Universidad Politécnica de Madrid
Facultad de Informática
Computer Languages and Systems and Software Engineering Department

Efficient Information Reconciliation for Quantum Key Distribution

Reconciliación Eficiente de Información para la Distribución Cuántica de Claves

Ph.D. Thesis

Jesús Martínez Mateo

Madrid, 2011

Efficient Information Reconciliation for Quantum Key Distribution

Dissertation submitted to the

Universidad Politécnica de Madrid

in conformity with the requirements for the Degree of

Doctor of Phylosophy.

Author: Mr. Jesús Martínez Mateo

Bachelor Degree in Computer Science Engineering

Master in Computational Mathematics

Supervisor: Dr. Vicente Martín Ayuso

Ph.D. in Physics

Associate Professor at UPM

Madrid, 2011

Tribunal nombrado por el Magnífico y Excelentísimo Sr. Rector de la Universidad Politécnica de Madrid,

Presidente: .....

Vocal: .....

Vocal: .....

Vocal: .....

Secretario: .....

Realizado el acto de lectura y defensa de la Tesis Doctoral en Madrid, a ..... de ..... de 20..... .

El tribunal acuerca entregar la calificación de .....

El Presidente Los Vocales

El Secretario

Abstract

Advances in modern cryptography for secret-key agreement are driving the development of new methods and techniques in key distillation. Most of these developments, focusing on information reconciliation and privacy amplification, are for the direct benefit of quantum key distribution (QKD). In this context, information reconciliation has historically been done using heavily interactive protocols, i.e. with a high number of channel communications, such as the well-known *Cascade*. In this work we show how modern coding techniques can improve the performance of these methods for information reconciliation in QKD. Here, we propose the use of low-density parity-check (LDPC) codes, since they are good both in efficiency and throughput. A price to pay, a priori, using LDPC codes is that good efficiency is only attained for very long codes and in a very narrow range of error rates. This forces to use several codes in cases when the error rate varies significantly in different uses of the channel, a common situation for instance in QKD. To overcome these problems, this study examines various techniques for adapting LDPC codes, thus reducing the number of codes needed to cover the target range of error rates. These techniques are also used to improve the average efficiency of short-length LDPC codes based on a feedback coding scheme. The importance of short codes lies in the fact that they can be used for high throughput hardware implementations. In a further advancement, a protocol is proposed that avoids the a priori error rate estimation required in other approaches. This *blind* protocol also brings interesting implications to the finite key

analysis.

Keyword: quantum key distribution, key distillation, information reconciliation, low-density parity-check codes, rate-adaptive, feedback coding.

Resumen (Spanish)

Los avances en la criptografía moderna para el acuerdo de clave secreta están empujando el desarrollo de nuevos métodos y técnicas para la destilación de claves. La mayoría de estos desarrollos, centrados en la reconciliación de información y la amplificación de privacidad, proporcionan un beneficio directo para la distribución cuántica de claves (QKD). En este contexto, la reconciliación de información se ha realizado históricamente por medio de protocolos altamente interativos, es decir, con un alto número de comunicaciones, tal y como ocurre con el protocolo *Cascade*. En este trabajo mostramos cómo las técnicas de codificación modernas pueden mejorar el rendimiento de estos métodos para la reconciliación de información en QKD. Proponemos el uso de códigos *low-density parity-check* (LDPC), puesto que estos son buenos tanto en eficiencia como en tasa de corrección. Un precio a pagar, a priori, utilizando códigos LDPC es que una buena eficiencia sólo se alcanza para códigos muy largos y en un rango de error limitado. Este hecho nos obliga a utilizar varios códigos en aquellos casos en los que la tasa de error varía significativamente para distintos usos del canal, una situación común por ejemplo en QKD. Para superar estos problemas, en este trabajo analizamos varias técnicas para la adaptación de códigos LDPC, y así poder reducir el número de códigos necesarios para cubrir el rango de errores deseado. Estas técnicas son también utilizadas para mejorar la eficiencia promedio de códigos LDPC cortos en un esquema de codificación con retroalimentación o realimentación (mensaje de retorno). El interés de los códigos cortos reside en el

hecho de que estos pueden ser utilizados para implementaciones hardware de alto rendimiento. En un avance posterior, proponemos un nuevo protocolo que evita la estimación inicial de la tasa de error, requerida en otras propuestas. Este protocolo *ciego* también nos brinda implicaciones interesantes en el análisis de clave finita.

Palabras clave: distribución cuántica de claves, reconciliación de información, códigos low-density parity-check, adaptación de la tasa de información, codificación con reentrada de información.

Acknowledgments

I would like to especially thank Prof. Vicente Martín from the Research Group on Quantum Information and Computation, Faculty of Computer Sciences, at Universidad Politécnica de Madrid, and David Elkouss, for their help in the preparation of this work. This would not have been possible without their contribution, support and helpful discussions. I am deeply indebted to them.

I would also like to thank Prof. Nicolas Gisin and Dr. Hugo Zbinden from the Group of Applied Physics, GAP-Optique[1], at Geneva University, and Dr. Andreas Burg from the Integrated Systems Laboratory[2] at ETH Zürich (Eidgenössische Technische Hochschule Zürich), for their help and support during my stay in Switzerland. They opened the doors for me to participate in two internationally renowned laboratories.

I am also very grateful to all the people with whom I shared my research time and interest in Switzerland: Dr. Jürg Treichler, Pascal Meinerzhagen and Christoph Roth at ETH Zurich; Dr. Patrick Eraerds, Dr. Jun Zhang, Jean-Daniel Bancal, Claudio Barreiro, Raphael Houlmann, and Ci Wen (Charles) Lim at GAP-Optique, Geneva. And especially, I would like to express my gratitude to Nino Walenta for his help before, after and during my stay in Geneva, helpful discussions, and for the time he spent to teach me the intricacies of physics and technology of quantum key distribution

[1] http://www.gap-optique.unige.ch

[2] http://www.iis.ee.ethz.ch

devices.

This work has been partially supported by grant UPM/CAM Q061005127 funded by Universidad Politécnica de Madrid and Comunidad Autónoma de Madrid, project SEGURA@ (Third Call of the CENIT Programme) funded by the Centre for the Development of Industrial Technology, Ministry of Industry and Trade, and project Quantum Information Technologies in Madrid[3] (QUITEMAD), Project P2009/ESP-1594, funded by Comunidad Autónoma de Madrid.

The author gratefully acknowledge the computer resources, technical expertise and assistance provided by the *Centro de Supercomputación y Visualización de Madrid*[4] (CeSViMa) and the Spanish Supercomputing Network.

[3]http://www.quitemad.es

[4]http://www.cesvima.upm.es

List of Publications

Scientific Journals

1. Jesus Martinez-Mateo, David Elkouss, and Vicente Martin (2011), *Blind Reconciliation*, submitted to Quantum Information and Computation.

2. David Elkouss, Jesus Martinez-Mateo, and Vicente Martin (2011), *Untainted Puncturing for Irregular Low-Density Parity-Check Codes*, submitted to IEEE Communications Letters, arXiv:1103.6149 [cs.IT].

3. David Elkouss, Jesus Martinez-Mateo, and Vicente Martin (2011), *Information Reconciliation for Quantum Key Distribution*, Quantum Information and Computation, Vol. 11, No. 3&4, pp. 226-238, arXiv:1007.1616 [quant-ph].

4. Jesus Martinez-Mateo, David Elkouss, and Vicente Martin (2010), *Improved construction of irregular progressive edge-growth Tanner graphs*, IEEE Communications Letters, Vol. 14, No. 12, pp. 1155-1157, arXiv:1007.3588 [cs.IT].

Conference Proceedings

1. David Elkouss, Jesus Martinez-Mateo, and Vicente Martin (2010), *Secure rate-adaptive reconciliation*, in ISITA 2010, International Symposium on Information Theory and its Applications, Taichung (Taiwan), October 17-20, pp. 179-184, arXiv:1007.0904 [cs.IT].

2. Jesus Martinez-Mateo, David Elkouss, and Vicente Martin (2010), *Interactive Reconciliation with Low-Density Parity-Check Codes*, in ISTC 2010, 6th Int. Symposium on Turbo Codes & Iterative Information Processing, Brest (France), September 6-10, pp. 270-274, arXiv:1006.4484 [cs.IT].

3. D. Elkouss, J. Martinez, D. Lancho, and V. Martin (2010), *Rate Compatible Protocol for Information Reconciliation: An application to QKD*, in ITW 2010, IEEE Information Theory Workshop, Cairo (Egypt), January 6-8, pp. 145-149, arXiv:1006.2660 [cs.IT].

4. D. Lancho, J. Martinez, D. Elkouss, M. Soto, and V. Martin (2009), *QKD in Standard Optical Telecommunications Networks*, in QuantumComm 2009, First International Conference on Quantum Communication and Quantum Networking, Naples (Italy), October 26-30, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), Vol. 36, pp. 142-149, arXiv:1006.1858 [quant-ph].

5. J. Davila, D. Lancho, J. Martinez, and V. Martin (2009), *On QKD Industrialization*, in QuantumComm 2009, First International Conference on Quantum Communication and Quantum Networking (Workshop: Quantum and Classical Information Security), Naples (Italy), October 26-30, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (LNICST), Vol. 36, pp. 297-302.

Patents

1. M. Soto, D. Menendez, J.A. Pozas, V. Martin, D. Lancho, and J. Martinez, WO/2011/036322 (A2), *System for Integration of Channels with Quantum Information in Communication Networks*.

2. V. Martin, D. Lancho, D. Elkouss, and J. Martinez, Application number P201030099,

Método y sistema de comunicaciones para la reconciliación de información en QKD mediante el uso de códigos LDPC adaptando la tasa de información.

Other Intellectual Property

1. D. Elkouss, J. Martinez, and V. Martin, *QKD-Cascade*, Registro Territorial de la Propiedad Intelectual, Comunidad de Madrid, No. M-008489/2009, Exp. 12/RTPI-009159/2009, Ref. 12/068660.3/09.

Contents

List of Figures

List of Tables

List of Algorithms

Part I

Introduction and Preliminaries

Chapter 1

Introduction

Charles H. Bennett and Gilles Brassard combined two originally unconnected disciplines, cryptography and quantum mechanics, to propose the first quantum cryptography protocol in 1984 [1]. It is then when the well-known BB84 protocol came to light, and a new discipline emerged with this: *quantum cryptography*[1]. In the BB84 protocol two legitimate parties, typically named Alice and Bob, want to agree on a common information-theoretic secret-key even in the presence of any adversary or eavesdropper, typically named Eve. To accomplish this, the parties first communicate through a quantum channel (e.g. free air or optical fiber) whereby the parties exchange quantum states (e.g. single photons) carrying classical information, the *raw key*. In a second step, the parties communicate over a noiseless, public and authenticated channel to conclude the protocol with a basis reconciliation procedure, also referred as sifting. At this point, assuming perfect communication —i.e. without noise in the quantum channel nor in the devices— and no eavesdropping, Alice and Bob share two identical strings, the *sifted key*. Both procedures belong to a family of secret-key agreement protocols known as *quantum key distribution* (QKD) or *quantum*

[1] It should be noted that the origin of quantum cryptography is probably due to the commonly unknown but original idea of *quantum money* proposed by Stephan Wiesner [2].

key growing[2].

Unfortunately, any physical implementation of a QKD protocol is affected by imperfections in the devices and the quantum channel. These imperfections introduce noise into the shared strings, making them different. In addition, further discrepancies can be also introduced by any hypothetical eavesdropper wiretapping the quantum channel, and thus modifying the transmitted information. Therefore, both parties, Alice and Bob, need to reconcile those discrepancies in their strings to make them identical. This process is known as *information reconciliation*, or simply reconciliation [4, 5]. The parties disclose side information about the shared strings through a public but authenticated channel, such that this information can be read but it cannot be modified by an adversary. The adversary or eavesdropper gains information about the secret-key both by wiretapping the quantum channel and listening the public discussion for the reconciliation. Afterward, the parties must agree on an additional procedure, called *privacy amplification* [6, 7], used to reduce the information about the key that may have been derived by any eavesdropper. In the privacy amplification procedure the parties amplify the uncertainty of the eavesdropper at the expense of compressing their shared strings. Then, the parties generate a common information-theoretic secret-key. Both procedures, information reconciliation and privacy amplification, are part of a process known as *key distillation* [8–10].

An optimal reconciliation procedure discloses the minimum information required for correcting all discrepancies between two previously shared keys, thus minimizing the key material that is discarded during the privacy amplification, and maximizing the final secret-key length. The performance of a QKD protocol —i.e. secret-key rate— depends on both (1) the efficiency in the preparation, manipulation and measurement of quantum states [3], and (2) the efficiency of those classical procedures

[2]A QKD protocol uses part of an exchanged key to authenticate following communications. Therefore, the protocol also requires an initial secret-key, shared by both parties, to authenticate the first communications. We can see then a QKD protocol as a quantum key growing (or quantum secret growing) protocol, where from an initial secret the parties generate a larger secret-key [3].

carried out for key distillation.

We consider in this work the scenario described above where the quantum channel is a noisy channel represented by a binary symmetric channel (BSC) with crossover probability ϵ, since errors in the quantum channel are considered uncorrelated (discrete memoryless channel) and symmetric. It should be noted that in this work we refer to the error rate in the channel as crossover probability, while in the QKD literature it is usually known as quantum bit error rate or QBER.

1.1 Background

Information reconciliation in the secret-key agreement context is a problem already studied for the authors of the original BB84 protocol [4–6]. In the pioneer BBBSS protocol [4] the authors propose a reconciliation protocol based in the exchange of a number of syndromes[3] per transmitted key. A syndrome consists of a set of parity-check equations, and a block (as used in Refs. [4,5]) consists of all symbols or key bits involved in a parity-check equation. If any parity-check equation of an exchanged syndrome is not verified, the parties carry out a binary or dichotomic search to find the corresponding error within that block. Note that, in each parity-check equation only an odd number of errors can be detected, but only one of them can be corrected using a binary search. The procedure works iteratively shuffling the bits of the key to reconcile, and exchanging successive syndromes.

Later, in Ref. [5] the authors realized that each located error produces side information that can be used with a previously exchanged syndrome. The new protocol is called *Cascade* in reference to the iterative or cascading process of identifying errors after each new error found. The protocol is characterized by one parameter: the block size, i.e. the number of bits involved in every parity-check equation of the syndrome.

[3]Originally, the authors write about the parity of blocks without introduce the concept of *syndrome* commonly used in the communication and information theory community. For convenience and consistency with the classic literature in the community we prefer to use this concept.

An initial value for this block size k_1 is determined depending on the channel parameter ϵ or quantum bit error rate, $k_1 \approx 0.73/\epsilon$ [11]. In each step the block size is doubled, i.e. the block size for the i-th step is determined as $k_i = 2k_{i-1}$.

Several optimizations were proposed for the BBBSS and *Cascade* protocols [12–20]. However, these protocols are highly interactive since they require many communication rounds. The parties have to exchange a large number of messages where parities of different blocks and sub-blocks of a sifted key are transmitted. Despite of the interactivity of *Cascade*, it continues being one of the most widely used protocols for information reconciliation in QKD, probably due to its simplicity and relatively good efficiency.

Other protocols have been also proposed in the literature. For instance, in *Winnow* the authors propose the use of Hamming codes for the calculation of separate syndromes in each block instead of a simple parity-check equation [21]. However, the efficiency of this protocol is still far from the theoretical limit or Shannon limit (see Section 2.3 below).

As early as 2003 Chip Elliott, from the DARPA group in Los Alamos, hinted at the use of parity-checks as in telecommunication systems [22], but the group did not present any result referring to the use of low-density parity-check (LDPC) codes until one year later [23,24]. This is one of the first applications of the modern coding theory [25] to the information reconciliation problem in QKD. Furthermore, the use of LDPC codes in the field remained stalled until 2009, it is then when specific codes for several information rates were designed taking into account the communication channel considered for QKD [26]. Recently, LDPC codes are becoming of interest in QKD and they have also been implemented for QKD networks [27,28].

1.2 Motivation

Since the publication of the first quantum cryptography protocol, QKD has matured into a commercial technology. Nowadays, it is even possible to find manufacturers of QKD systems selling their products. Nevertheless, this technology is still far from reaching its potential due to the lack of suitable developments in some of its fundamental processes, such as the key distillation or, more precisely, the key reconciliation step.

As commented above, information reconciliation for the secret-key agreement is a problem already addressed in the literature. Currently, we can find some well-known protocols for this purpose, such as the previously mentioned *Cascade*. This is a protocol specifically designed for information reconciliation in the QKD context. It attempts to minimize the information disclosed using an estimation of the error rate in the sifted key. However, it has the drawback of interactivity, requiring lots of channel uses that limit the final rate of reconciled key. As an alternative to *Cascade*, we analyze several modern techniques for correcting errors adapted to the problem of information reconciliation. We focus this study on three main objectives: (1) improve the efficiency in the reconciliation process —i.e. minimize the amount information disclosed during the reconciliation—, (2) reduce the number of messages communicated through the channel, and (3) increase the throughput of final reconciled key.

This work focuses on the use of binary low-density parity-check (LDPC) codes [29–31] for the information reconciliation problem. LDPC codes were specially designed for the binary symmetric channel (BSC), and adapted for the source coding with side information problem using *syndrome source coding*. Several techniques for the construction of rate-adaptive LDPC codes are studied. Using these techniques, LDPC codes are adapted for different error rates, minimizing the information published for reconciliation. A protocol based on a feedback communication scheme is also analyzed to improve the average efficiency of short length rate-adaptive LDPC codes. The proposed interactive approach, called *blind* reconciliation, allows to re-

concile a key without any channel parameter estimation. This work is based on the previously published papers [26,32–37].

Chapter 2

Notation and Information-Theoretic Preliminaries

This research, as many others, is a consequence of the original work published by Claude E. Shannon in 1940s. We highly recommend reading his two famous contributions [38,39] where the author introduces both disciplines, information theory and information-theoretic security (or theoretical secrecy as referred by Shannon), related to the secret-key distillation here discussed.

Information summarized here has been mostly obtained from the book of Thomas M. Cover and Joy A. Thomas [40]. In addition, the books of the following authors have also been consulted throughout this work, David J.C. MacKay [41], Gilles Van Assche [42], Susan Loepp and William K. Wootters [43], and Thomas J. Richardson and Rüdiger L. Urbanke [25].

2.1 Notation

For convenience to the reader, this contribution was written using the same notation throughout the document. Random variables are denoted in capital letters using the final letters of the alphabet, X, Y, Z. Matrices are also denoted in uppercase but using

preferably the first letters of the alphabet, A, B, C. Vectors are denoted in boldface lowercase, $\mathbf{x}, \mathbf{y}, \mathbf{z}$, and each element of a vector is denoted by a subscript, always beginning by 1, x_1, x_2, x_3. $\mathbf{x}^t$ and H^t denote the transpose vector and the transpose matrix, respectively. Finally, sets are denoted in uppercase using Calligraphy style, $\mathcal{A}, \mathcal{B}, \mathcal{C}$.

2.2 Discrete Probability Theory

2.2.1 Uncertainty and Entropy

Entropy

Let X be a discrete random variable with possible values within the set $\mathcal{X}$, and let $p_X(x)$ be the probability mass function defined for each value $x \in \mathcal{X}$, $p_X(x) = \Pr(X = x)$, such that $\sum_{x \in \mathcal{X}} p_X(x) = 1$. We denote this probability mass function by $p(x)$ rather than $p_X(x)$, for convenience.

Definition 1. *Let X be a discrete random variable with discrete alphabet $\mathcal{X}$ and probability mass function $p(x)$. The entropy, or Shannon entropy [38], of X is given by:*

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x) \tag{2.1}$$

This entropy is a measure of the average uncertainty of a single random variable. Unless stated otherwise we use the logarithm base 2, even when the subscript is omitted for convenience.

