, the simplest and most reasonable case we can consider is the chain that for every state i goes to state 1 with probability $\sum_{j=0}^{i-1} Z^{(j)}$ and to state 0 with probability $1 - \sum_{j=0}^{i-1} Z^{(j)}$.
In this case we have

$$D(i) = 1 + \frac{\sum_{j=0}^{i-1} Z^{(j)}}{1 - Z^{(0)}} \leq 1 + \frac{\sum_{i \in \mathcal{A}} Z^{(i)}}{1 - \sum_{i \in \mathcal{A}} Z^{(i)}} \simeq 1 + \frac{2^{-\sqrt{N}}}{1 - 2^{-\sqrt{N}}} \tag{8.13}$$

for N big enough.
Moreover, we also have that

$$\sum_{i \in \mathcal{A}} Z^{(i)} \leq \frac{D^{max} - 1}{D^{max}} < 1 \Rightarrow D(i) \leq D^{max} \tag{8.14}$$

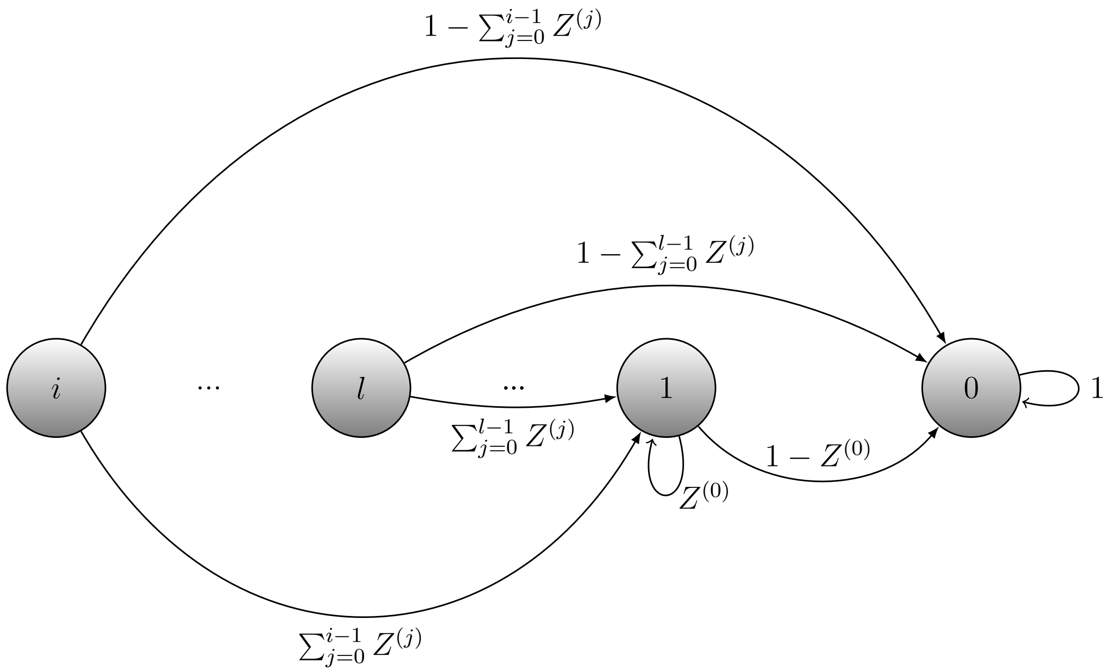


Figure 8.7: Markov model with maximum variance.

since in this case

$$D(i) = 1 + \frac{\sum_{j=0}^{i-1} Z^{(j)}}{1 - Z^{(0)}} \leq 1 + \frac{\sum_{i \in \mathcal{A}} Z^{(i)}}{1 - \sum_{i \in \mathcal{A}} Z^{(i)}} \leq D^{max} \tag{8.15}$$

and as proven in [23],

$$\sum_{i \in \mathcal{I}} Z^{(i)} \leq \frac{D^{max} - 1}{D^{max}} \tag{8.16}$$

requires as a necessary condition

$$N \geq \frac{\alpha}{(I(W) - R)^{\mu}} \tag{8.17}$$

8.1.2 Lower Bound

The chain with the higher variance is clearly the one that starting from state i stays in i with probability $\frac{\sum_{j=0}^{i-1} Z^{(j)}}{i}$ and goes to 0 with the complementary probability. In this case we obtain a geometrical random variable, and the expected delay is

$$D(i) = \frac{1}{1 - \frac{\sum_{j=0}^{i-1} Z^{(j)}}{i}} \geq \frac{1}{1 - \frac{\max_{j\in\{0\dots i-1\}} Z^{(j)}}{i}} \geq \frac{1}{1 - \frac{Z^{(0)}}{i}} \tag{8.18}$$

which is a very optimistic lower bound.