When the alphabet $\mathcal{X}$ consists of two values, the Shannon entropy is measured in bits. The associated function, typically referred as *binary Shannon entropy* and denoted by $h(p)$, is given by:

$$h(p) = -p \log_2 p - (1 - p) \log_2 (1 - p) \tag{2.2}$$

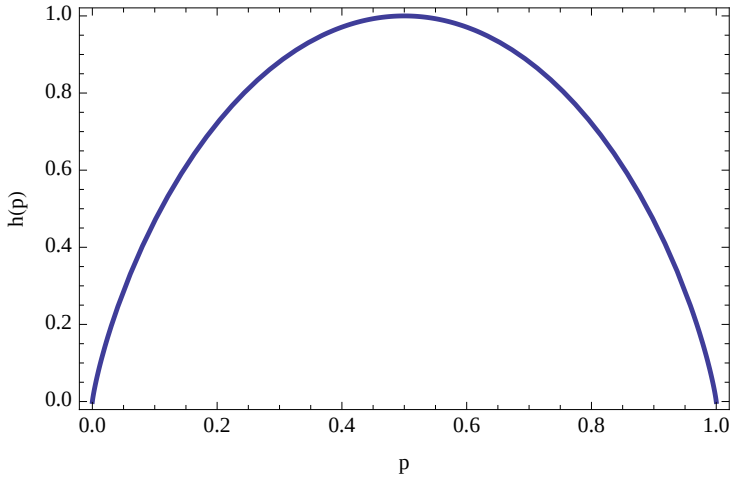

Figure 2.1: Binary Shannon entropy.

where we consider p is the probability that the bit value is 0, $p = \Pr(x = 0)$, and the complementary $1 - p$ the probability of 1, since $\Pr(x = 0) + \Pr(x = 1) = 1$.

Notice that, although $0 \log_2 0$ is not mathematically defined, the convention $0 \log_2 0 = 0$ is used. Figure 2.1 shows how this function is maximized for the equiprobable case, $p = 1/2$.

Conditional Entropy

We can define conditional entropy, denoted by $H(X|Y)$, as the entropy of a random variable conditional on the knowledge of another random variable.

Definition 2. *Let X and Y be two discrete random variables with discrete alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively, and joint probability distribution $p(x, y)$. The conditional entropy of X given Y is defined as:*

$$H(X|Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x|y) \tag{2.3}$$

Mutual Information

The mutual information $I(X;Y)$ is a measure of the dependence between the two random variables.

Definition 3. *Let X and Y be two discrete random variables with discrete alphabets $\mathcal{X}$ and $\mathcal{Y}$, respectively, and joint probability distribution $p(x,y)$. The mutual information of X and Y is given by:*

$$I(X;Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \tag{2.4}$$

We can also define the mutual information between two random variables as a function of the entropy and the conditional entropy of both variables as follow:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) && (2.5)\\ &= H(Y) - H(Y|X) && (2.6)\\ &= I(Y;X) && (2.7) \end{aligned}$$

As shown in the previous equation, the mutual information is symmetric in X and Y. This is always non-negative, and equal to zero if and only if X and Y are independent variables.

2.3 Communication Theory

When considering the information reconciliation problem in the QKD context we commonly use a discrete memoryless channel to model how errors occur during the communication through the quantum channel. We usually model this discrete memoryless channel as binary symmetric channel (BSC), since it consists of two binary alphabets for input and output, X and Y respectively, and a symmetric probability transition matrix $p(y|x) = p(x|y)$, such that $x \in X$ and $y \in Y$. This channel is said

to be memoryless when the probability of observing the output symbol y, given that the symbol x was transmitted, $p(y|x)$, is conditionally independent of previous inputs and outputs in the channel.

Communication Channel

A *communication channel* is a system in which the output depends probabilistically on the input. This is characterized by a probability transition matrix $p(y|x)$ defined for each value of x and y, input and output respectively, that determines the conditional distribution of the output given the input.

Channel Capacity

Let X and Y be two variables taking values within an input and output alphabet, $\mathcal{X}$ and $\mathcal{Y}$ respectively, both representing the input and output of a communication channel.

Definition 4. *The capacity C for this channel is defined as the maximum mutual information between input and output.*

$$C = \max_{p(x)} I(X;Y) \tag{2.8}$$

where the maximum is taken over all possible input distributions $p(x)$.

Information Rate

Definition 5. *The information rate R is the ratio between the bits carrying information, k, and the total amount of bits sent, n.*

$$R = \frac{k}{n} = \frac{n-m}{n} \tag{2.9}$$

where m is, thus, the bits of redundancy.

For every communication channel there exist an upper bound on the information rate typically called the Shannon limit, it is the channel capacity. Thus, $R \leq C$.

Binary Symmetric Channel

The binary symmetric channel, or BSC, is one the simplest channels for communications where errors are considered. In this discrete memoryless channel, input and output values are considered within the binary alphabet $\mathcal{B} = \{0,1\}$. An error is represented by a transition from one value to the other as shown in Figure 2.2. The channel is considered symmetric since the probability (crossover probability) to get an error in the channel, ϵ, is considered constant for every symbol. Thus, the channel is fully characterized by its crossover probability, ϵ, and it is commonly denoted by BSC(ϵ).

The mutual information for input and output in this channel is upper bounded by [40]:

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) && (2.10)\\ &= H(Y) - \sum p(x) H(Y|X=x) && (2.11)\\ &= H(Y) - \sum p(x) H(\epsilon) && (2.12)\\ &= H(Y) - H(\epsilon) && (2.13)\\ &\leq 1 - H(\epsilon) && (2.14) \end{aligned}$$

The capacity as defined in Eq. (2.8) is then given by:

$$C_{\mathrm{BSC}(\epsilon)} = 1 - h(\epsilon) \tag{2.15}$$

where $h(\epsilon)$ is the binary Shannon entropy as defined in Eq. (2.2).

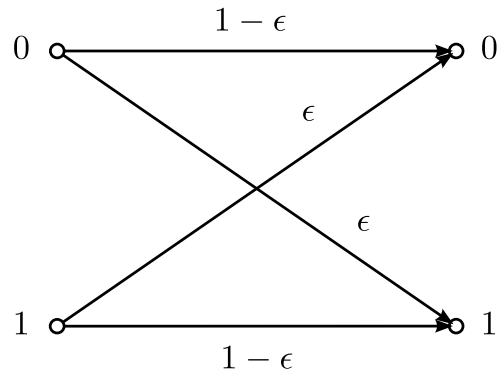

Figure 2.2: Binary symmetric channel with crossover probability ϵ.

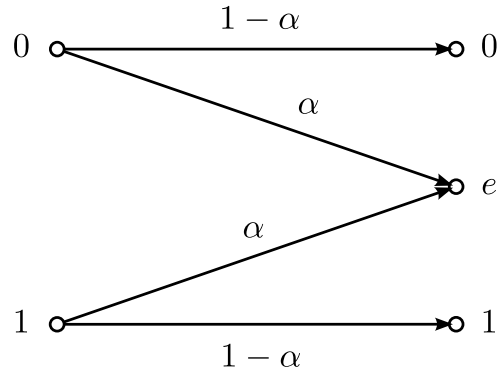

Figure 2.3: Binary erasure channel with erasure probability α.

Binary Erasure Channel

A channel similar to the binary symmetric channel is the one described below. The binary erasure channel (BEC) is also a discrete memoryless channel, where input values are considered within the binary alphabet $\mathcal{B} = \{0, 1\}$, but the output alphabet includes an additional element, the *erasure* (see Figure 2.3). When an error occurs in this channel the transmitted bit value is lost. It is depicted in Figure 2.3 with a transition from the initial value to the erasure. The channel is then characterized by the erasure probability, α, and it is commonly denoted by BEC(α).

The capacity of this channel is given by [40]:

$$
\begin{aligned}
C_{\mathrm{BEC}(\alpha)} &= \max_{p(x)} I(X;Y) && (2.16)\\
&= \max_{p(x)} \left(H(X) - H(X|Y)\right) && (2.17)\\
&= \max_{p(x)} (H(X) - \underbrace{H(X|Y=e)}_{H(X)} \underbrace{\Pr(Y=e)}_{\alpha}) && (2.18)\\
&= \max_{p(x)} \left(H(X) - H(X)\alpha\right) && (2.19)\\
&= \max_{p(x)} \left(H(X)(1-\alpha)\right) && (2.20)\\
&= 1-\alpha && (2.21)
\end{aligned}
$$

2.4 Coding Theory

2.4.1 Linear Codes

Let $\mathbb{F}_q$ be a finite field with q elements, sometimes referred as the Galois field $\mathrm{GF}(q)$, and let $\mathbb{F}_q^n$ be the vector space containing all n-length vectors. We define a linear code as follows.

Definition 6. *A linear code of length n and dimension k over $\mathbb{F}_q$, denoted by $\mathcal{C}(n,k)$, is a linear subspace of $\mathbb{F}_q^n$, $\mathcal{C} \subseteq \mathbb{F}_q^n$.*

In other words, we say that a non empty $\mathcal{C}(n,k)$ is a linear code if:

1. For every pair of n-length vectors $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ we have that $\mathbf{x} + \mathbf{y} \in \mathcal{C}$.
2. For every $\alpha \in \mathbb{F}_q$ and $\mathbf{x} \in \mathcal{C}$ we also have that $\alpha \cdot \mathbf{x} \in \mathcal{C}$.
3. And $\mathbf{0} \in \mathcal{C}$.

Note that, for convenience, we omit the length and dimension of the code, and we denote it by $\mathcal{C}$ instead of $\mathcal{C}(n,k)$, when there is no potential for ambiguity.

Definition 7. *Every n-length vector $\mathbf{x}$ in a code, $\mathbf{x} \in \mathcal{C}$, is a codeword.*

Therefore, the set composed of all codewords makes up a code.

Generator Matrix

A linear code $\mathcal{C}(n,k)$ can be described by a $k \times n$ matrix G, called *generator matrix*. Rows in the matrix G form a basis for the subspace $\mathcal{C}$, such that every n-length vector of the code $\mathbf{x}$ is calculated as $\mathbf{x} = \mathbf{u}G$, for all $\mathbf{u} \in \mathbb{F}_q^k$. And thus:

$$\mathcal{C} = \{\mathbf{x} : \mathbf{x} = \mathbf{u}G, \forall \mathbf{u} \in \mathbb{F}_q^k\} \tag{2.22}$$

Parity-Check Matrix

A linear code $\mathcal{C}(n,k)$ is equivalently described by a $(n-k) \times n$ matrix H, called *parity-check matrix*. Each row in the matrix H represents a n-length parity-check equation that has to be satisfied by every codeword $\mathbf{x}$, such that:

$$\mathcal{C} = \{\mathbf{x} : H\mathbf{x}^t = \mathbf{0}\} \tag{2.23}$$

In other words, the set of codewords composing the code $\mathbf{x} = (x_1, x_2, ..., x_n)$ consist of all solutions for m independent parity-check equations:

$$\sum_{k=1}^{n} H_{j,k} \cdot x_k = 0\,, \quad 1 \leq j \leq m \tag{2.24}$$

where $m = n - k$.

A parity-check matrix H corresponding to a generator matrix G can be obtained by:

$$GH^t = \mathbf{0} \tag{2.25}$$

If H has full row rank, then the information rate of the code is k/n.

Binary Linear Codes

When working with binary linear codes we commonly use two measures to judge some code properties. Let $\mathcal{C}(n,k)$ be a binary linear code. We define the Hamming

weight and distance as follow.

Definition 8. *Let* $\mathbf{x}$ *be a codeword,* $\mathbf{x} \in \mathcal{C}$*. The Hamming weight of* $\mathbf{x}$*, denoted by* $w(\mathbf{x})$*, is defined as the number of ones in* $\mathbf{x}$*.*

Definition 9. *Let* $\mathbf{x}$ *and* $\mathbf{y}$ *be two binary words. The Hamming distance between* $\mathbf{x}$ *and* $\mathbf{y}$*, denoted by* $d_{\mathrm{H}}(\mathbf{x}, \mathbf{y})$*, is defined as number of bits where both words differ, thus* $d_{\mathrm{H}}(\mathbf{x}, \mathbf{y}) = w(\mathbf{x} \oplus \mathbf{y})$*.*

Cosets

Let $\mathcal{C} \subset \mathbb{F}_q^n$ be a linear code.

Definition 10. *A coset of* $\mathcal{C}$ *is the subset of* $\mathbb{F}_q^n$ *obtained by adding any n-length vector* $\mathbf{y} \in \mathbb{F}_q^n$ *to every codeword in* $\mathcal{C}$*, and it is denoted by* $\mathbf{y} + \mathcal{C}$*.*

$$\mathbf{y} + \mathcal{C} = \{\mathbf{x} + \mathbf{y} : \forall \mathbf{x} \in \mathcal{C}\} \tag{2.26}$$

It should be noted that from every n-length vector in a coset we obtain the same coset, i.e. given $\mathbf{y} \in \mathbf{x} + \mathcal{C}$ we have that $\mathbf{y} + \mathcal{C} = \mathbf{x} + \mathcal{C}$. Since $\mathbf{y} \in \mathbf{x} + \mathcal{C}$ there exist some $\mathbf{z} \in \mathcal{C}$ such that $\mathbf{y} = \mathbf{x} + \mathbf{z}$, and thus we can construct $\mathbf{y} + \mathcal{C}$ as follows:

$$\begin{aligned} \mathbf{y} + \mathcal{C} &= \{\mathbf{y} + \mathbf{u} : \forall \mathbf{u} \in \mathcal{C}\} && (2.27) \\ &= \{\mathbf{x} + \mathbf{z} + \mathbf{u} : \forall \mathbf{u} \in \mathcal{C}\} && (2.28) \\ &= \{\mathbf{x} + \mathbf{v} : \forall \mathbf{v} \in \mathcal{C}\} = \mathbf{x} + \mathcal{C} && (2.29) \end{aligned}$$

The lowest weight vector in a coset is called *coset leader*. Given the above property, we can use then this coset leader to index every coset. However, a coset leader need not be unique, and therefore we can use more that one leader to index a coset.

Let H be the parity-check matrix of a code $\mathcal{C}$.

Definition 11. *The syndrome of any word* $\mathbf{x}$*, denoted by* $s(\mathbf{x})$*, is defined as* $s(\mathbf{x}) = H\mathbf{x}^t$*.*

Other interesting property of cosets is that the syndrome of two words $\mathbf{x}$ and $\mathbf{y}$ coincides, i.e. $s(\mathbf{x}) = s(\mathbf{y})$, if and only if both words belong to the same coset.

2.4.2 Bipartite and Tanner Graphs

A parity-check matrix, and thus a linear code, can be equivalently represented by a bipartite graph, also called Tanner graph [44]. Let $G(\mathcal{V}; \mathcal{E})$ be an undirected graph with vertices and edges belonging to the sets $\mathcal{V}$ and $\mathcal{E}$, respectively. The graph G is said to be bipartite if the set $\mathcal{V}$ is composed by two disjoint subsets of vertices. When referring to a code, we call both *symbol* and *check* nodes[1], although sometimes they are also referred as variable and parity nodes, respectively. The set of symbol nodes is typically denoted by $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ and the set of check nodes by $\mathcal{P} = \{c_1, c_2, ..., c_m\}$, such that $\mathcal{V} = \mathcal{S} \cup \mathcal{P}$ and $\mathcal{S} \cap \mathcal{P} = \emptyset$. In a Tanner graph, every symbol node s_i and check node c_j corresponds to the i-th column and j-th row in the equivalent parity-check matrix, respectively. Therefore, in a binary linear code a symbol node s_i is connected by an edge to a check node c_j if the entry $H_{j,i} = 1$.

An example is depicted in Figure 2.4. The figure shows the Tanner graph corresponding to a code defined by the parity-check matrix H described in Eq. (2.30). Symbol nodes are commonly depicted by circles, and check nodes by boxes.

$$H = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{2.30}$$

The number of edges incident to a symbol node s_i is called *symbol node degree* and denoted by $d(s_i)$. Similarly, the number of edges incident to a check node c_j is called *check node degree* and denoted by $d(c_j)$. Symbol and check node degrees, $d(s_i)$ and

[1]Henceforth, we use the term node instead of vertex when referring to a code graph.

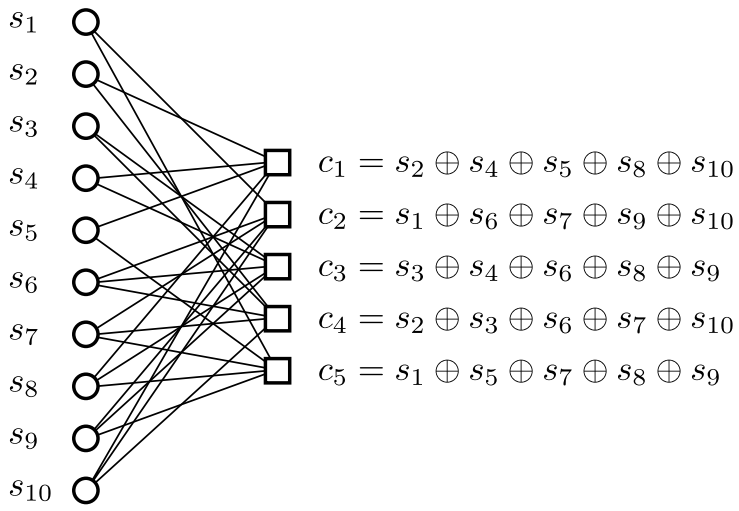

Figure 2.4: Bipartite or Tanner graph, and parity-check equations.

$d(c_j)$, correspond to the number of ones in the i-th column and j-th row of the parity-check matrix, respectively. In Figure 2.4 every check node has a constant degree, $d(c_j) = 5$ for all $1 \leq j \leq 5$, but different symbol node degrees, $d(s_i) = 2$ for $1 \leq i \leq 5$ and $d(s_i) = 3$ for $5 < i \leq 10$.

Adjacent Nodes

Two nodes are said to be *adjacent* if both are directly connected by an edge. Note that, in a Tanner graph as defined above, a symbol node is adjacent to one or more check nodes, but never to other symbol nodes. Similarly, a check node is adjacent to one or more symbol nodes, but never to other check nodes.

In the graph literature, the term *neighbor* is usually referred to a set of nodes adjacent to other node. Given a $m \times n$ parity-check matrix H. Let $\mathcal{N}(j)$ denote the set of symbol nodes adjacent to the check node c_j, this is given by $\mathcal{N}(j) = \{i : H_{j,i} = 1, 1 \leq i \leq n\}$, and let $\mathcal{M}(i)$ be the corresponding set of check nodes adjacent to the symbol node s_i, $\mathcal{M}(i) = \{j : H_{j,i} = 1, 1 \leq j \leq m\}$.

In this work, we extend the concept of neighborhood for two symbol nodes as follows. Two symbol nodes, s_i and s_k, are said to be *next neighbors* if both are directly

connected through a common check node, i.e. $\mathcal{M}(i) \cap \mathcal{M}(k) \neq \emptyset$, and thus there exist a 2-length path consisting of two edges that connect both symbols. Let $\mathcal{N}^2(k)$ denote the next neighboring set of a symbol node s_k, it is given by:

$$\mathcal{N}^2(k) = \{i : i \in \mathcal{N}(j), \forall j \in \mathcal{M}(k)\} \tag{2.31}$$

Cycles

In a graph, a *path* is a sequence of adjacent nodes. In a Tanner graph, this is an alternating sequence of symbol and check nodes. In this context, it is assumed that every path is simple, i.e. there are no repeated nodes in the path.

Definition 12. *In a graph, a path that starts and ends at the same node is called closed path or cycle.*

It should be noted that any finite length graph has necessarily cycles. An n-length cycle is a cycle with n vertices (or n edges). Cycles in a Tanner graph have always an even length.

Definition 13. *The girth of a graph is the length of a shortest cycle in the graph.*

Subgraphs and Local Graphs

Let $G(\mathcal{V}; \mathcal{E})$ be a graph, the subgraph $G'(\mathcal{V}'; \mathcal{E}')$, such that $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$, is an *induced subgraph* if for every edge $e \in \mathcal{E}$ connecting two nodes $v_i, v_j \in \mathcal{V}$ we have that $e \in \mathcal{E}'$ if and only if $v_i, v_j \in \mathcal{V}'$.

Starting from a node, we can expand the induced subgraph called *local graph* first adding those nodes and edges with the shortest path to the initial node.

Assuming that there are no cycles in a local graph, the *depth* of this local graph is the length of the longest path.

Part II

Information Reconciliation with Low-Density Parity-Check Codes

Chapter 3

Information Reconciliation and Low-Density Parity-Check Codes

This chapter is organized as follows. In Section 3.1 we introduce the problem of information reconciliation and its equivalence to the source coding with side information problem proposed by Slepian and Wolf. Afterward, we discuss the relation discovered by Wyner between channel coding and Slepian-Wolf coding. Next, in Section 3.2 we introduce low-density parity-check (LDPC) codes and the most common techniques used for their decoding. We emphasize on the iterative message-passing decoding based on belief propagation. Finally, in Section 3.3 we propose a new algorithm to construct good irregular LDPC codes that belong to an ensemble of codes.

3.1 Introduction to Information Reconciliation

In this section we consider the problem of information reconciliation from an information theoretic perspective and study how common coding techniques can be used for this problem.

Information reconciliation refers to any method used to ensure that two parties agree on a common string provided they have access to two correlated sequences

x and **y** [42]. During reconciliation two parties exchange a set of messages M over a noiseless channel, such that at the end of the process they agree on some string function of their sequences and the exchanged messages. In our case, the correlated sequences are obtained by Alice and Bob after the quantum phase of a QKD protocol. It does not matter whether an actual quantum channel has been used to transmit qubits from Alice to Bob, as in a standard prepare and measure protocol or an entangled pairs emitter acts as the source of correlations. In both cases, assuming that errors and attacks are independent identically distributed, X and Y can be regarded as correlated random variables and every symbol in Y can be seen as given by transition probability $p_W(y|x)$, or equivalently as if every symbol were the output of a memoryless channel W.

Typically, channels are classified in families characterized by some continuous variable, ξ, selected to parameterize its behavior. The variable ξ is chosen such that increasing values of ξ imply a degraded version of the channel [31]. For example, the family of binary symmetric channels is parameterized by the crossover probability and the family of additive white Gaussian noise channels by the signal-to-noise ratio. A channel $W_{\tilde{\xi}'}$ is a degraded, or noisier, version of the channel $W_{\tilde{\xi}}$ if:

$$p_{W_{\tilde{\xi}'}}(y'|x) = p_Q(y'|y)\, p_{W_{\tilde{\xi}}}(y|x) \tag{3.1}$$

where Q is some auxiliary channel.

Let the information rate be the proportion of non redundant symbols sent through a channel. A linear code $\mathcal{C}(n,k)$ transforms an string of k symbols in a codeword $\mathbf{x}$, $\mathbf{x} \in \mathcal{C}$, of n symbols with k independent symbols and $n-k$ redundant symbols, and in consequence it achieves an information rate, R, of k/n.

This parameterization allows to study two related concepts: the capacity of a channel, that is, the maximum information rate that can be transmitted for a fixed ξ and, for a specific error correcting code $\mathcal{C}$, the maximum value $\xi_{\max}$, i.e. the noisiest channel for which a sender can reliably transmit information with $\mathcal{C}$. The relation

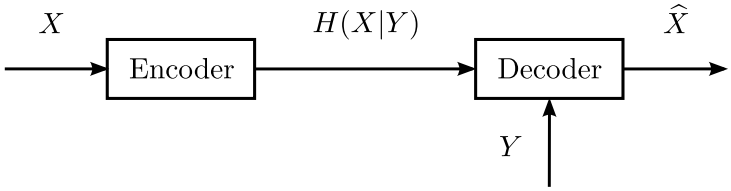

Figure 3.1: Asymmetric Slepian-Wolf coding scheme: source coding with side information at the decoder.

between both answers gives an idea of the efficiency of the code, or in other words, how close is the coding rate of a code to the optimal value.

We can measure, analogously, the efficiency f of an information reconciliation protocol as the additional information required for compressing a source, i.e. the relation between the length of the messages exchanged to reconcile the strings, $|M|$, and the theoretical minimum message length.

3.1.1 Slepian-Wolf Coding and Source Coding with Side Information at the Decoder

The problem of information reconciliation in secret key agreement is formally equivalent to the problem of source coding with side information at the decoder studied by Slepian and Wolf in 1973 [45], also known as asymmetric Slepian-Wolf coding. In their original work, the authors analyze the problem of correlated sources, and they determine $R \geq H(X,Y)$ the minimum information required to encode two correlated information sequences, represented by two discrete random variables X and Y respectively, even if both sequences are encoded separately.

This contribution can be also used to determine how should X (the source) be encoded in order to allow a decoder with access to side information Y to recover the information in X with high probability. The minimum message length is given by the conditional entropy $H(X|Y)$, and thus given the decoder access to side information Y no encoding of X shorter than $H(X|Y)$ allows for reliable decoding. This asymmetric

Slepian-Wolf coding scheme is depicted in Figure 3.1.

Using this approach, the efficiency of an information reconciliation procedure is given by:

$$f = \frac{|M|}{H(X|Y)} \geq 1 \tag{3.2}$$

where $f = 1$ stands, then, for perfect reconciliation.

3.1.2 Syndrome Source Coding

The relation between Slepian-Wolf coding and channel coding was pointed out by Aaron D. Wyner a year later in Ref. [46]. Based on Wyner's idea powerful error correcting techniques can be then used for the information reconciliation problem. For instance, LDPC codes have been widely considered in the literature for the coding problem of correlated sources [47, 48].

In channel coding, let $\mathcal{C}(n,k)$ be a linear code and let H be its corresponding parity-check matrix. The code $\mathcal{C}$ maps each k-length word into a n-length codeword $\mathbf{x} \in \mathcal{C}$, such that $H\mathbf{x}^t = \mathbf{0}$. A codeword is then transmitted through a communication channel which introduces errors in the received word $\mathbf{y}$. Errors can be denoted by other n-length vector $\mathbf{e}$, such that $\mathbf{y} = \mathbf{x} \oplus \mathbf{e}$. A decoding algorithm tries to find the correct $\mathbf{x}$ using the syndrome of $\mathbf{y}$, $\mathbf{z} = H\mathbf{y}^t$. This syndrome only depends on the error vector, since:

$$\mathbf{z} = H\mathbf{y}^t = H(\mathbf{x}^t + \mathbf{e}^t) = H\mathbf{x}^t + H\mathbf{e}^t = H\mathbf{e}^t \tag{3.3}$$

It is assumed that there exists a decoding function $g(\cdot)$ that attempts to find the error vector $\mathbf{e}$ with the least weight, such that $\mathbf{e}$ can be decoded from its syndrome $H\mathbf{e}^t$ with high probability, a technique called *syndrome decoding* (see decoding techniques in Section 3.2.2).

As introduced in Section 2.4.1, two vectors yield the same syndrome when they are elements of the same coset, and therefore syndromes and cosets are closely related.

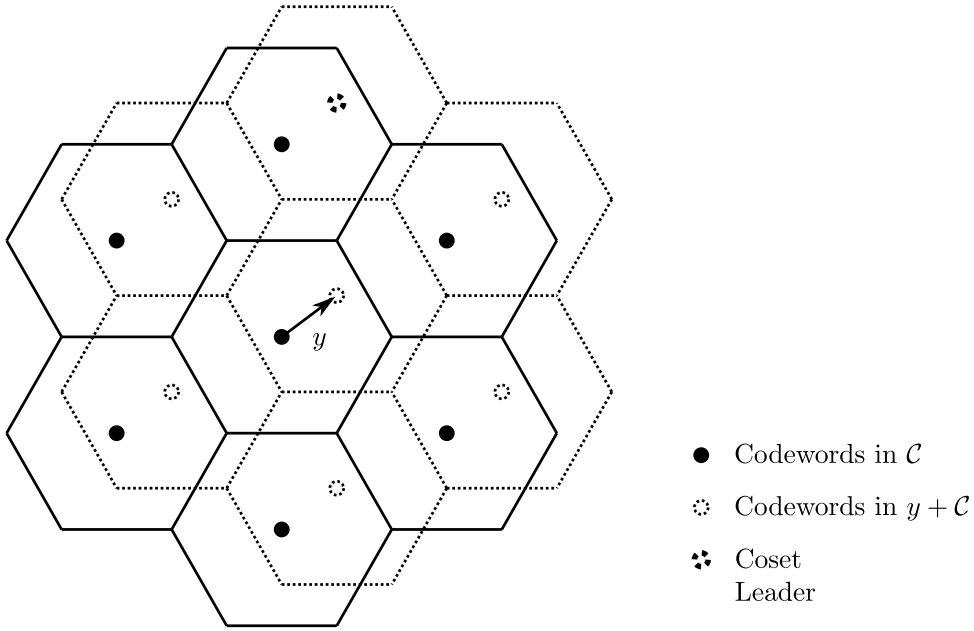

Figure 3.2: Coset and coset leader.

The coset leader is the lowest weight vector in a coset, and thus this is the error vector decoded. An example of cosets and coset leader is shown in Figure 3.2. A similar depiction was already proposed in Ref. [49]. Codewords in the original code $\mathcal{C}$ are depicted in the figure with a solid point, while codewords in the coset $y + \mathcal{C}$ are depicted by a dotted circle.

This decoding technique can be also applied to the asymmetric Slepian-Wolf coding problem as follows. In information reconciliation, Y is a noisy version of X (or viceversa), and both encoder and decoder have access to a noiseless channel. The syndrome of $\mathbf{x}$, an instance of X, $\mathbf{z} = H\mathbf{x}^t$, with length per symbol $1 - R$, indicates in which of the cosets of $\mathcal{C}$ is $\mathbf{x}$ a codeword. The decoder tries to recover $\mathbf{x}$ from $\mathbf{z}$ given the side information $\mathbf{y}$, an instance of Y, ant it can be done since:

$$\mathbf{z} \oplus H\mathbf{y}^t = H(\mathbf{x}^t \oplus \mathbf{y}^t) = H\mathbf{e}^t \tag{3.4}$$

Let R_s be the rate of information disclosed by the source, and let R_c be the coding rate of a channel code $\mathcal{C}(n,k)$ (e.g. an LDPC code) used for reconciliation. Both

information rates are related by $m = n - k$, the redundancy of $\mathcal{C}$, equivalent to the redundancy disclosed by the source. And thus:

$$R_s = \frac{m}{n} = 1 - \frac{k}{n} = 1 - R_c \tag{3.5}$$

The efficiency of an information reconciliation method based in syndrome source coding is then given by:

$$f_{\mathcal{C}} = \frac{1 - R_c}{H(X|Y)} \tag{3.6}$$

In the special, but also important, case of the binary symmetric channel with crossover probability ϵ, the efficiency can be written as:

$$f_{\mathrm{BSC}(\epsilon)} = \frac{1 - R_c}{h(\epsilon)} \tag{3.7}$$

where $h(\epsilon)$ is the binary Shannon entropy as defined in Eq. (2.2).

3.2 Low-Density Parity-Check Codes

Low-density parity-check (LDPC) codes were introduced by Robert G. Gallager in the early 1960s [29,30], however this work was generally neglected for years —with very few exceptions, as in Ref. [44]—. These codes were rediscovered thirty years later by MacKay and Neal in Refs. [50,51], where the authors show that the performance of regular LDPC codes is almost as close to the Shannon limit as the Turbo codes one. But it was later when these codes become of interest since it was demonstrated that these codes are capacity achieving for some communication channels [31], and it was also introduced different techniques for designing good ensembles of irregular LDPC codes [52–54].

An LDPC code is a linear code defined in terms of a very *sparse* (low-density) parity-check matrix H, i.e. there are few non zero entries in the parity-check matrix. This parity-check matrix is typically depicted by a bipartite graph, or Tanner graph

(see Figure 2.4). Nodes in the graph are then divided between symbol and check nodes, and denoted by s_i and c_j, respectively.

An LDPC code is said to be (j,k)-regular, or *regular*, when each symbol and check node have a constant number of incident edges (degree), j and k respectively, i.e. $d(s_i) = d(s_j)$ for all $i \neq j$ and $d(c_k) = d(c_l)$ for all $k \neq l$. Therefore, in the parity-check matrix there is a constant number of ones in each row and column. Otherwise, the code is said to be *irregular* when there are symbols or check nodes with different degrees. It should be noted that in this work we only use irregular codes since they improve the performance of regular ones [55].

An ensemble of irregular LDPC codes —i.e. a family of codes— are usually defined by two generating polynomials, $\lambda(x)$ and $\rho(x)$. The coefficients of each term, λ_i and ρ_j, define the fraction of edges in the code graph connected to a i-degree symbol node and j-degree check node, respectively. The degree of each term, i and j, define the number of incident edges to a node, symbol and check node, respectively. These polynomials are defined as follow:

$$\lambda(x) = \sum_{i=2}^{d_s^{\max}} \lambda_i x^{i-1} \; ; \quad \rho(x) = \sum_{j=2}^{d_c^{\max}} \rho_j x^{j-1} \tag{3.8}$$

where $d_s^{\max}$ and $d_c^{\max}$ are the highest degree for symbol and check nodes, respectively. Note that the sum of these coefficients must be one, thus $\sum_i \lambda_i = 1$ and $\sum_j \rho_j = 1$.

The information rate, as defined in Eq. (2.9), of a family of LDPC codes can be then calculated from the edge distribution provided by both generating polynomials as follow:

$$R = \frac{n-m}{n} = 1 - \frac{m}{n} = 1 - \frac{\sum_{j=2}^{d_c^{\max}} \rho_j / j}{\sum_{i=2}^{d_s^{\max}} \lambda_i / i} \tag{3.9}$$

The fraction of edges connected to a symbol or check node can be also translated to the probability of finding an i-degree symbol node or a j-degree check node in the graph, λ_i^* and ρ_j^*, respectively.

$$\lambda_i^* = \frac{\lambda_i / i}{\sum_{i=2}^{d_s^{\max}} \lambda_i / i} \; ; \quad \rho_j^* = \frac{\rho_j / j}{\sum_{j=2}^{d_c^{\max}} \rho_j / j} \tag{3.10}$$

In everything discussed here, it is assumed that we work with binary LDPC codes, even when we omit the term binary for convenience. However, non-binary LDPC codes can also be used in this context, and they were already proposed for information reconciliation in QKD [56]. A brief introduction to the problem of information reconciliation in QKD can be found in Ref. [57].

3.2.1 Design of Irregular LDPC Codes

The asymptotic behavior of a family of LDPC codes can be analyzed using, for instance, a *density evolution* algorithm [31]. Two versions of this algorithm, Gaussian approximation and discretized density evolution, were originally applied to design good families of LDPC codes in Refs. [53] and [54], respectively. Differential evolution and density evolution can be used together to find good edge distributions [58].

The problem of finding families of good LDPC codes for the binary symmetric channel is beyond the scope of this work. It was already discussed by Elkouss *et al.* in Ref. [26] and it is also within the scope of his Ph.D. dissertation [37].

3.2.2 Efficient Decoding Techniques for LDPC Codes

Let $\mathcal{C}(n,k)$ be a linear code to be used in a data communication, and H its corresponding parity-check matrix. Let $\mathbf{x}$ be the transmitted codeword, and $\mathbf{y}$ the received word. The goal of any decoder is to find the most likely codeword $\hat{\mathbf{x}}$ that was sent, such that $\hat{\mathbf{x}} = \mathbf{x}$ with high probability. On the binary symmetric channel, this problem is equivalent to finding the minimum-weight vector $\mathbf{e}$ that satisfies $H(\mathbf{y}^t \oplus \mathbf{e}^t) = \mathbf{0}$, such that $\hat{\mathbf{x}} = \mathbf{y} \oplus \mathbf{e}$, i.e. $\hat{\mathbf{x}}$ is the codeword with the smallest Hamming distance to $\mathbf{y}$. A decoder that maximizes the probability that $\hat{\mathbf{x}}$ was sent given that $\mathbf{y}$ was received is called *maximum-likelihood* decoder, and it is known that maximum-likelihood de-

coding for the BSC is an NP-complete problem when using a general code where the decoder has to consider all the 2^k possible vectors [59].

Fortunately, there are some decoding algorithms that perform quite well with sparse graphs, such as LDPC codes. They are the *message-passing* algorithms (MPA) already introduced by Gallager [29,30]. These algorithms are so called MPA because they operate by exchanging messages, iteratively, along the edges of a bipartite graph, i.e. messages are exchanged between symbol and check nodes. A family of well-analyzed iterative message-passing decoders are those based on *belief propagation*. In a belief propagation based algorithm, the messages passed along the edges are probabilities, or beliefs. An LDPC code is efficiently decoded by using the *sum-product* algorithm, the better known implementation of belief propagation.

Belief propagation is in general less powerful than maximum-likelihood decoding under general assumptions. However, belief propagation performs equally under certain assumptions, such as: (1) there are no cycles in the graph, and (2) only extrinsic information is passed along the paths —i.e. a message sent along an edge may not depend on the message previously received along this edge—. As it is shown below, both assumptions are taken into account when constructing an LDPC code. Cycles, for instance, cannot be avoided in a finite graph, but we can make these as large as possible. Furthermore, the graph is constructed avoiding special topologies, such as stopping and trapping sets [60,61], thus maximizing the exchange of extrinsic information.

Other well-known decoding algorithms, such as mim-sum and bit flipping based algorithms, are not considered in this thesis even though they may be much more interesting for an efficient hardware implementation of LDPC codes.

Sum-Product Algorithm

A comprehensive description of the sum-product algorithm is provided by MacKay in Ref. [41]. The algorithm consists of three phases, an initialization phase and two other

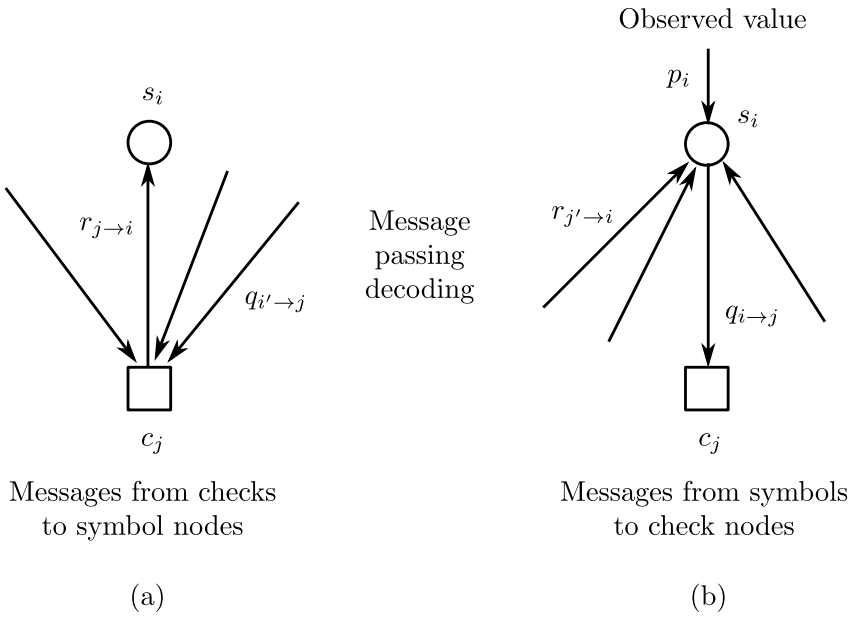

Figure 3.3: Observed values and messages exchanged between symbol and check nodes in the sum-product algorithm.

phases that run iteratively to compute messages from check nodes to symbol nodes (horizontal step), and messages from symbol nodes to check nodes (vertical step). Initialization from the observed values, and messages computed in both horizontal and vertical steps are depicted in Figure 3.3.

Initialization.— Let p_i^0 be the a priori probability that the i-th symbol s_i is 0, $p_i^0 = \Pr(s_i = 0)$, and let p_i^1 be the a priori probability that the i-th symbol s_i is 1, $p_i^1 = \Pr(s_i = 1) = 1 - p_i^0$. Assuming a communication over the BSC with crossover probability ϵ, if for instance the i-th symbol at the receiver is $s_i = 0$, p_i^0 and p_i^1 are initialized to $1 - \epsilon$ and ϵ respectively.