8.2 Asymptotic Behavior of the Effective Rate

In section 3.2.2 we defined the effective rate as

$$R_{eff} = \frac{K - M}{N} = R - \frac{M}{N} \tag{8.19}$$

with $M = \mathbb{E}[\text{number of retransmitted bits }] = \mathbb{E}[\text{number of genie helps }]$. Therefore we have

$$R_{eff} = R - \frac{\sum_{i=0}^{K-1} Z^{(i)}}{N} \tag{8.20}$$

but as an intermediate result of the asymptotic behavior of P_e (eq. 2.59) we have

$$\sum_{i=0}^{K-1} Z^{(i)} = o(2^{-N^{\beta}}) \tag{8.21}$$

for $\beta < \frac{1}{2}$. Therefore roughly speaking we can say that $\sum_{i=0}^{K-1} Z^{(i)}$ scales as $2^{-\sqrt{N}}$ for $N \to \infty$, and hence we have

$$R_{eff} \to R - \frac{2^{-\sqrt{N}}}{N} \text{ for } N \to \infty \tag{8.22}$$

Chapter 9

Conclusions and Future Work

The aim of this Thesis was to improve finite-length performance of polar codes, so that their usage could be made practically convenient.

To fulfill this goal we provided three schemes that take advantage of feedback to improve the packet error rate. They have been presented in an increasing order of complexity. However, we also verified that this complexity is not useless, since the most sophisticated scheme is able to achieve better performance than the others. For this reason the first scheme was discarded for practical purposes, and we concentrated our efforts on the other two, and mainly on the most complicated and interesting one.

Indeed, after having simulated the schemes, we concluded that in exchange for a moderate delay and a small rate reduction, they were able to provide a transmission with zero packet error rate.

Successively, stochastic models for the schemes were provided. In particular, we verified by simulation the accuracy of the model for scheme III, and we found it to be very good in estimating the expected delay for decoding.

The impact of list decoding was investigated. We concluded that the adoption of list decoding in place of successive cancellation is worthy only for small size of the list, as we increase the list size the gain we obtain is less and less significant.

The application of scheme III for the Gaussian channel was also investigated, given its importance in telecommunications. Scheme III and its theoretical model proved to be flexible enough to be easily adapted to the Gaussian case with just some minor but still

significant modifications. Again, we evaluated the performance of the scheme and the accuracy of the model for this case, and we found good results. This is promising for the practical applications of the scheme.
Finally, we greatly simplified the mathematical models in order to derive some simple bounds, and we developed an asymptotic analysis for the effective rate. As they are based themselves on bounds and scaling laws, these results are mainly asymptotic. If there is the necessity of avoiding the scheme simulation, and still obtain accurate estimations, the only viable alternative is the usage of the Markov model. However, as many scaling laws, they can provide valuable insight on the behavior of the system.
The main drawback of the schemes is clearly the presence of the delay. Therefore, their applicability is not universal, but it is limited only to those communication scenarios in which a delay is tolerated. For example, many real time communication scenarios have to be excluded, as well as the majority of broadcast transmissions. However, there is still a very large amount of situation that can take great advantage from this system, just as there is for the use of any ARQ technique. Just to provide some examples, bi-directional reliable communications, communications using TCP over IP, communications using FTP, and so on.

9.1 Future Work

The work of this Thesis opens and leaves many possibilities for future work and research.
The development of the schemes in case of non-full feedback is still at a primitive stage, and only some outlines have been given in appendix B.
It would be also very interesting to be able to find an explicit mathematical relation between the two most important metrics, namely R_{eff} and D.
More work on the mathematical analysis could be directed into find bounds for the delay that are tighter and more accurate.
The application of the model to other types of channels, such as the binary symmetric channel, could also be investigated.
An extension to M-ary channels could also be devised.