For every edge in the graph (i, j), such that $H_{j,i} = 1$, the algorithm also initializes the first message from symbols to check nodes as follows:

$$q_{i\to j}^0 = p_i^0 \tag{3.11}$$

$$q_{i\to j}^1 = p_i^1 \tag{3.12}$$

Horizontal step.— In this step the algorithm run through the check nodes and it computes the corresponding messages $r_{j\to i}$ from check to symbol nodes. For each j-th check node it computes:

$$r^0_{j\to i} = \sum_{s_{i'}:i'\in\mathcal{N}(j)\backslash\{i\}} \Pr(c_j|s_i=0, s_{i'}: i'\in\mathcal{N}(j)\backslash\{i\}) \prod_{i'\in\mathcal{N}(j)\backslash\{i\}} q^{s_{i'}}_{i\to j} \quad (3.13)$$

$$r^1_{j\to i} = \sum_{s_{i'}:i'\in\mathcal{N}(j)\backslash\{i\}} \Pr(c_j|s_i=1, s_{i'}: i'\in\mathcal{N}(j)\backslash\{i\}) \prod_{i'\in\mathcal{N}(j)\backslash\{i\}} q^{s_{i'}}_{i\to j} \quad (3.14)$$

where the conditional probability in the sum, $\Pr(c_j|s_i=0, s_{i'}: i'\in\mathcal{N}(j)\backslash\{i\})$ and $\Pr(c_j|s_i=1, s_{i'}: i'\in\mathcal{N}(j)\backslash\{i\})$, is either zero or one, depending on whether the observed check node c_j verifies the corresponding parity-check equation.

Vertical step.— In this step the algorithm run through the symbol nodes and it computes the corresponding messages $q_{i\to j}$ from symbol to check nodes. For each i-th symbol node it computes the corresponding $d(s_i)$ messages:

$$q^0_{i\to j} = \alpha_{j,i}\, p^0_i \prod_{j'\in\mathcal{M}(i)\backslash\{j\}} r^0_{j'\to i} \quad (3.15)$$

$$q^1_{i\to j} = \alpha_{j,i}\, p^1_i \prod_{j'\in\mathcal{M}(i)\backslash\{j\}} r^1_{j'\to i} \quad (3.16)$$

where $\alpha_{j,i}$ is chosen such that $q^0_{j\to i} + q^1_{j\to i} = 1$.

Finally, the posterior probabilities for each symbol are calculated as follow:

$$q^0_i = \alpha_i\, p^0_i \prod_{j\in\mathcal{M}(i)} r^0_{j\to i} \quad (3.17)$$

$$q^1_i = \alpha_i\, p^1_i \prod_{j\in\mathcal{M}(i)} r^1_{j\to i} \quad (3.18)$$

where α_i is chosen such that $q^0_i + q^1_i = 1$. At this point, each symbol value s_i is then updated such that $s_i = 0$ if $q^0_i > 1/2$, otherwise $s_i = 1$.

The algorithm stops if every check node is satisfied, and thus $H\mathbf{s}^t = \mathbf{0}$. The algorithm aborts if it reaches a maximum number of iterations. Otherwise, the algorithm continues going back to the horizontal step.

Conventionally, messages between symbol and check nodes are computed according to the order described above. Thus, at each iteration, first, all symbol nodes compute and send messages to their neighbors, and secondly it is performed the opposite process, in which all check nodes send back their corresponding messages to the symbol nodes. This conventional scheme is called *flooding schedule*.

It was demonstrated that this schedule is not efficient and can be outperformed by other approaches [62,63]. A simple alternative is the *serial schedule*, in which extrinsic information tends to spread twice as fast. In this new schedule, only check nodes are processed sequentially, and at each check node both messages from symbols to check nodes and messages from checks to symbol nodes are computed and sent along its incident edges.

Sum-Product Algorithm (Flooding Schedule)

The previous description of the sum-product algorithm can be efficiently implemented using likelihoods, or even log-likelihoods, instead of probabilities. Then, given a binary variable x, it can be represented by a single value using the log-likelihood ratio:

$$L(x) = \log \frac{\Pr(x=0)}{\Pr(x=1)} \tag{3.19}$$

where log is the logarithm to base e.

To translate from log-likelihood ratios back to probabilities we use:

$$\begin{aligned} \Pr(x=1) &= \frac{\Pr(x=1)}{\Pr(x=0)+\Pr(x=1)} \\ &= \frac{\Pr(x=1)/\Pr(x=0)}{1+\Pr(x=1)/\Pr(x=0)} = \frac{e^{-L(x)}}{1+e^{-L(x)}} \end{aligned} \tag{3.20}$$

and

$$\begin{aligned} \Pr(x=0) &= \frac{\Pr(x=0)}{\Pr(x=0)+\Pr(x=1)} \\ &= \frac{\Pr(x=0)/\Pr(x=1)}{1+\Pr(x=0)/\Pr(x=1)} = \frac{e^{L(x)}}{1+e^{L(x)}} \end{aligned} \tag{3.21}$$

Another benefit of working with logarithms is that the product of probabilities, as defined in Eqs. (3.13) to (3.18), results in a sum of log-likelihood ratios.

Let ℓ be the iteration number, such that $q_{i\to j}^{(\ell)}$ stands for the message send from the i-th symbol to the j-th check node in the ℓ-th iteration, and $r_{j\to i}^{(\ell)}$ stands for the message send from the j-th check to the i-th symbol node in the ℓ-th iteration. Let $p_i = L(s_i)$ be the log-likelihood ration of the i-th observed value s_i. The sum-product algorithm can be rewritten as follows.

$$q_{i\to j}^{(\ell)} = \begin{cases} p_i, & \text{if } \ell = 0 \\ p_i + \sum\limits_{j'\in\mathcal{M}(i)\setminus\{j\}} r_{j'\to i}^{(\ell)}, & \text{if } \ell > 0 \end{cases} \tag{3.22}$$

$$r_{j\to i}^{(\ell)} = \gamma^{-1}\left(\sum_{i'\in\mathcal{N}(j)\setminus\{i\}} \gamma\left(q_{i'\to j}^{(\ell-1)}\right)\right) \tag{3.23}$$

where $\gamma(\cdot)$ and $\gamma^{-1}(\cdot)$ are defined as follow [52]:

$$\gamma(x) = \left(\operatorname{sign}(x), -\log\tanh\frac{|x|}{2}\right) \tag{3.24}$$

$$\gamma^{-1}(\operatorname{sign}, x) = (-1)^{\operatorname{sign}} \cdot -\log\tanh\frac{x}{2} \tag{3.25}$$

and

$$\operatorname{sign}(x) = \begin{cases} 0, & x > 0 \\ 1, & x < 0 \\ \zeta, & x = 0 \end{cases} \tag{3.26}$$

where ζ is 0 or 1 with probability 1/2.

A review of the state of the art in the design of LDPC decoders can be found in Ref. [64]. The authors analyze the challenges in the hardware implementation of several decoding architectures (with emphasis on parallel architectures) for different ensembles of LDPC codes. Different performance metrics are also discussed in relation to the design of digital integrated circuits.

Sum-Product Algorithm (Syndrome Source Coding)

Note that in the channel coding setting, the belief propagation algorithm is designed to output a codeword with syndrome $\mathbf{0}$, whereas in the source coding setting, this algorithm needs to be modified so that it outputs a vector satisfying a given syndrome [65].

$$q_{i\to j}^{(\ell)} = \begin{cases} p_i, & \text{if } \ell = 0 \\ p_i + \sum\limits_{j' \in \mathcal{M}(i)\setminus\{j\}} r_{j'\to i}^{(\ell)}, & \text{if } \ell > 0 \end{cases} \tag{3.27}$$

$$r_{j\to i}^{(\ell)} = \underbrace{(-1)^{c_j}}_{syndrome} \gamma^{-1}\left(\sum_{i' \in \mathcal{N}(j)\setminus\{i\}} \gamma\left(q_{i'\to j}^{(\ell-1)}\right)\right) \tag{3.28}$$

where $(c_1, c_2, ..., c_m)$ is the syndrome transmitted by the source, and c_j is the value of this syndrome corresponding to the parity-check equation defined by the j-th check node.

3.3 Constructing Low-Density Parity-Check Codes

Belief propagation based algorithms provide optimum decoding over cycle-free Tanner graphs [31]. However, any finite length graph has necessarily cycles. It was shown that it is important to make those cycles as large as possible. It has been also shown that a large girth improves the performance of LDPC codes using iterative decoding

as it enforces a reasonable minimum distance [66]. However, note that a large girth does not automatically imply a a large minimum distance.

The progressive edge-growth (PEG) algorithm is an efficient method for constructing Tanner graphs with large girth [66,67], in most cases with better performance than randomly constructed codes. PEG algorithm's interest lies in its simplicity, and its flexibility when constructing irregular codes from a complex symbol node degree distribution.

The performance of these codes can be also improved, for instance, analyzing each additional cycle. Recent works have been focused on the study of those cycles, since it has been shown that it is possible to improve the performance of LDPC codes, for instance avoiding small stopping sets and trapping sets (near codewords) [60, 61]. New definitions have also been introduced, such as the extrinsic message degree (EMD) or the approximate cycle EMD (ACE), which are two common measures used to calculate the connectivity of symbol nodes [68–70].

3.3.1 Progressive Edge-Growth Algorithm

A PEG-based algorithm consists of two basic procedures: a local graph expansion and a check node selection procedure. Both procedures are executed sequentially in order to construct a Tanner graph connecting symbol and check nodes in an edge-by-edge manner. In the first procedure it is performed the expansion of the local graph from a symbol node, this expansion is used to detect and avoid short cycles when adding a new edge to the graph. The result is that check nodes that will produce a cycle are pruned, or if it is not possible to avoid a cycle, there only remains a set of candidate check nodes producing the largest cycle. The selection procedure is used to reduce this list of candidate nodes according to the current graph setting. In typical PEG-based algorithms, this procedure attempts to balance the degree of any check node selecting those candidates with the lowest check node degree.

In the original PEG algorithm [66,67], a code is constructed according to a symbol

node degree sequence. This sequence is previously calculated with the number of symbol nodes n and the edge degree distribution established by $\lambda(x)$ (see Eq. (3.8) and Eq. (3.10)). Note that the original proposal does not take into account the second polynomial, $\rho(x)$, for the check node degree distribution. The algorithm proposed in this thesis follows both degree distributions during the code construction procedures, changing the edge selection criterion, thus obtaining a better approach to codes with an irregular degree distribution.

3.3.2 Free Check-Node Degree Criterion

The edge selection procedure used in this proposal differs from the selection procedure proposed in the original PEG algorithm. The graph is analyzed to avoid local short cycles, however check nodes are not chosen according to its number of assigned edges, $d^k(c_i)$, i.e. its current (or partial) degree. Instead of this, the check node with the highest difference between its partial and final-defined degree is chosen, $f(c_i) = d(c_i) - d^k(c_i)$, i.e. the difference between the number of currently assigned edges and the total number of edges to be assigned. The lowest check node degree procedure is replaced by a highest free check node degree (FCD) procedure (see Figure 3.4). The FCD concept, comes from the concept of *sockets* previously described in Refs. [31, 71].

Figure 3.4 shows three Tanner graphs illustrating some characteristics of the algorithm. In (a) the zig-zag pattern used for 2-degree symbol nodes is shown. Subfigures (b) and (c) show check nodes with different degrees, $d(c_i)$ and $d(c_j)$, and different number of incident edges (partially assigned), $d^k(c_i) = 1$ and $d^k(c_j) = 2$. The first check node, c_i, is selected if a lowest check node degree criterion is used, while the second node, c_j, is selected if the criterion used is the FCD [72].

We introduce the concept of *compliance* of a constructed code as the distance between the distribution of nodes (symbol or check nodes) in the code and the pre-established node degree distribution. Let ρ_j be the pre-established probability distri-

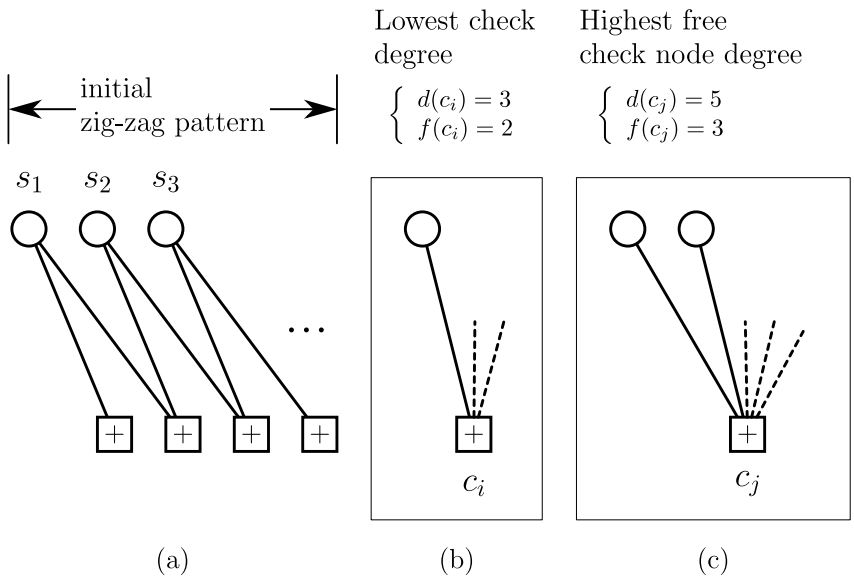

Figure 3.4: Progressive edge-growth criteria for the selection of check nodes and zig-zag pattern.

bution for a degree j check node, and ρ_j^* the actual probability of a degree j check node in the constructed graph, we calculate the ρ-compliance of a code as follows:

$$\eta = \sum_{j=2}^{d_c^{\max}} \left| \rho_j - \rho_j^* \right| \tag{3.29}$$

where $\rho^*(x) = \sum_{j=2}^{d_c^{\max}} \rho_j^* x^{j-1}$.

3.3.3 Proposed Algorithm

A modified PEG algorithm is described in Algorithm 1.

The same notation as in Ref. [66] is used, where $d(s_j)$ is the s_j symbol node degree —i.e. the number of incident edges, it corresponds to the cardinality of the ensemble E_{s_j} after the code construction—, $d(c_i)$ is the c_i check node degree, $f(c_i)$ is the number of edges that can be added to the check node c_j under the current graph setting, such that $d^k(c_i) = f(c_i) - d(c_i)$, E_{s_j} is the ensemble of edges connected to the symbol node s_j, $E_{s_j}^k$ the edge added in the step k of the progressive construction, and $\mathcal{N}_{s_j}^l$ is the

ensemble of nodes reached after the graph expansion from the symbol node s_j up to depth l.

Algorithm 1 Improved Progressive Edge-Growth

Require: $d(s_i) \leq d(s_j)\ \forall i < j$ and $f(c_i) = d(c_i)\ \forall i$

for $j = 1$ to n **do**

for $k = 1$ to $d(s_j)$ **do**

if $k = 1$ **then**

if $d(s_j) = 2$ **then**

$E^1_{s_j} \leftarrow (c_i, s_j)$, where $E^1_{s_j}$ is the first edge incident to s_j and c_i is a check node such that it has the *lowest check-node degree* under the current graph setting $E_{s_1} \cup E_{s_2} \cup \cdots \cup E_{s_{j-1}}$.

else

$E^1_{s_j} \leftarrow (c_i, s_j)$, where $E^1_{s_j}$ is the first edge incident to s_j and c_i is a check node such that it has the *highest free check-node degree*.

end if

else

Expand a subgraph from symbol node s_j up to depth l under the current graph setting, such that $\mathcal{N}^l_{s_j} = \mathcal{N}^{l+1}_{s_j}$, or $\overline{\mathcal{N}}^l_{s_j} \neq \emptyset$ but $\overline{\mathcal{N}}^{l+1}_{s_j} = \emptyset$.

$E^k_{s_j} \leftarrow (c_i, s_j)$, where $E^k_{s_j}$ is the k-th edge incident to s_j and c_i is a check node picked from the set $\overline{\mathcal{N}}^l_{s_j}$ having the *highest free check-node degree*.

end if

$f(c_i) = f(c_i) - 1$

end for

end for

A zig-zag construction for 2-degree symbol nodes (see Figure 3.4) is forced by using an special criterion when adding the first edge to a symbol node. In this particular selection, a list of eligible check nodes is limited to those check nodes already connected under the current graph setting, $E_{s'} = E_{s_1} \cup E_{s_2} \cup \cdots \cup E_{s_{j-1}}$, i.e. to

Table 3.1: ρ-compliance of LDPC codes constructed using four different PEG-based algorithms.

Relaxed edge-selection	(1)	(2)	(3)	(4)
No	1.938722	0.050631	0.057456	0.050050
Yes	–	0.001623	0.001634	0.000510

the list of check nodes that have been chosen at least once from the first to j-th step. This construction is used to avoid cycles with 2-degree symbol nodes, thus obtaining a good performance in the error floor region as shown in the simulation results (see Section 3.3.4 below).

Relaxed edge selection

The proposed PEG algorithm can be modified to work with a relaxed edge selection. In this case, if there are not check nodes with free edges in the final ensemble of candidate check nodes, $\overline{\mathcal{N}}_{s_j}^{l}$, check nodes with free edges are searched in the previous candidate ensemble, $\overline{\mathcal{N}}_{s_j}^{l-1}$. This procedure improves the ρ-compliance, η, with the target check node degree distribution, $\rho(x)$, at the expense of the current local cycle length (see Table 3.1).

3.3.4 Simulation Results

Simulations results have been computed to compare four different PEG-based construction methods: (1) the original PEG algorithm as proposed in Refs. [66,67]; (2) the modified PEG algorithm proposed by Richter in Ref. [71]; (3) the modified PEG algorithm proposed here; and (4) a mixed version, where the lowest check node degree criterion is used to connect the first edge to a symbol node (not only to 2-degree symbol nodes as proposed here), and the FCD criterion is used for the remaining edges. All constructed codes have a codeword of 10^5 bits length and rate one half, $R = 0.5$.

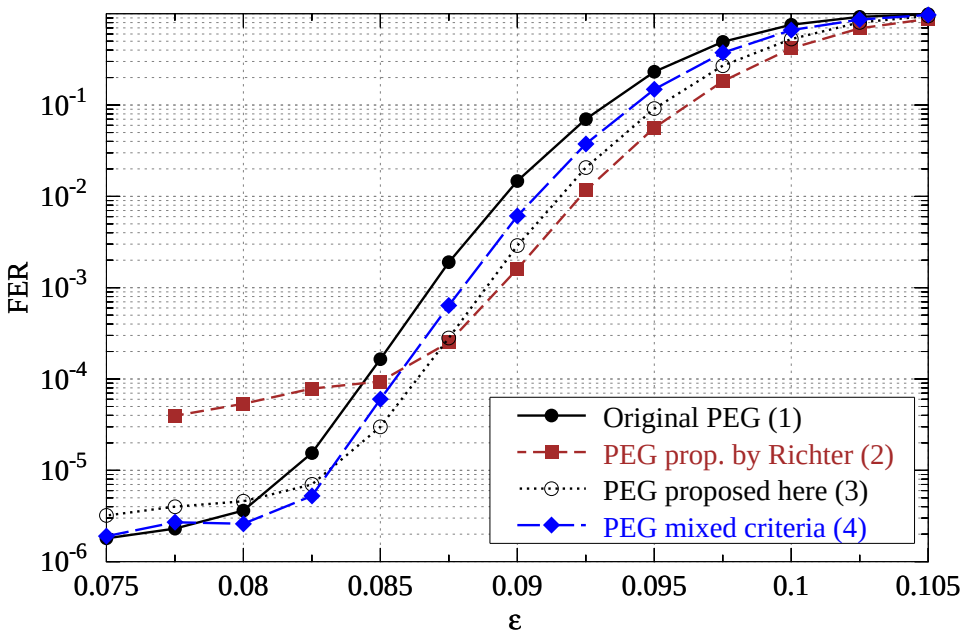

Figure 3.5: Performance over the BSC with crossover probability ϵ of four LDPC codes constructed using different PEG-based algorithms.

In Table 3.1 the ρ-compliance, calculated as defined in Eq. (3.29), is compared for two edge selection criteria: the relaxed edge selection and the selection criterion proposed in Algorithm 1. We have constructed codes ad-hoc for these simulations. The values shown in this table were calculated for those codes. Results show that the relaxed edge selection criterion allows a better approximation of the degree distribution in $\rho(x)$.

Performance has been computed under iterative decoding by using the sum-product algorithm with flooding schedule. The maximum number of iterations for the decoder was set to 2000.

Figure 3.5 shows the performance of these codes over the BSC. Frame error rate as a function of the crossover probability, ϵ, is analyzed for four construction methods. Codes have been constructed using the optimized generating polynomials from Ref. [26]. Since the $\rho(x)$ distribution is more complex in this ensemble of codes, and thus it is possible to better appreciate the differences among the various algo-

rithms used for constructing the codes. The error floor is improved using the zig-zag construction for 2-degree symbol nodes. On the other hand, within a given graph setting, when the first check node connected to a symbol node rule is used, there is no relevant improvement.