Finally, we analyzed the performance improvement provided by list decoding. However there exist many techniques developed to improve polar codes performance, such as using them in combination with other types of codes, or adding a CRC and so on. Evaluating how the scheme behaves when using these techniques in combination with polar codes would also be worthy.

Capitolo 9*b*

Conclusioni e Sviluppi Futuri

Lo scopo della presente Tesi era di migliorare le prestazioni a in regime di lunghezza di blocco finita dei codici polari, in modo da poter rendere il loro uso conveniente per impieghi pratici.
Per raggiungere questo scopo, abbiamo ideato tre schemi che traggono profitto dal feedback per miglioreare il tasso di errore di pacchetto. Essi sono stati presentati in ordine crescente di complessità. Comunque, abbiamo anche verificato che questa complessità non è inutile, dal momento che lo schema più sofisticato è in grado di raggiungere prestazioni migliori rispetto agli altri. Per questa ragione il primo schema è stato scartato per applicazioni pratiche, ed abbiamo concentrato i nostri sforzi sugli altri due, e principalmente sul più complicato ed interessante.
Infatti, dopo aver simualto gli schemi, abbiamo concluso che in cambio di un ritardo moderato e una piccola riduzione del rate, essi sono in grado di fornire una trasmissione con tasso di errore di pacchetto nullo.
Successivamente, modelli stocastici per gli schemi sono stati presentati. In particolare, abbiamo verificato attraverso simulazioni l'accuratezza del modello per lo schema III, e abbiamo verificato che è molto buono nello stimare il ritardo atteso per la decodifica.
Abbiamo investigato anche l'impatto della decodifica a lista. Abbiamo conlcuso che l'impiego della decodifica a lista al posto delle cancellazioni sequenziali è valida solo per una taglia della lista piccola, e incrementando la taglia della lista il guadagno che si ottiene è meno significativo.

Anche l'applicazione dello schema III per il canale gaussiano è stata oggetto di indagine, data la sua importanza nelle telecomunicazioni. Lo schema III e il relativo modello teorico si sono dimostrati sufficientemente flessibili da poter essere adattati al caso gaussiano con delle modifiche minori, ancorché significative. Di nuovo, abbiamo valutato le prestazioni dello schema e l'accuratezza del modello per questo scenario, e abbiamo trovato dei buoni risultati. Ciò è promettente per quanto riguarda le applicazioni pratiche dello schema.
In ultimo, abbiamo notevolmente semplificato i modelli matematici al fine di derivare qualche semplice limite, e abbiamo sviluppato un'analisi asintotica per il rate effettivo. Essendo essi stessi basati su limiti e leggi di scala, questi risultati sono principalmente asintotici. Se vi è la necessità di evitare la simulazione dello schema, ed ottenere comunque delle estime accurate, l'unica valida alternativa è l'uso del modello markoviano. Ad ogni modo, come molte leggi di scala, essi possono fornire una buona compresione del comportamento del sistema.
Il principale inconveniente degli schemi è chiaramente la presenza di ritardo. Perciò la loro applicabilità non è universale, ma è limitata solamente a quegli scenari di comunicazione in cui un ritardo è tollerabile. Per esempio, molti scenari di comunicazione in tempo reale devono essere esclusi, come anche molte trasmissioni in broadcast. Comunque, vi è ancora una gran quantità di situazioni che possono trarre un notevole vantaggio da questo sistema, come ve ne è per l'uso di qualunque tecnica ARQ. A mero titolo di esempio, comunicazioni bidirezionali affidabili, comunicazioni che usano TCP su IP, comunicazioni che usano FTP e così via.

9*b*.1 Sviluppi Futuri

Il lavoro di questa Tesi apre e lascia molte possibilità per futuri sviluppi e ricerche.
Lo sviluppo di schemi in caso di feedback non completo è ancora ad uno stadio embrionale, e solo qualche lineamento è stato dato in appendice B.
Sarebbe anche molto interessante saper trovare una relazione matematica esplicita tra le due metriche più importanti, ovvero R_{eff} e D.
Ulteriore lavoro sull'analisi matematica potrebbe essere diretto a trovare limiti per il

ritardo più stringenti e accurati.
Pure l'applicazione del modello ad altri tipi di canali, come il canale binario simmetrico, potrebbe essere oggetto di indagine. Anche un'estensione a canali M-ari potrebbe essere ideata.
Infine, abbiamo analizzato il miglioramento delle prestazioni fornito dalla decodifica a lista. Esistono però molte tecniche sviluppate per migliorare le prestazioni dei codici polari, come il loro uso in combinazione con altri tipi di codici, o l'aggiunta di CRC e così via. Varrebbe lo sforzo di valutare come questi schemi si comportano quando vengono usate queste tecniche in combinazione con i codici polari.

Appendix A

Channels

A.1 Binary Erasure Channel

The binary erasure channel of parameter ε, BEC(ε), is a binary channel which either correctly delivers the information or completely destroys it. It is a simple channel widely used in information theory, since it is suitable for theoretical analysis, whereas with other channels the analysis may result much more complicated.
It is characterized by its erasure probability ε and transition probabilities

$$\begin{cases} p(0|0) = p(1|1) = 1 - \varepsilon \\ p(?|0) = p(?|1) = \varepsilon \end{cases} \tag{A.1}$$

and diagram depicted in Fig. A.1. Its capacity is

$$C = I(W) = 1 - \varepsilon \tag{A.2}$$

A.2 Binary Additive White Gaussian Noise Channel

The binary additive white Gaussian noise channel of parameter σ, BAWGNC(σ_w) is a channel which takes binary input $x \in \{-1, +1\}$ (typically) and adds a white Gaussian

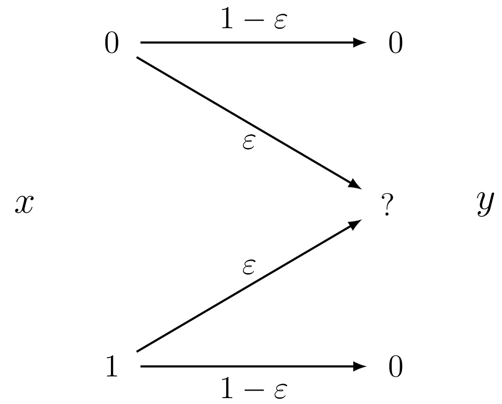


Figure A.1: diagram of BEC(ε).

noise, modeled as a random variable $w \in \mathcal{N}(0, \sigma_w^2)$.
The diagram is given in Fig. A.2. The capacity of the BAWGNC(σ_w) is derived in [24]:

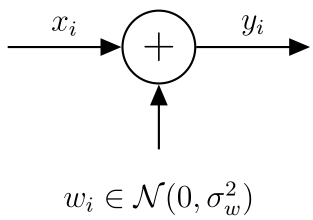


Figure A.2: diagram of BAWGNC(σ_w)

$$\begin{aligned} C(\sigma_w) =& 1 + \frac{1}{\ln 2}\left(\left(\frac{2}{\sigma_w^2} - 1\right) Q\left(\frac{1}{\sigma_w}\right) - \sqrt{\frac{2}{\pi\sigma_w^2}} e^{-\frac{1}{2\sigma_w^2}} + \right. \\ & \left. + \sum_{i=1}^{\infty} \frac{(-1)^i}{i(i+1)} e^{\frac{2i(i+1)}{\sigma_w^2}} Q\left(\frac{1+2i}{\sigma_w}\right)\right) \quad \text{[bit/channel use]} \end{aligned} \tag{A.3}$$

Appendix B

Considerations for the Case of Partial Feedback

In this case the system we propose is the following: the receiver transmits to the sender a feedback containing the indices of the first genie aids that it would need for all the received packets that still have to be decoded, ordered from the most recent to the oldest one (in order to take advantage of the backpropagation in the resolution of the genie helps). It also uses a special character (e.g., -1) to resolve any ambiguity on the attribution of the genie aid to each packet. In particular this happens when there is a packet that is decoded out of order, i.e., before a previously received packet that is still undecoded (an event that now, with the feedback that is not full anymore, may happen). However these special characters have to be sent only once per packet, since afterwards the sender keeps track that a given packet has been decoded at the receiver, and then acts consequently. The sender sends its packet adding at the beginning the genie aids requested by the receiver, putting first the genie aids regarding the most recently received packets (which allows us to exploit a propagation effect in the decoding). In the worst (decodable) case, that is all received packets have only the first bit decodable and the following t bits wrong, we verify that the delay to decode each packet is constant and finite, but it scales exponentially with the number of errors per packet t.
This system requires a feedback of size order of $M \log(R2^n)$ bits to encode the indices, where M is the number of pending received packets yet to be decoded at the receiver.