Chapter 4

Rate-Adaptive Reconciliation with Low-Density Parity-Check Codes

Throughout this chapter we analyze different techniques used to adapt the information rate of a linear code, and we emphasize how these techniques behave when using LDPC codes. To this end, the chapter is organized as follows. In Section 4.1 we show some figures of merit to introduce the interest of rate-compatible codes for the information reconciliation problem. Next, in Section 4.2 we introduce some techniques commonly used to adapt the information rate of LDPC codes. In Section 4.3 we propose a rate-adaptive protocol specifically designed for reconciliation using LDPC codes. Finally, in Section 4.4 we show some simulation results for the rate-adaptive protocol proposed here.

4.1 Introduction

When using an LDPC code the redundancy is determined by the information rate of the code (see Eq. (3.9)). Once the coding rate has been established, the efficiency for reconciliation, as defined in Eq. (3.7), is then a function that depends on a single parameter, the error rate ϵ. This efficiency decreases in the range $\epsilon \in [0, \epsilon_{\max}]$, where

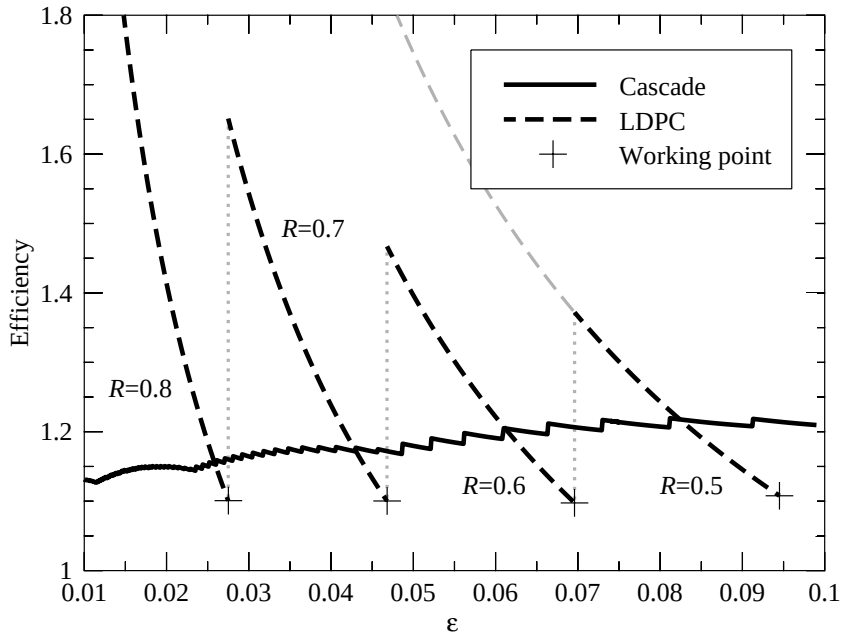

Figure 4.1: Reconciliation efficiency for the error rate ϵ of *Cascade* and LDPC codes without using any rate-adaptive technique.

$\epsilon_{\max}$ is the maximum error rate that can be reconciled using the selected code. For error rate values far from this $\epsilon_{\max}$ the efficiency is also far from its optimal value since the redundancy used is excessive.

We can improve the reconciliation efficiency using a set of LDPC codes instead of just one. We choose then the code with the best efficiency for every value of ϵ, but we cannot avoid a characteristic saw behavior: the efficiency is good for ϵ values just below the $\epsilon_{\max}$ of every code, and it degrades till the next code is used. This forces the use of many codes in order to cover a broad range of ϵ with good efficiency, not a very practical proposition, for instance when working in time-varying channels or simply considering a finite length analysis.

This behavior of the reconciliation efficiency using LDPC codes as a function of the characteristic parameter, here the error rate ϵ, is shown in Figure 4.1. The figure shows the reconciliation efficiency using LDPC codes of 2×10^5 bits length as a function of the channel error rate, ϵ. Since the redundancy is fixed for a ϵ range,

it is excessive and far from optimal for good channels, i.e. low values of ϵ; as the light grey line shows for a reconciliation method based in a single LDPC code. The dashed curve marked as LDPC shows a set of codes where the best efficiency code is chosen for every value of ϵ. When ϵ is just below or at the *working* point[1], no more redundancy than necessary is used, thereby producing high efficiency codes. As we move away from this point using the same code, the efficiency moves further from the optimal value. In the figure, a solid line depicts the efficiency of *Cascade* [5].

A rate-adaptive coding is then considered crucial in this context —i.e. secret-key agreement—, since the efficiency during the information reconciliation process is a determining factor for the final secret-key length.

4.2 Rate-Adaptive LDPC Coding

An error correcting code is considered to be *rateless* or *rate-compatible* when the information rate of the code can be dynamically adapted according to the communication requirements. LDPC codes are not rate-compatible in nature. However, there exist some techniques, such as puncturing or shortening, that can be used to adapt the information rate of these codes. A rate-adapting procedure is usually known as *rate modulation*. Construction of rate-compatible LDPC codes was originally analyzed in Refs. [73,74].

4.2.1 Puncturing

A well-known technique commonly used to modulate the information rate of a linear code is *puncturing*. It modulates the rate of a previously constructed code, $\mathcal{C}(n,k)$, by deleting a set of p symbols from the codewords, $p < n$, converting it into a $\mathcal{C}(n-p,k)$

[1]We refer to working point as the maximum value of ϵ for which the code is able to reconcile an string with high probability, i.e. assuming a low frame error rate (FER), typically an appropriate value of FER is 10^{-3}.

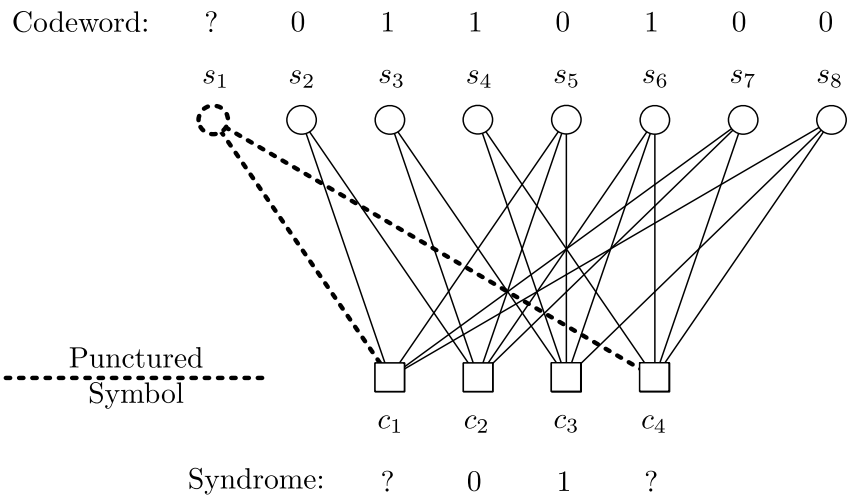

Figure 4.2: Example of puncturing applied to a linear code represented by its Tanner graph.

code. The coding rate is then increased to:

$$R = \frac{k}{n-p} = \frac{R_0}{1-\pi} \tag{4.1}$$

where $R_0 = k/n$ is the rate of the original code (mother code) and $\pi = p/n$ is the fraction of punctured symbols, $\pi < 1$.

Figure 4.2 shows an example of puncturing applied to a linear code. In the corresponding Tanner graph, one symbol is deleted from the word and a $\mathcal{C}(8,4)$ code with rate one half $R_0 = 1/2$, is converted to a $\mathcal{C}(7,4)$ code, increasing its rate to $R = 4/7$. The value of the punctured symbol is then considered unknown, and this uncertainty is translated to its neighboring set (check nodes).

Puncturing was originally studied for its application with LDPC codes in Ref. [75]. Later, it was proved that the performance of punctured LDPC codes is as good as the performance of ordinary ones —i.e. punctured LDPC codes are also capacity achieving—, existing a puncturing threshold for every family of these codes [76,77]. Optimized puncturing distributions were also analyzed using the density evolution for the asymptotic case [78,79]. Furthermore, the behavior of finite length LDPC

codes in the waterfall region has been also studied when puncturing over the binary erasure channel [80]. A nice survey about puncturing and rate-adaptive LDPC codes can be found in Ref. [81]. In this regard, it should also be noted that other strategies, not discussed here, can be used to improve the performance of rate-compatible LDPC codes. For instance, the design of good irregular codes suitable for high puncturing rates[2] was analyzed in Refs. [82–84].

The rate of a code can be adapted in a syndrome source coding scheme as follows. Let $\mathcal{C}(n,k)$ be a code that can be used to correct noise up to $\epsilon_{\max}$ for some channel family, and let $\mathbf{x}$ and $\mathbf{y}$ be two m-length strings, with $m = n - p$, correlated as if they were the input and output of a channel characterized by $\epsilon < \epsilon_{\max}$ —i.e. $\mathbf{x}$ and $\mathbf{y}$ are two instances of X and Y, respectively—. The encoder sends the syndrome in $\mathcal{C}$ of a word $\hat{\mathbf{x}}$ constructed by embedding $\mathbf{x}$ in a string of length n and filling the other p positions with random bits. If the new coding rate, $R(p) = R_0/(1-p)$ is adapted to ϵ the decoder should recover $\mathbf{x}$ from $\hat{\mathbf{x}}$ with high probability.

We can think of a reconciliation protocol based only in punctured codes: the parties would agree on an acceptable frame error rate (FER) and, depending on their estimation of the error rate, they would choose the optimal value of p. If we consider the behavior of FER as a function of ϵ for a set of fixed p values, as depicted in Figure 4.3, this procedure can be regarded as moving along the horizontal axis from one code to the next. However, this way of proceeding has the shortcoming that if the channel is time varying —i.e. ϵ varies over time—, the length of $\mathbf{x}$ and $\mathbf{y}$ also varies to accommodate the different values of p needed to adapt the coding rate. We could think of scenarios where $m \gg n$ and two instances of X and Y can be divided in packets of length $n - p$ but this clearly does not apply to many situations.

Figure 4.3 shows the frame error rate (FER) over the binary symmetric channel (BSC) with crossover probability ϵ for a binary LDPC code of 2×10^3 bits length and rate one half, $R_0 = 1/2$. Several curves have been simulated for different proportions

[2] Efficiently-encodable rate-compatible (E^2RC) codes.

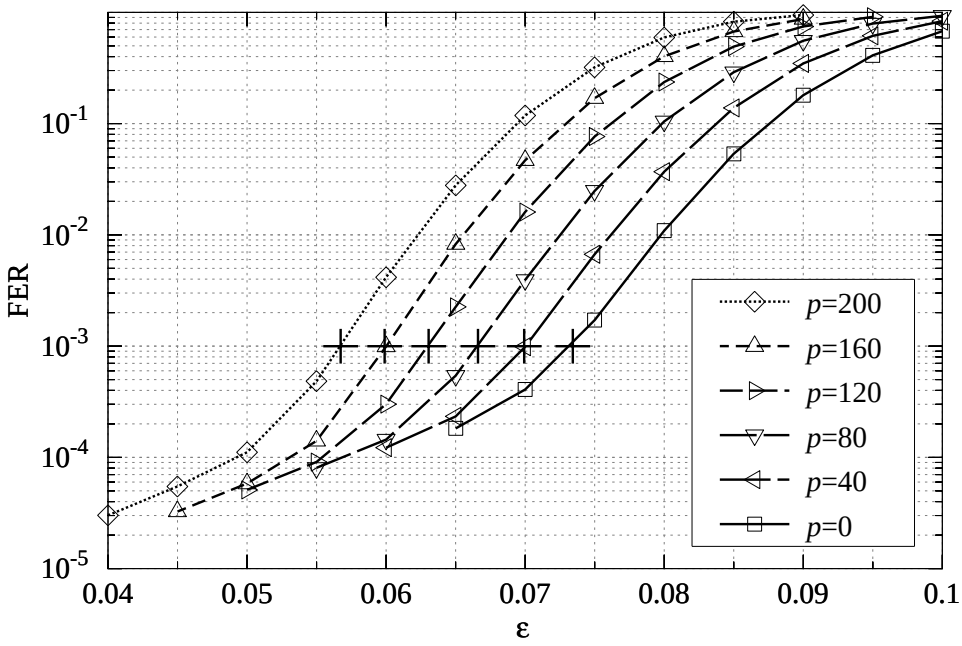

Figure 4.3: Performance over the BSC with crossover probability ϵ for a short-length LDPC code and different proportion of punctured symbols.

of punctured symbols. Due to the short code length, the distribution of punctured symbols has been intentionally chosen according to an optimized pattern as proposed in Ref. [85].

4.2.2 Intentional Puncturing

Specially when working with short length codes, but also when working with high puncturing rates, the ensemble of punctured symbol nodes determines the decoding performance. In this regard, puncturing has been analyzed for the finite length case, and several algorithms have been proposed for finding good puncturing patterns that can be efficiently applied in finite length LDPC codes [85–89]. In Ref. [85] the authors introduce the concept of *intentional* puncturing as alternative to the random puncturing previously used. In intentional puncturing the ensemble of symbol nodes to puncture is chosen following a list of puncturable nodes —previously computed—

instead of a random fashion.

A well-known method for determining maximum puncturing patterns in LDPC codes is proposed in Ref. [85]. These puncturing patterns have to be generated prior to the coding procedure, and it can be a relatively convoluted procedure for long codes since it requires to process the entire parity-check matrix. The authors later analyze how the ensemble of punctured nodes affects the decoding, and they propose a particular schedule for decoding that improves the decoding performance of layered-based LDPC codes [90]. Basically, in this *intentional* decoding the algorithm tries to compute first those messages from non-punctured symbol nodes in order to recover punctured ones, as it can be done when using for instance a serial schedule scheme for decoding as described in Section 3.2.

Puncturing Short-Length LDPC Codes

In this work, we focus exclusively on those intentional puncturing methods that minimize the impact of puncturing in the decoding of short length codes. We describe here a new finite length method for intentional puncturing based on the concept of next neighboring set introduced in Section 2.4. Contrary to the method proposed in Ref. [85], this algorithm focused on finding good puncturing patterns for short-length LDPC codes and it can be efficiently applied analyzing only the 2-depth local graph of each punctured node.

In Ref. [85] the concept of *one-step recoverable* (1-SR) is defined for a symbol node when there is at least one survived node within the set of adjacent check nodes, and therefore the symbol node can be recovered in one step. A check node is considered survived if there are no punctured nodes within the set of adjacent symbol nodes. We introduce here the concept of *one-step untainted*, based on a similar definition of 1-SR, to propose a simple method that chooses symbols such that all the adjacent check nodes are survived nodes.

Let $\mathcal{N}^2(k)$ be the next neighboring set of a symbol node s_k as defined in Eq. (2.31).

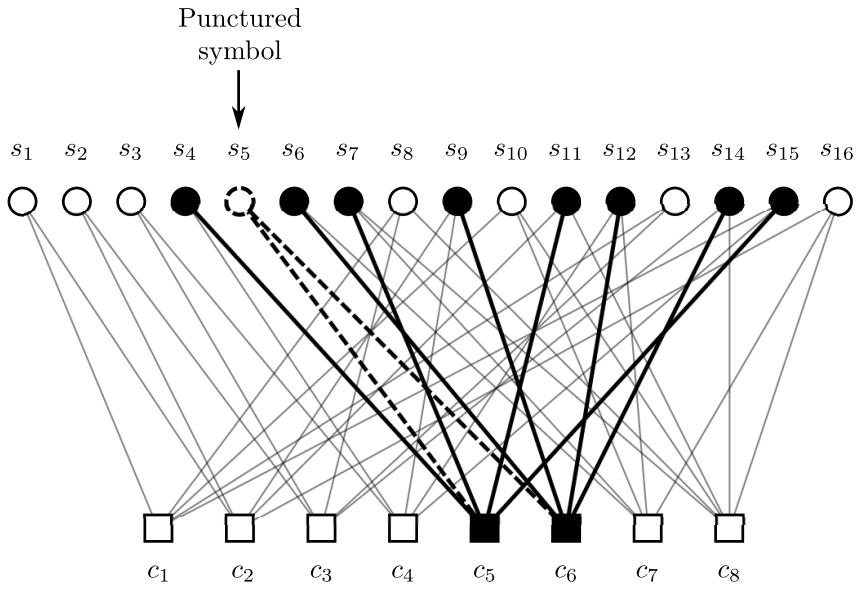

Figure 4.4: Next-neighboring set of a punctured symbol node.

Definition 14. *A symbol node s_k is said to be one-step untainted (1-SU) if there are no punctured symbols within its next neighboring set $\mathcal{N}^2(k)$.*

Figure 4.4 shows an example. In this example, x_5 is a symbol node selected to be punctured, and the set $\{s_4, s_6, s_7, s_9, s_{11}, s_{12}, s_{14}, s_{15}\}$ is the next neighboring set of symbol nodes that will be excluded in following selections of the proposed method. The neighboring set is computed from the set of check nodes adjacent to the selected symbol, $\{c_5, c_6\}$ in the current example.

Proposed Algorithm

Let $\mathcal{X}_\infty$ be a set of symbol nodes that are not affected by the current selection of punctured symbol nodes, i.e. $\mathcal{X}_\infty$ is the ensemble including every 1-SU symbol node. And let $\mathcal{Z}_\infty$ be the set containing every check node which is not adjacent to any punctured symbol. At the beginning, when there are no punctured symbols, both sets $\mathcal{X}_\infty$ and $\mathcal{Z}_\infty$ consist of every symbol and check node, respectively. Let $p_{\max}$ be the number of symbols to be punctured, the proposed intentional puncturing can be then described as in Algorithm 2.

Algorithm 2 Intentional Puncturing

{*Initialization*}

$\mathcal{V} = \emptyset$

$\mathcal{X}_\infty = \{1, ..., n\}$

$\mathcal{Z}_\infty = \{1, ..., m\}$

$p = 1$

while $p \leq p_{\max}$ and $\mathcal{X}_\infty \neq \emptyset$ **do**

{*Step 1.– Compute 1-SU under the current pattern*}

Make the next neighboring set $\mathcal{G}(k)$ under the current puncturing pattern, a subset of $\mathcal{N}^2(k)$, such that $\mathcal{G}(k) = \{i : i \in \mathcal{N}(j), \forall j \in \mathcal{M}(k) \cap \mathcal{Z}_\infty\}$, for each $k \in \mathcal{X}_\infty$. Similarly, there should be $\mathcal{G}(k) = \mathcal{N}^2(k) \cap \mathcal{X}_\infty$.

{*Step 2.– Look for candidates*}

Make the set of candidates Ω, a subset of $\mathcal{X}_\infty$, such that $i \in \Omega$ if $|\mathcal{G}(i)| \leq |\mathcal{G}(k)|$ for all $k \in \mathcal{X}_\infty$.

{*Step 3.– Selection for puncturing*}

Pick a symbol node s_i, such that $i \in \Omega$. Pick one randomly if there exist more than one symbol in Ω.

{*Step 4.– Updating sets*}

$\mathcal{V} = \mathcal{V} \cup \{i\}$

$\mathcal{X}_\infty = \mathcal{X}_\infty \backslash \{i\}$

$\mathcal{X}_\infty = \mathcal{X}_\infty \backslash \{k\}$ for all $k \in \mathcal{G}(i)$, $k \neq i$

$\mathcal{Z}_\infty = \mathcal{Z}_\infty \backslash \mathcal{M}(i)$

$p = p + 1$

end while

Table 4.1: Highest and lowest upper bound for maximum puncturing.

Rate	$R = 0.3$	$R = 0.4$	$R = 0.5$	$R = 0.6$
$p^{+}_{\max}$ [a]	4740	4135	3551	2978
$p^{-}_{\min}$	4642	4032	3444	2877
$p^{+}_{\max}$ [b]	2685	2373	1986	1643
$p^{-}_{\min}$	2608	2299	1916	1585

[a] Algorithm proposed in Ref. [85].
[b] Algorithm proposed here.

The algorithm concludes when it chooses $p_{\max}$ symbol nodes or there are no more selectable symbols to be punctured, i.e. $\mathcal{X}_{\infty} = \emptyset$. Table 4.1 shows the highest and lowest upper bound of $p_{\max}$, $p^{+}_{\max}$ and $p^{-}_{\max}$ respectively, observed over all simulations. This algorithm allows for a smaller number of punctured symbols, compared to the proposed in Ref. [85], which also implies a reduction in the achievable rate through puncturing. However, it is shown below that the performance of intentional punctured codes with the proposed algorithm is better than the one in Ref. [85].

At the end, the algorithm returns the set $\mathcal{V}$ consisting of those symbol nodes selected during the third step.

Simplified Version. Lowest Check-Node Degree Criterion

Notice that, whenever a code is constructed with an almost regular fraction of edges per check node —as occurs in the original PEG algorithm—, the first and second step can be simplified. Instead of looking for a symbol node with the smallest next neighboring set under the current puncturing pattern, a symbol node s_i with the lowest check node degree $\mathcal{M}(i)$ can be used. This simplified version is described in Algorithm 3.

Algorithm 3 Intentional Puncturing (Simplified Version)

{*Initialization*}

$\mathcal{V} = \emptyset$

$\mathcal{X}_\infty = \{1, ..., n\}$

$\mathcal{Z}_\infty = \{1, ..., m\}$

$p = 1$

while $p \leq p_{\max}$ and $\mathcal{X}_\infty \neq \emptyset$ **do**

{*Step 1.– Look for candidates*}

Make the set of candidates Ω, a subset of $\mathcal{X}_\infty$, such that $i \in \Omega$ if $|\mathcal{M}(i) \cap \mathcal{Z}_\infty| \leq |\mathcal{M}(k) \cap \mathcal{Z}_\infty|$ for all $k \in \mathcal{X}_\infty$.

{*Step 2.– Selection for puncturing*}

Pick a symbol node s_i, such that $i \in \Omega$. Pick one randomly if there exist more than one symbol in Ω.

{*Step 3.– Updating sets*}

$\mathcal{V} = \mathcal{V} \cup \{i\}$

$\mathcal{X}_\infty = \mathcal{X}_\infty \backslash \{i\}$

for all $j \in \mathcal{M}(i)$ **do**

$\mathcal{X}_\infty = \mathcal{X}_\infty \backslash \{k\}$ for all $k \in \mathcal{N}(j)$, $k \neq i$

$\mathcal{Z}_\infty$ = $\mathcal{Z}_\infty \backslash \{j\}$

end for

$p = p + 1$

end while

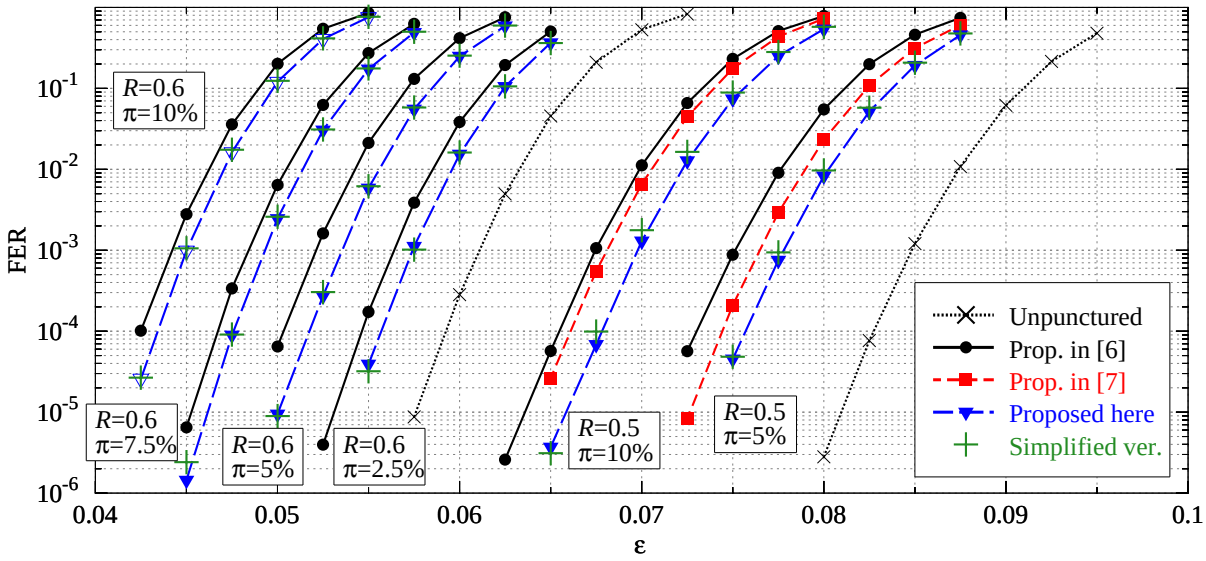

Figure 4.5: Performance over the BSC with crossover probability ϵ of different strategies for intentional puncturing. Coding rates used: $R = 0.5$ and $R = 0.6$.

Simulation Results

We have simulated the behavior of punctured codes over the binary symmetric channel with crossover probability ϵ. Results were computed using LDPC codes of 10^4 bits length and different coding rates: $R = 0.3$, $R = 0.4$, $R = 0.5$ and $R = 0.6$. These codes were constructed using the original PEG algorithm as proposed in Ref. [66]. Results were computed under iterative decoding, using a sum-product algorithm with serial schedule and 200 iterations maximum.

Figures 4.5 and 4.6 show FER over the BSC with crossover probability ϵ for different intentional puncturing strategies. Two LDPC codes were used with different coding rates and different proportions of punctured symbols. In these figures, the use of puncturing patterns as in Ref. [85] is compared with the algorithm proposed here, which is also compared with the one in Ref. [88].

These results demonstrate that the degree of punctured symbol nodes should be taken into account. The proposed algorithm is also compared to its simplified version,

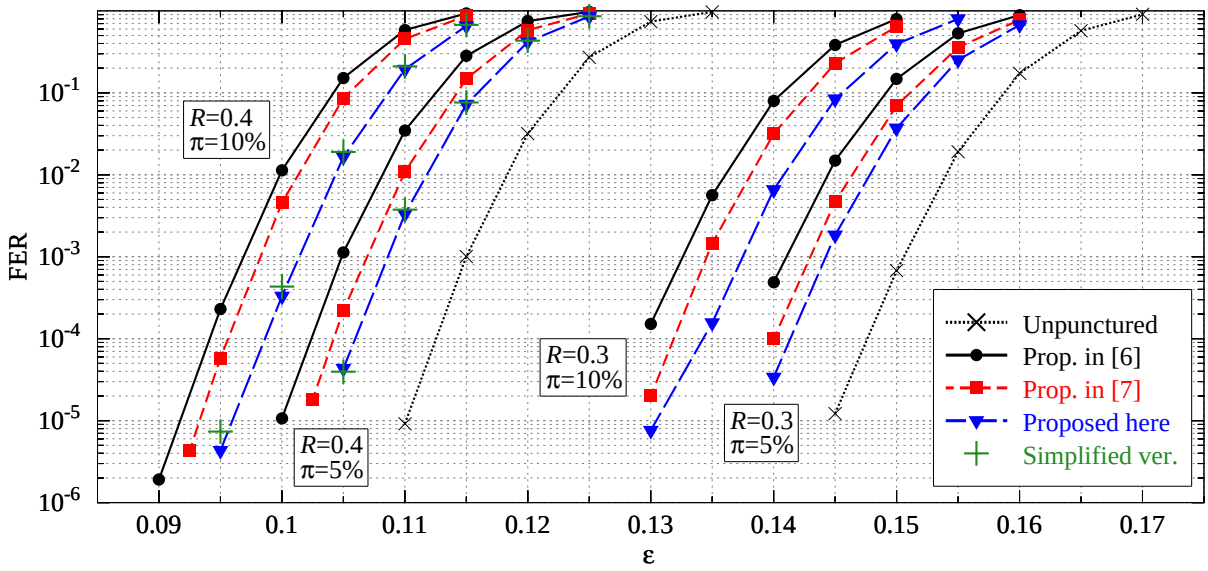

Figure 4.6: Performance over the BSC with crossover probability ϵ of different strategies for intentional puncturing. Coding rates used: $R = 0.3$ and $R = 0.4$.

and it shows that the lowest symbol degree criterion is preferable, for simplicity, at least when using PEG-based LDPC codes.

4.2.3 Shortening

Puncturing increases the rate by reducing the redundancy. The opposite is achieved through *shortening*: by increasing the redundancy, the information rate is reduced. This is done by fixing the value of a set of s symbols from the codewords in positions known to encoder and decoder. Shortening, then, converts a $\mathcal{C}(n,k)$ code in a $\mathcal{C}(n-s,k-s)$ one [91].

Figure 4.7 shows an example of shortening applied to a linear code. In the corresponding Tanner graph one symbol is deleted from the encoding and a $\mathcal{C}(8,4)$ code with rate one half $R_0 = 1/2$, is converted to a $\mathcal{C}(7,3)$ code, decreasing the rate to $R = 3/7$.

Let s be the number of shortened symbols, such that $s < n$, the information rate

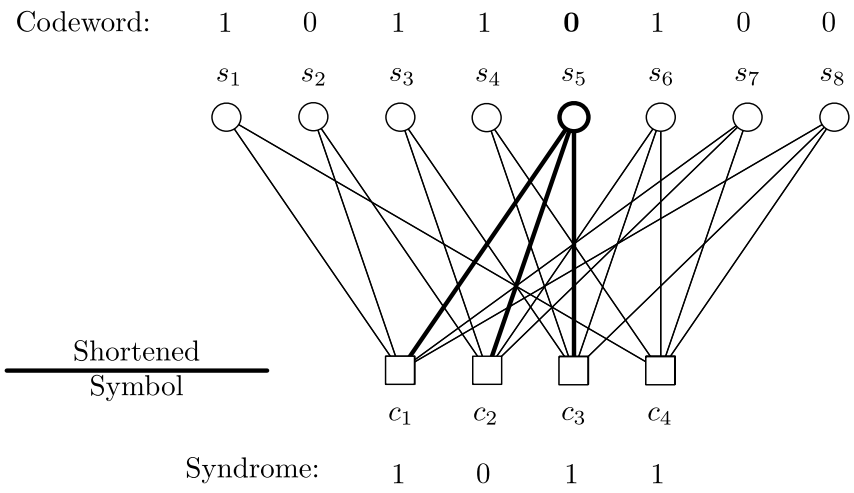

Figure 4.7: Shortening applied to a linear code.

of a shortened code is given by:

$$R = \frac{k-s}{n-s} = \frac{R_0 - \sigma}{1-\sigma} \tag{4.2}$$

where R_0 is the rate of the mother code (unshortened code), and $\sigma = s/n$ is the fraction of shortened symbols, such that $\sigma < 1$.

Note that the set of shortened symbols is chosen now randomly from the set of selectable symbol nodes. Techniques for the *intentional* selection of these symbols are not needed, even when using short-length codes —contrary to puncturing where these techniques, as the commented intentional puncturing, provide a considerable improvement for decoding—. However, intentional shortening can be considered when using puncturing and shortening simultaneously as it is commented below.

Figure 4.8 shows the performance over the BSC with crossover probability ϵ for a binary LDPC code of 2×10^3 bits length and rate one half, $R_0 = 1/2$. Several curves have been simulated for different proportions of shortened symbols. The distribution of shortened symbols has been chosen randomly.

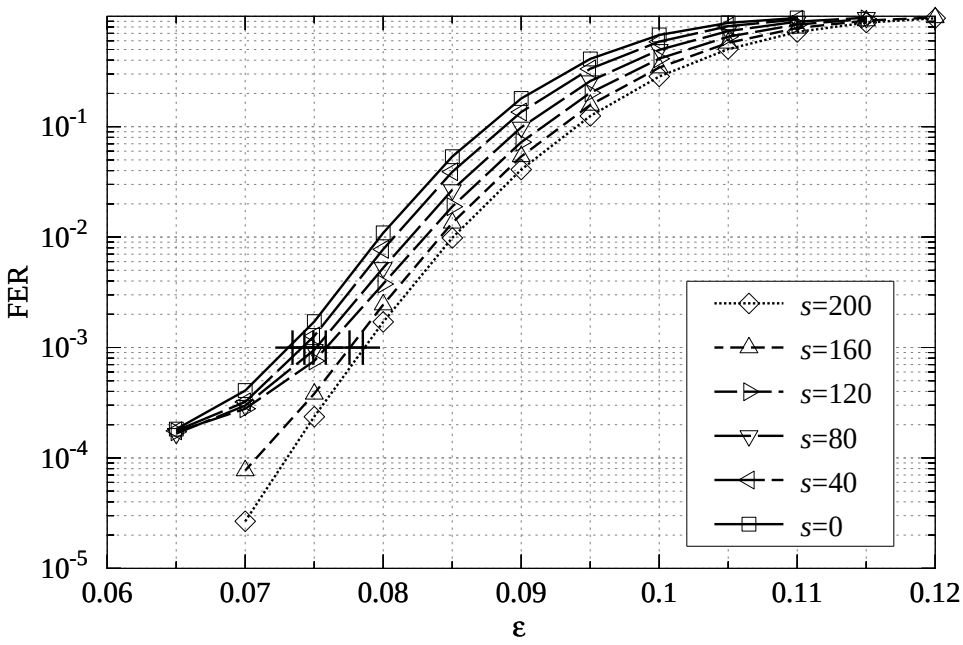

Figure 4.8: Performance over the BSC with crossover probability ϵ for a short-length LDPC code and different proportion of shortened symbols.

4.3 Rate-Adaptive LDPC Reconciliation

Typically, only puncturing or shortening are used to adapt the information rate of a code. However, when using syndrome coding over time varying channels, using just one of the two has the drawback that modifying the value of p or s implies modifying also the length of the reconciled strings with every code use. The combined application of both techniques allows to fix the length of the strings to reconcile and overcome this problem. In this case, a modulation parameter $d = p + s$ can be set, thus fixing the lengths of two instances of X and Y to $n - d$ while allowing to modify p and s in order to adapt to different values of the channel parameter ϵ.

The result of simultaneously puncturing p symbols and shortening s symbols in the original code, as proposed in Ref. [32], is thus a $\mathcal{C}(n - p - s, k - s)$ code with information rate:

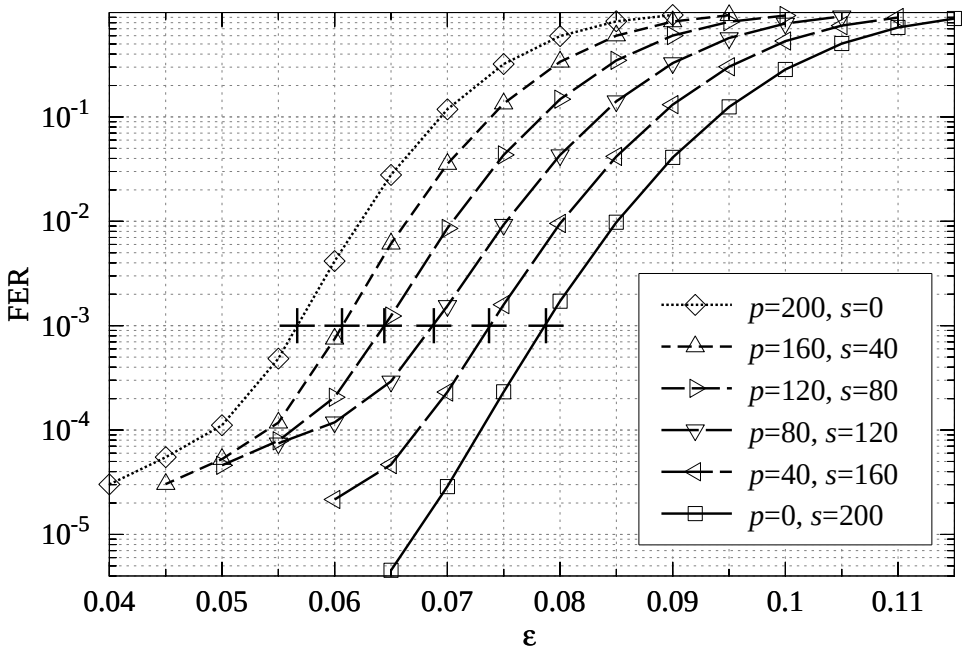

Figure 4.9: Performance over the BSC with crossover probability ϵ for a short-length LDPC code and different proportion of punctured and shortened symbols.

$$R = \frac{k-s}{n-p-s} = \frac{R_0 - \sigma}{1-\pi-\sigma} \tag{4.3}$$

where p and π are the number and the fraction of punctured symbols, respectively, as defined above, such that $\pi \geq 0$, $\sigma \geq 0$ and $\pi + \sigma < 1$.

Let δ be the fraction of punctured and shortened symbols, $\delta = d/n = \pi + \sigma$, a *$\delta$-modulated* rate-adaptive code covers the range of rates $[R_{\min}, R_{\max}]$ given by:

$$R_{\min} = \frac{R_0 - \delta}{1-\delta} \leq R \leq \frac{R_0}{1-\delta} = R_{\max} \tag{4.4}$$

Figure 4.9 shows the performance of an error correcting code, again depicted as FER versus the error rate of a BSC(ϵ) using both techniques simultaneously. A binary short-length LDPC code of 2×10^3 bits length and rate one half, $R_0 = 1/2$, was used. Several curves have been simulated for different proportions of punctured and

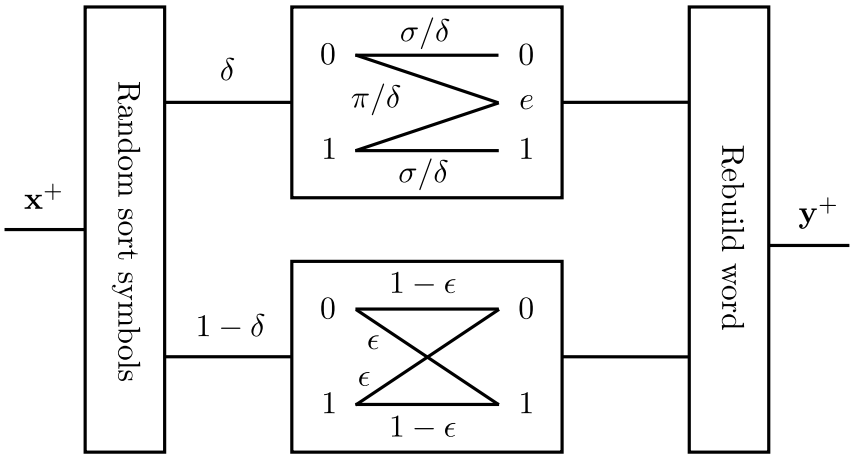

Figure 4.10: Channel model for the proposed rate-adaptive reconciliation protocol assuming random puncturing and shortening.

shortened symbols, with $d = 200$. The fraction of punctured and shortened symbols was chosen taking into account the asymptotic behavior of a *δ-modulated* rate-adaptive code shown in Appendix A. Punctured symbols have been chosen according to a pattern previously estimated, as proposed in Ref. [85], while shortened symbols have been chosen in random fashion.

4.3.1 Rate-Adaptive Reconciliation Protocol

We formally describe below (see Algorithm 4) the method for rate-adaptive reconciliation using puncturing and shortening techniques outlined above. Note that the proposed reconciliation protocol can be carried out in the opposite direction —i.e. exchanging Alice and Bob's roles—. The process is then commonly referred as *reverse* reconciliation [92].

A graphical interpretation of the proposed protocol is also depicted in Figure 4.10. A similar depiction was already proposed in Ref. [76]. As shown in the figure, puncturing and shortening can be interpreted as the transmission of punctured and shortened symbols over a binary erasure channel (BEC) with crossover probability π/δ. In the figure, it is used the same notation as used above to denote the fraction of punc-

Algorithm 4 Rate-Adaptive Reconciliation Protocol

Step 0: Set up.— Let $\mathcal{C}(n,k)$ be a code $\mathcal{C}$ that can correct noise up to $\epsilon_{\max}$ for some channel family. Let $\mathbf{x}$ and $\mathbf{y}$ be two strings that two parties Alice and Bob wish to reconcile. Let $\mathbf{x}$ and $\mathbf{y}$ be of length m, with $m = n - d$, and every symbol of $\mathbf{y}$ the output of a memoryless channel characterized by $\epsilon < \epsilon_{\max}$. Alice and Bob establish:

$$s = \lceil \left(R_0 - \frac{1-d}{n} R \right) \cdot n \rceil \tag{4.5}$$

$$p = d - s \tag{4.6}$$

Step 1: Encoding.— Alice sends the syndrome in $\mathcal{C}$ of a word $\hat{\mathbf{x}}$ consisting on embedding $\mathbf{x}$ in a n-length string and filling the remaining d positions (punctured and shortened positions) with random symbols. Together with the syndrome, Alice and Bob have to share the set of d positions and their values (i.e. every value in a shortened symbol has to be transmitted).