Bibliography

[1] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[2] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, pp. 3051–3073, July 2009.

[3] C. Leroux, I. Tal, A. Vardy, and W. J. Gross, "Hardware architectures for successive cancellation decoding of polar codes," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1665–1668, May 2011.

[4] C. Leroux, A. J. Raymond, G. Sarkis, I. Tal, A. Vardy, and W. J. Gross, "Hardware implementation of successive-cancellation decoders for polar codes," *Journal of Signal Processing Systems*, vol. 69, no. 3, pp. 305–315, 2012.

[5] G. Sarkis, P. Giard, A. Vardy, C. Thibeault, and W. J. Gross, "Fast polar decoders: Algorithm and implementation," *IEEE Journal on Selected Areas in Communications*, vol. 32, pp. 946–957, May 2014.

[6] B. Li, D. Tse, K. Chen, and H. Shen, "Capacity-achieving rateless polar codes," *CoRR*, vol. abs/1508.03112, 2015.

[7] S. Hong, D. Hui, and I. Maric, "Capacity-achieving rate-compatible polar codes," *CoRR*, vol. abs/1510.01776, 2015.

[8] T. Erseghe, *Channel coding*. Padova University Press, 2016.

[9] S. B. Korada, *Polar codes for channel and source coding*. PhD dissertation, La faculté Informatique et Communications, EPFL, Lausanne, Switzerland, 2009.

[10] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, "Llr-based successive cancellation list decoding of polar codes," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3903–3907, May 2014.

[11] S. H. Hassani, R. Mori, T. Tanaka, and R. L. Urbanke, "Rate-dependent analysis of the asymptotic behavior of channel polarization," *IEEE Transactions on Information Theory*, vol. 59, pp. 2267–2276, April 2013.

[12] M. Mondelli, R. Urbanke, and S. H. Hassani, "Unified scaling of polar codes: Error exponent, scaling exponent, moderate deviations, and error floors," in *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 1422–1426, June 2015.

[13] T. Erseghe, "Coding in the finite-blocklength regime: Bounds based on laplace integrals and their asymptotic approximations," *CoRR*, vol. abs/1511.04629, 2015.

[14] `http://sites.stat.psu.edu/~dhunter/asymp/lectures/asymp.pdf`. Accessed: 2016-04-15.

[15] M. B. Parizi and E. Telatar, "On the correlation between polarized becs," in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pp. 784–788, July 2013.

[16] J. S. L. Sean X. Chen, "Statistical applications of the poisson-binomial and conditional bernoulli distributions," *Statistica Sinica*, vol. 7, no. 4, pp. 875–892, 1997.

[17] M. Pinsky and S. Karlin, *An Introduction to Stochastic Modeling (Fourth Edition)*. Boston: Academic Press, fourth edition ed., 2011.

[18] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Transactions on Information Theory*, vol. 61, pp. 2213–2226, May 2015.

[19] A. Balatsoukas-Stimming, M. B. Parizi, and A. Burg, “Llr-based successive cancellation list decoding of polar codes,” *IEEE Transactions on Signal Processing*, vol. 63, pp. 5165–5179, Oct 2015.

[20] M. Mondelli, S. H. Hassani, and R. L. Urbanke, “Scaling exponent of list decoders with applications to polar codes,” *IEEE Transactions on Information Theory*, vol. 61, pp. 4838–4851, Sept 2015.

[21] R. Mori and T. Tanaka, “Performance of polar codes with the construction using density evolution,” *IEEE Communications Letters*, vol. 13, pp. 519–521, July 2009.

[22] Y. H. Wang, “On the number of successes in independent trials,” *Statistica Sinica*, vol. 3, no. 2, pp. 295–312, 1993.

[23] S. H. Hassani, K. Alishahi, and R. L. Urbanke, “Finite-length scaling for polar codes,” *IEEE Transactions on Information Theory*, vol. 60, pp. 5875–5898, Oct 2014.

[24] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.