Note that these positions and values can be synchronized using a pseudo-random generator, avoiding then the corresponding channel bandwidth —and its corresponding authentication—.

Step 2: Decoding.— Bob constructs the word $\hat{\mathbf{y}}$ consisting on the concatenation of $\mathbf{y}$ the received s symbols and p random symbols. If Bob recovers $\mathbf{x}$ he reports success and the protocol ends.

tured and shortened symbols, π and σ respectively, such that $\delta = \pi + \sigma$. Random puncturing and shortening are assumed in the figure.

4.4 Simulation Results

Simulation results were computed to compare the efficiency of the rate-adaptive reconciliation protocol proposed here and the efficiency of *Cascade* as proposed in Ref. [5]. Figure 4.11 shows the efficiency, calculated as defined in Eq. (3.2), over the binary symmetric channel with crossover probability ϵ. Unmodulated LDPC codes as discussed in Ref. [26] are also depicted in the figure.

The efficiency of *Cascade* was computed for the reconciliation of 2×10^5 bit-length strings. Discontinuities in the curve of *Cascade* are due to initial block size used in the protocol, given by $k_1 = 0.73/\epsilon$ [11].

The efficiency of four LDPC codes covering a range of high error rates is also depicted. They are referred as *unmodulated* codes since it was not used any method to adapt the coding rate of these codes. It was used binary LDPC codes of 2×10^5 bits length and coding rates $R = 0.5$, $R = 0.6$, $R = 0.7$ and $R = 0.8$. These codes were constructed using ensembles of LDPC codes specifically optimized for the BSC[3] and the progressive edge-growth algorithm for the construction of irregular codes described in Section 3.3. For this simplest approach, the efficiency is getting worse quickly as it can be appreciated in the figure. The unmodulated LDPC codes exhibit an undesirable saw behavior that can lead to efficiencies worse than that of *Cascade* unless many different codes are used.

Simulation results for the proposed rate-adaptive reconciliation protocol were computed using two LDPC codes of 2×10^3 and 2×10^5 bits length, respectively, and coding rate one half $R_0 = 0.5$. The rate-adaptive approach was carried out with a 10% of punctured and shortened symbols, $\delta = 0.1$, and thus covering the high error

[3]Generating polynomials of these families of LDPC codes can be found in Appendix C.

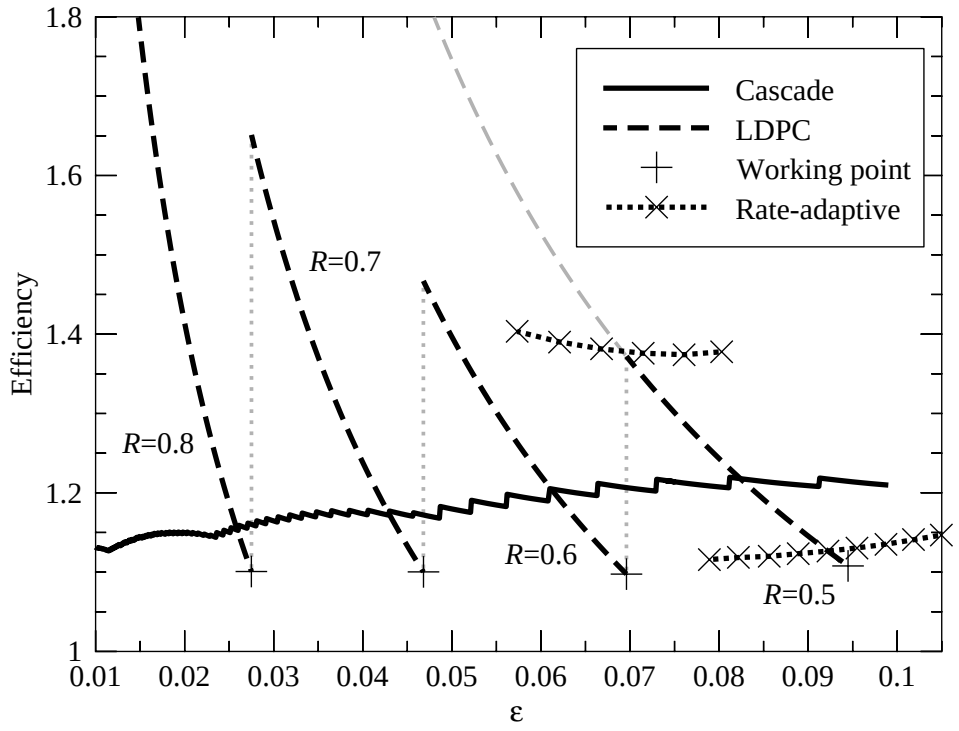

Figure 4.11: Simulated efficiency for a range of error rates ϵ using the rate-adaptive reconciliation protocol proposed here and other reconciliation approaches.

rate range $[0.055, 0.11]$.

Note how the saw tooth behavior is eliminated. For long LDPC codes and the δ value chosen the penalty is quite small —i.e. the efficiency is close to the working point, for which no rate-adaptive technique was used— and the rate-adaptive protocol allows to reconcile strings in all the range with an efficiency close to $f \approx 1.1$. The new protocol works at a much better efficiency than *Cascade*, that performs in all the tested range around $f \geq 1.2$. However, when using short length codes, the efficiency is worse than that of *Cascade*.

Chapter 5

Interactive Reconciliation with Low-Density Parity-Check Codes

This chapter is organized as follows. In Section 5.1 we introduce some feedback coding schemes, and their application for interactive reconciliation is considered. In Section 5.2 we propose an interactive version of the rate-adaptive protocol previously described that improves the average efficiency. We refer to this protocol as *blind*. Next, in Section 5.3 we analyze the average efficiency of the proposed interactive protocol from a theoretical perspective. Finally, in Section 5.4 we show some simulation results of this protocol using short-length LDPC codes.

5.1 Introduction

In classical communications common error detecting techniques, such as cyclic redundancy check (CRC) codes, are used to detect errors at the receiver. The receiver validates the transmission of any message and responds with an acknowledgment (ACK) if no errors were detected. Otherwise, the receiver requests the retransmission of corrupted messages with a negative acknowledgment (NAK). This typical communication scheme is known as automatic repeat request (ARQ).

Powerful error correcting techniques, such as LDPC codes, can be used together with ARQ schemes to improve the performance of classical communications. These schemes are known as *hybrid automatic repeat request* or hybrid ARQ (HARQ) [81, 93, 94]. They operate quite well for a narrow error rate range, since the error correcting code is chosen based on the channel parameter. Use of error correcting codes increase the probability of successful transmissions in HARQ-based communications, increasing thus the throughput of these communications.

Furthermore, a family of hybrid ARQ schemes was also proposed by adapting the coding rate to time varying channels. A channel is considered time varying when its parameter, e.g. the error rate, may vary in a known range. This new scheme is know as *incremental redundancy* HARQ or *type-II* hybrid ARQ. Most of rate-compatible solutions and rate-adaptive techniques for LDPC codes, such as puncturing, are analyzed for this type of HARQ scheme [89, 95–100].

Here we propose a variation of the rate-adaptive reconciliation method described above, Algorithm 4, based on the incremental redundancy HARQ idea. This original interactive reconciliation protocol is based on the simultaneous use of punctured and shortened symbols as described below.

5.2 Blind Reconciliation

In the adaptive-rate algorithm just outlined in Section 4.3, the proportion, δ, of punctured plus shortened symbols is held constant. This proportion is calculated after an error rate (channel parameter) estimation. Once is estimated the channel parameter, the only classical communication that is needed among Alice and Bob is one message from Alice to send the syndrome and the shortened information bits. This makes for a close to minimal interactivity protocol that is also highly efficient. Now, if we relax the interactivity condition and allow for a limited amount of communications, the panorama changes significantly.

Let us start by assuming again a value for δ covering the range of rates $[R_{\min}, R_{\max}]$ with the code with $R_{\min}$ able to correct words transmitted through the noisiest channel expected.

In a first message, Alice can include only the syndrome and no shortened bits, i.e. all the d symbols that can be either punctured or shortened, are punctured ($\pi = \delta$). If we look back at Figure 4.8, where we plot the behavior of FER as a function of ϵ using different proportions of punctured and shortened symbols, we can see that we are trying to correct errors with the code with the highest FER and highest rate, which is the one with $d = 200$.

If the reconciliation fails, no other information than the syndrome has been leaked, since punctured symbols do not disclose information. Alice can then reveal a set of the values of the previously punctured symbols. In this way the rate of the code is reduced, but the decoding success probability is increased. Returning to Figure 4.8, this is like moving along the dotted vertical line and changing the code with $p = 200$, $s = 0$ ($\mathcal{C}(2000-200, 1000)$) by the code with $p = 160$, $s = 40$ ($\mathcal{C}(2000-200, 1000-40)$) and using it to correct the same string. Only the previously punctured but now shortened symbols reveal extra information. The protocol runs on the same string by revealing more information on the values of previously punctured symbols till success is achieved (or all the symbols were shortened without syndrome matching and it fails), effectively by using at each iteration codes with lower rate and FER.

5.2.1 Blind Protocol

In Figure 5.1 we illustrate two iterations of the protocol in use to reconcile a string of length $m = 8$ using $d = 8$ extra symbols. It is also assumed that in every iteration $\Delta = 4$ symbols can be changed from punctured to shortened. In the first step, the m symbols are incremented with the $d = 8$ punctured ones to a total length of $n = m + d = 16$. At this point, the syndrome is calculated and the value sent to Bob. It is assumed that there is no syndrome match, hence the next iteration in which Δ of

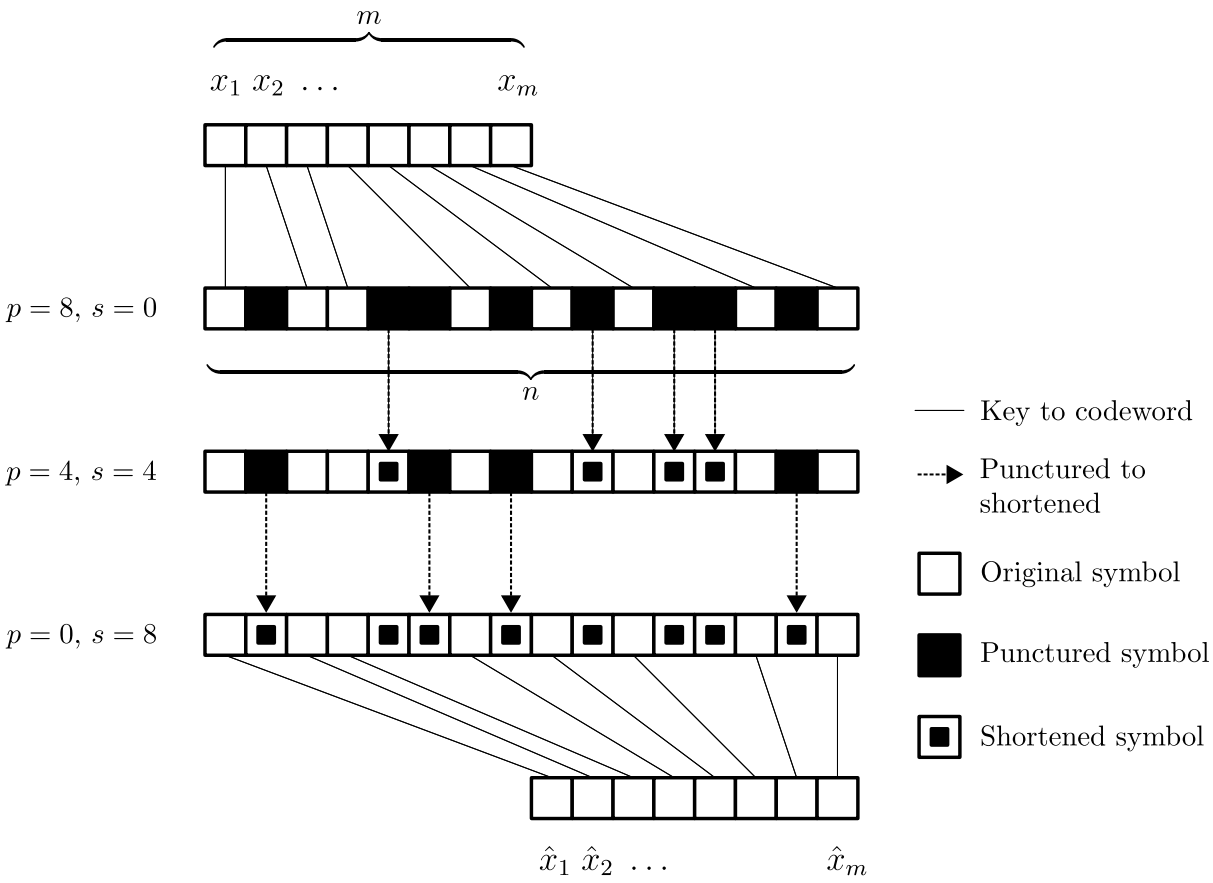

Figure 5.1: Blind reconciliation protocol schema for a three iteration version.

the previously punctured symbols change to shortened. This information is sent to Bob. Again, a no match is assumed and the protocol proceeds to its second iteration, where another Δ symbols are revealed changing from punctured to shortened. Here the protocol ends, no matter whether there is a syndrome match or not, since all the punctured symbols have changed to shortened. If the syndrome is validated, then it can be safely assume that the string $(x_1, x_2, \ldots, x_m)$ in Alice's side and $(\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m)$ in Bob's side are the same. Otherwise, the protocol fails for this string.

This whole procedure is done using the same base code and without needing an estimate for ϵ, hence the *blind* name. Only a rough estimate of the channel parameter is needed to design the base code. Note that this protocol requires some interactivity since, at each iteration in which there is no syndrome matching, a set of values for the shortened symbols must be communicated. As we show in the results section, a protocol with a very high average efficiency can be obtained using short codes and using only three iterations.

5.2.2 Interactive/Blind Protocol

We formally describe below the method for blind reconciliation outlined above in Algorithm 5. Note how there is no need of an a priori error estimate (except for the one implicitly embodied in the selection of the code $\mathcal{C}$) and a controlled amount of interactivity (t messages are exchanged at most).

5.3 Average Efficiency

It is supposed now that we can repeat the reconciliation process n times. We refer to each recurrence of the protocol as *iteration*. In each iteration the protocol adapts the information rate of our code in order to reconcile as many errors as possible by providing the minimum information. Starting from the highest coding rate, the proposed protocol is decreasing the information rate in each iteration. The reconciliation

Algorithm 5 Blind Reconciliation Protocol

Step 0: Set up.— Let $\mathcal{C}(n,k)$ be a code $\mathcal{C}$ that can correct noise up to $\epsilon_{\max}$ for some channel family. Let $\mathbf{x}$ and $\mathbf{y}$ be two strings that two parties Alice and Bob wish to reconcile in at most t iterations. Let $\mathbf{x}$ and $\mathbf{y}$ be of length m, with $m = n - d$, and every symbol of $\mathbf{y}$ the output of a memoryless channel characterized by $\epsilon < \epsilon_{\max}$. Alice and Bob set $s = 0$, $p = d$ and $\Delta = d/t$. For simplicity in the description we assume $\Delta \in \mathbf{N}$.

Step 1: Encoding.— Alice sends the syndrome in $\mathcal{C}$ of a word $\hat{\mathbf{x}}$ consisting on embedding $\mathbf{x}$ in a length n string and filling the remaining d positions with random symbols.

Step 2: Decoding.— Bob constructs the word $\hat{\mathbf{y}}$ consisting on the concatenation of $\mathbf{y}$ the received s symbols and p random symbols. If Bob recovers $\mathbf{x}$ he reports success and the protocol ends.

Step 3: Re-transmission.— If $d = s$ the protocol fails, else Alice sets $s = s + \Delta$, reveals Bob Δ symbols and they return to Step 2 and perform a new iteration.

process stops in the iteration where the syndrome is validated, and thus every error has been reconciled with high probability.

The average efficiency over the BSC(ϵ) for the proposed blind reconciliation protocol can be calculated as:

$$\hat{f}(\epsilon) = \sum_{i=1}^{n} \alpha_i f^{(i)} \tag{5.1}$$

where α_i is the fraction of codewords that have been corrected in the step i, such that $\sum_{i=1}^{n} \alpha_i = 1$. Using Eq. (3.7) we obtain the expression for the average efficiency over the BSC(ϵ):

$$\hat{f}_{\text{BSC}}(\epsilon) = \frac{1 - \sum_{i=1}^{n} \alpha_i r_i}{h(\epsilon)} = \frac{1 - \hat{R}}{h(\epsilon)} \tag{5.2}$$

where r_i is the information rate used during the i-th iteration, and $\hat{R}$ is the average rate used during the reconciliation process.

Let $F^{(i)}$ be the frame error rate (FER) when correcting with adapted rate r_i. Then the fraction of corrected codewords during the i-th iteration is given by:

$$\alpha_i = \frac{F^{(i-1)} - F^{(i)}}{1 - F^{(n)}} \tag{5.3}$$

where $F^{(0)} = 1$.

Now, the average rate can be expressed as:

$$\hat{R} = \sum_{i=1}^{n} \frac{F^{(i-1)} - F^{(i)}}{1 - F^{(n)}} \cdot r_i \tag{5.4}$$

The sum can be simplified as follows:

$$\sum_{i=1}^{n}(F^{(i-1)} - F^{(i)}) \cdot r_i = \tag{5.5}$$

$$= F^{(0)}r_1 - F^{(1)}r_1 + F^{(1)}r_2 - F^{(2)}r_2 + \ldots + F^{(n-1)}r_n - F^{(n)}r_n \tag{5.6}$$

$$= F^{(0)}r_1 + F^{(1)}(r_2 - r_1) + F^{(2)}(r_3 - r_2) + \ldots - F^{(n)}r_n \tag{5.7}$$

$$= r_1 - F^{(n)}r_n + \sum_{i=1}^{n-1} F^{(i)}(r_{i+1} - r_i) \tag{5.8}$$

And thus:

$$\hat{R} = \frac{r_1 - F^{(n)}r_n}{1 - F^{(n)}} + \sum_{i=1}^{n-1} \frac{F^{(i)}}{1 - F^{(n)}}(r_{i+1} - r_i) \tag{5.9}$$

Assuming that in every iteration we translate a constant proportion of punctured symbols to shortened symbols, the information rate used during the i-th iteration is given by:

$$r_i = \frac{R_0 - \sigma_i}{1 - \delta} \tag{5.10}$$

where R_0 is the coding rate of the mother code, and σ_i is the fraction of shortened symbols during the i-th iteration, such that $\sigma_1 = 0$ and $\sigma_n = \delta$. The rate increment between two consecutive iterations is also constant:

$$r_{i+1} - r_i = \frac{-\delta/n}{1 - \delta} \tag{5.11}$$

Let us define $\beta = \delta/(1 - \delta)$ and then $r_{i+1} - r_i = -\beta/n$. The average rate can be now written as:

$$\hat{R} = \frac{r_1 - F^{(n)}r_n}{1 - F^{(n)}} - \frac{\beta}{n}\sum_{i=1}^{n-1}\frac{F^{(i)}}{1 - F^{(n)}} \tag{5.12}$$

$$= r_1 + \frac{\beta}{1 - F^{(n)}}\left(F^{(n)} - \frac{1}{n}\sum_{i=1}^{n-1} F^{(i)}\right) \tag{5.13}$$

Where we have taken into account that in the first iteration every selected symbol is punctured, while in the last one every selected symbol is shortened, hence, the first and last coding rate, r_1 and r_n, are given by:

$$r_1 = \frac{R_0}{1-\delta}; \quad r_n = \frac{R_0 - \delta}{1-\delta} = r_1 - \beta \tag{5.14}$$

Note that in the rate-adaptive approach a typical value for the frame error rate in a reliable reconciliation is 10^{-3}; i.e. we can then neglect the last contribution for the FER ($F^{(n)} \approx 0$), and thus an approximate average rate is given by:

$$\hat{R} \approx r_1 - \frac{\beta}{n} \sum_{i=1}^{n-1} F^{(i)} \tag{5.15}$$

An approach for estimating the frame error rate of a linear code is described in Appendix B. Using this approach we can accurately estimate the frame error rate of a finite length LDPC code without having to perform computer simulations. We use then Eq. (5.2) and Eq. (5.15) to calculate the average efficiency for the interactive reconciliation protocol analyzed here.

Figure 5.2 shows the estimated efficiency for a short-length LDPC code of 2×10^3 bits in the error rate range $\epsilon \in [0.04, 0.08]$. The rate adaptive protocol of Ref. [34] (code with $p = 200$ in Figure 4.8) is compared to the blind protocol for short codes. In the figure it is shown the average efficiency for different numbers of iterations. These curves are compared to the efficiency of the rate-adaptive solution described in Section 4.2.3. Rate-adaptive and blind protocol coincide for the starting base code (the code with $p = 200$ in Figure 4.8), point marked A in the figure. When the error rate increases, the blind protocol adapts its behavior to the new channel parameter. Depending on the number of iterations allowed, which limits its maximum interactivity, the protocol is shown to approach the threshold.

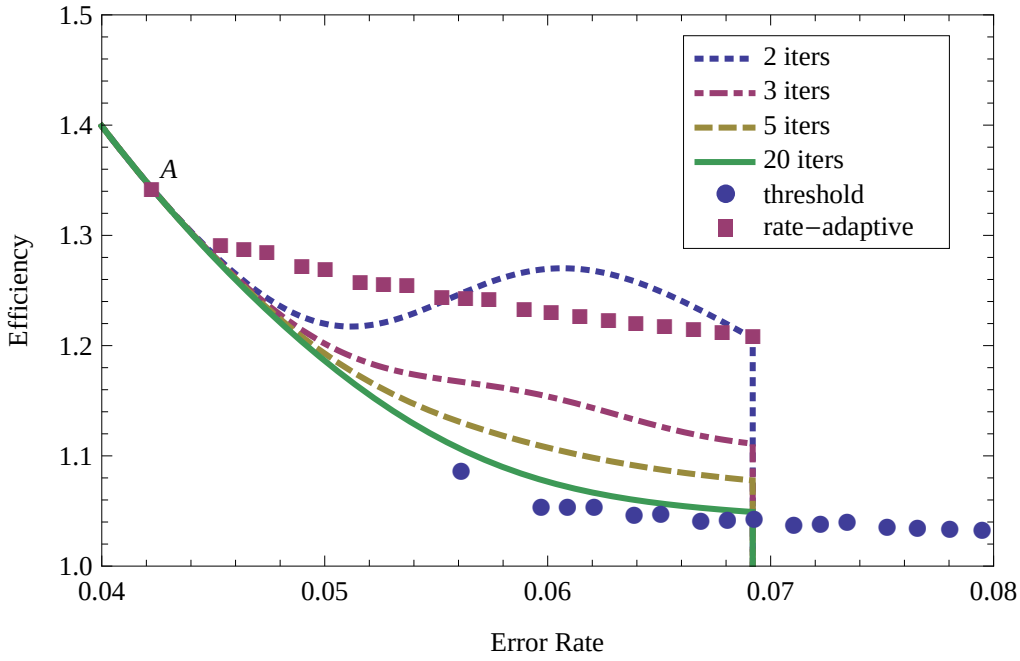

Figure 5.2: Average efficiency of the proposed blind reconciliation protocol for several maximum number of iterations.

5.4 Simulation Results

Simulation results have been computed to compare the protocol proposed in Ref. [34] with the interactive version proposed here, but using short-length LDPC codes. These simulations were performed for two error rate ranges, one with low error rates and other with high ones, over the binary symmetric channel. An LDPC decoder based on the sum-product algorithm with 200 maximum decoding iterations and serial schedule was used.

New families of LDPC codes were designed (see Appendix C), and the code graphs were constructed using the modified version of the progressive edge-growth algorithm proposed here and the improved version proposed in Ref. [66] in order to reduce the residual error in the error floor region. Punctured symbols were selected according to a computed pattern for intentional puncturing as described in Ref. [35], while shortened symbols were randomly selected.

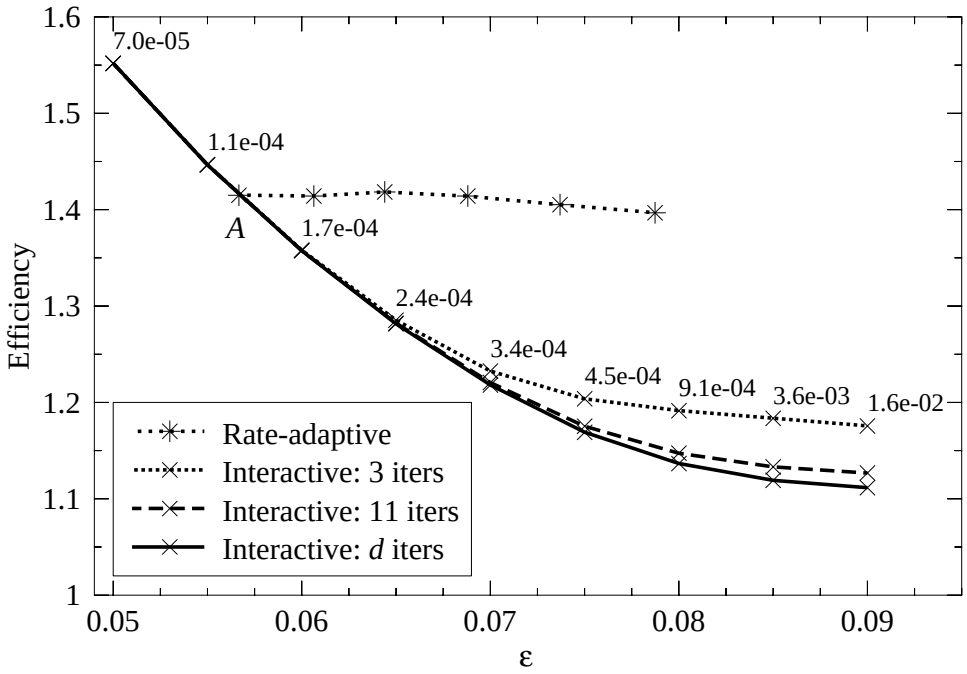

Figure 5.3: Simulated efficiency for the rate-adaptive and the interactive reconciliation protocols in the high error rate region.

Figures 5.3 and 5.4 show the efficiency, as defined in Eq. (3.2), of the rate-adaptive protocol proposed in Ref. [34], and the average efficiency of the interactive version proposed here. A lightweight interactive protocol, with 3 iterations maximum, is compared with the maximally interactive version where in every iteration only one punctured symbol becomes a shortened one. Simulations have been computed using an LDPC code of 2×10^3 bits length and coding rate $R = 0.5$ with $\delta = 0.1$. As expected, a behavior similar to Figure 5.2 is found, where the efficiency coincides for the rate-adaptive and interactive approaches for error rates below A. It is shown how the efficiency improves with interactivity (more iterations) and also with the error rate.

In both figures, the average frame error rate (FER) is printed for each point of the version with a maximum of three iterations. In Figure 5.3 it was used LDPC codes constructed with the modified PEG algorithm proposed here (see Section 3.3),

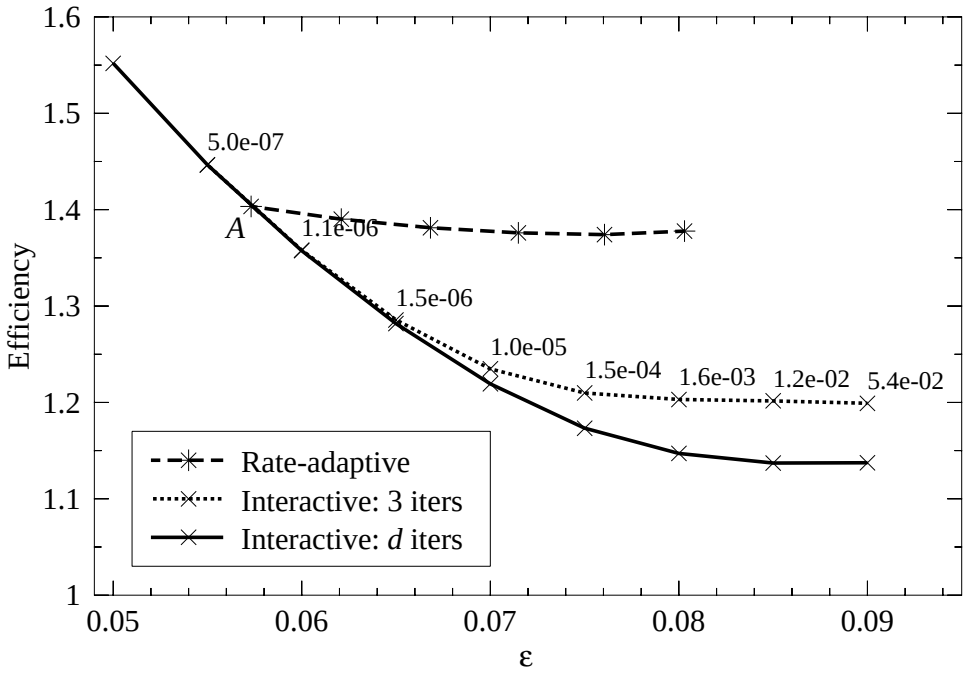

Figure 5.4: Simulated efficiency for the rate-adaptive and the interactive reconciliation protocols in the high error rate region.

while in Figure 5.4 LDPC codes were constructed using the improved PEG algorithm proposed in Ref. [66]. Comparing the values of the average FER printed in the three iterative version of both figures, it can be appreciated the impact of the error floor in the original LDPC code.

Figure 5.5 also compares the efficiency and the average efficiency for the reconciliation protocols analyzed here. In this case the efficiency is studied in the low error rate range in QKD. An LDPC code of 10^4 bits length and coding rate $R = 0.8$ is used. Due to this high coding rate, only the 5% of symbols were selected for puncturing and shortening, also using the aforementioned strategy for intentional puncturing. The figure shows that the average efficiency quickly improves with the blind protocol, even when using only three iterations.

If we try to increase the range of error rates covered, we can increase the proportion of punctured and shortened symbols (see Eq. (4.4)). The results are shown

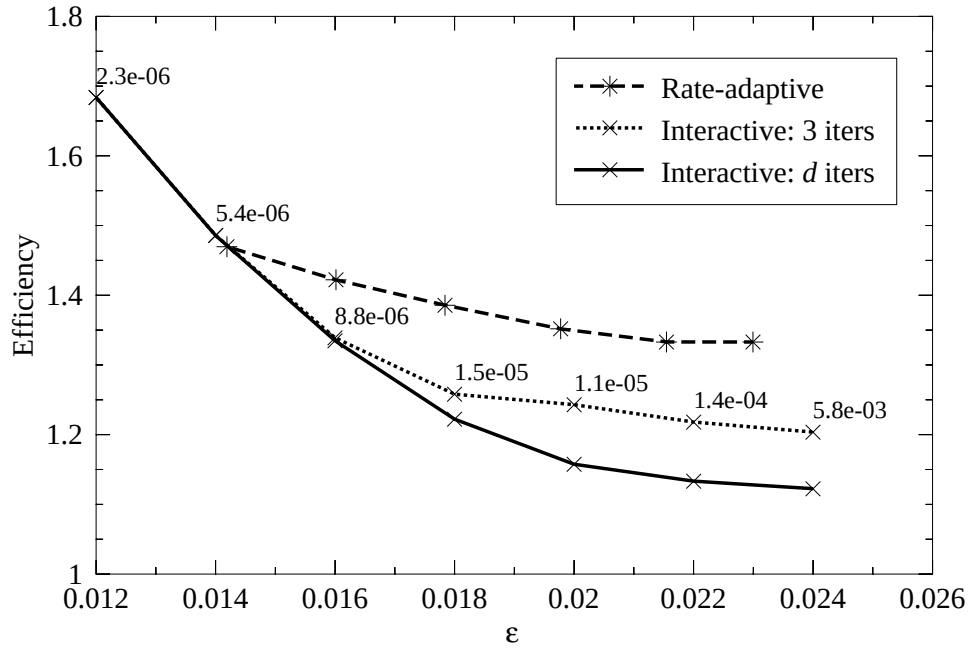

Figure 5.5: Simulated efficiency for the rate-adaptive and interactive protocols in the low error rate region.

in Figure 5.6, where the proportion is set to 8%, the maximum achievable value following the intentional puncturing proposal described in Ref. [35]. We can observe that for a fixed number of iterations the efficiency is worse (compared to Figure 5.5), see the dotted line with a maximum of three iterations; though the efficiency for the maximum d iterations, as d is higher, improves.

In order to understand the behavior of the curve for the interactive version with a maximum of three iterations, the figure shows the efficiency of using LDPC codes with the coding rates associated with each iteration:

$$r_1 = \frac{R_0}{1-\delta}; \quad r_2 = \frac{R_0 - \delta/2}{1-\delta}; \quad r_3 = \frac{R_0 - \delta}{1-\delta} \tag{5.16}$$

The increase in efficiency with the number of iterations opens the possibility of having both, high efficiency and high throughput.

The new generation of QKD systems are approaching speeds for encryption close to 1 Gbps [101]. Implementing real time error correction to provide secret keys at this

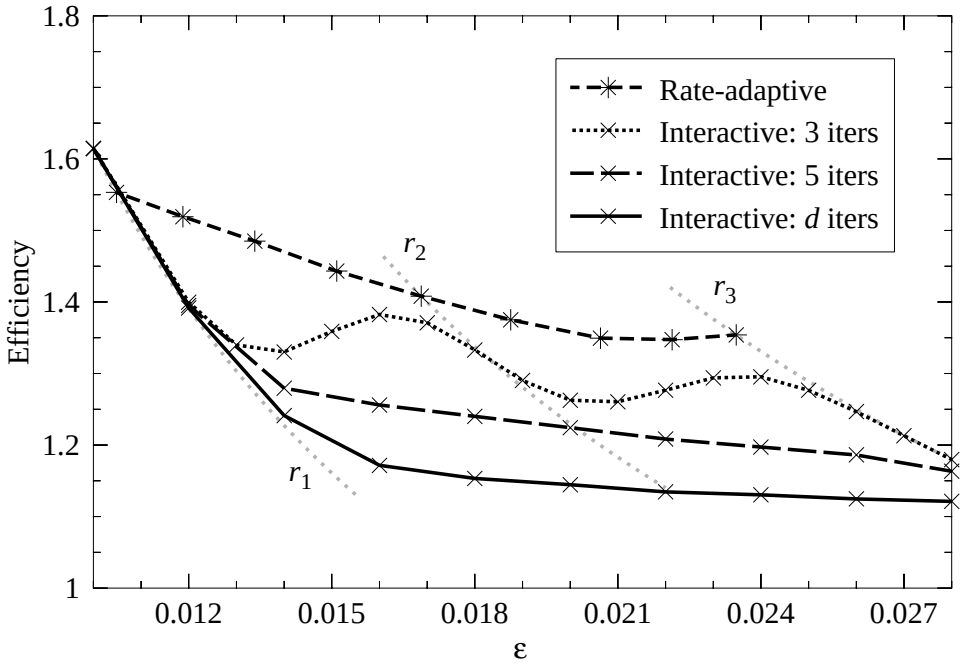

Figure 5.6: Simulated efficiency for the rate-adaptive and interactive protocols in the low error rate region but using a high proportion of punctured symbols.

speed is a challenging problem where a high throughput procedure with minimal communications is needed. Using *Cascade* under these constraints is unfeasible unless an extremely low latency network is used. Short-length LDPC codes have been implemented in hardware for other purposes, like wireless networks [102], where they have demonstrated to be an excellent solution.

In the QKD case, to use LDPC codes required to have codes designed for different error rates, thus making the process more complex and memory constrained. With the protocol presented here, the error estimation phase is not needed. The procedure can start directly and, if it fails, allowing a few iterations increases considerably the success probability. The price to pay is an extra message per failure. As shown in Figure 5.2 and Figures 5.4-5.6, the process converges quickly and only a few iterations are needed to increase the reconciliation efficiency significantly.

In a hardware implementation, the iterations are easily realized just by copying the same functional decoder block as many times as the number of desired iterations. The string to be reconciled would start the i-th iteration in the first hardware block. If the decoding fails, the next hardware block would continue processing in a pipeline fashion, since computation and communication can be arranged in a way such that the disclosed symbols would arrive packed in the same message than the syndrome of the following strings to reconcile. The new syndrome would start being processed in the first hardware block while the second would continue working on the second iteration on the previous string. This pipeline can increase the efficiency while maintaining a high and —mostly— constant throughput at the expense of some extra hardware.

Chapter 6

Reliable Reconciliation and Undetected Error Probability

Implicit in everything discussed so far is the reliability of a reconciled string (the key), or what is the same thing, it has not been previously commented how often the reconciliation process can end up with *undetected errors*. In this chapter, we study this undetected error probability and its impact on the reliability of a secret-key reconciliation process using LDPC codes. In this regard, everything discussed below is presented from the perspective of information reconciliation rather than from the traditional coding context.

Before proceeding, it should be noted that all simulations in previous chapters were computed taking into account both contributions to the error rate, detected and undetected errors.

6.1 Introduction

As its name suggests, an *undetected error* in the information reconciliation context is any discrepancy on the reconciled string which is unknown by the communication parties. Therefore, due to undetected errors it may occurs that two parties do not

share any secret after the reconciliation process, since their reconciled strings are not identical, and both parties are unaware of this error.

In practice, undetected errors can be avoided using a separate error detection scheme. Cyclic redundancy check (CRC) codes, checksums and hash functions are probably the most widespread methods for detecting errors in digital communications. However, even in such codes it remains a probability that some errors are not detected [103]. The analysis of this error detection probability of a given parity-check matrix is a classical problem in coding theory [104, 105]. Moreover, the analysis of these undetected errors are of special interest in practice when working with *feedback error correction* schemes, for instance based on ARQ or hybrid ARQ, such as the previously proposed *blind* algorithm (see Section 5.2). In this work, we focus on the study of this undetected error probability using LDPC codes, taking into account that these codes have the inherent capability to reconcile as well as detect errors by validating the final calculated syndrome. Error detection is accomplished by verifying satisfaction of all check node constraints at the end of every iteration.

Undetected errors in LDPC codes were originally analyzed by MacKay in Refs. [51, 106, 107]. As commented by the author, they reveal some properties of a code, such as minimum distance and weight distributions. In Ref. [108] Wadayama shows this relationship following a probabilistic approach of randomly generated codes:

> *"Since the undetected error probability can be expressed as a linear combination of weight distribution of a code, there are natural connection between expectation of weight distribution and expectation of undetected error probability."*

Let P_d to be the detected error probability, and P_u to be the undetected error probability. The total probability of codeword error is then given by $P_w = P_d + P_u$. Let H be the parity-check matrix of a code $\mathcal{C}$. An undetected error occurs when $H\mathbf{e}^t = \mathbf{0}$ and $\mathbf{e} \neq \mathbf{0}$, it means that the error vector is a codeword $\mathbf{e} \in \mathcal{C}$. Assuming that no decoding technique is used —i.e. the linear code $\mathcal{C}$ is used only to detect errors in the transmitted codeword by validating the syndrome—, the undetected

error probability, P_u, in the binary symmetric channel with crossover probability ϵ is then given by:

$$P_u = \sum_{\mathbf{e}\in\mathcal{C},\mathbf{e}\neq\mathbf{0}} \epsilon^{w(\mathbf{e})}(1-\epsilon)^{n-w(\mathbf{e})} \tag{6.1}$$

where $\mathbf{e}$ denotes an error vector, and $w(\mathbf{e})$ denotes the Hamming weight of this error vector.

However, the probability of an undetected error depends on both the code $\mathcal{C}$ and the decoding algorithm $\mathcal{D}$ used to reconcile the key, and thus P_u as defined in Eq. (6.1) is only a lower bound of the undetected error rate. Henceforth, we assume the use of LDPC codes and iterative message-passing algorithms for decoding.

6.2 Undetected Errors in LDPC Codes using Iterative Message-Passing Decoding

In an iterative message-passing algorithm, such as the sum-product algorithm, an error is said to be detected if the decoding process concludes after completing the maximum number of iterations without finding a codeword. Error detection is accomplished by verifying satisfaction of all check node constraints at the end of iterations. On the other hand, an undetected error occurs when the decoder finds a codeword $\hat{\mathbf{x}}$ satisfying $H\hat{\mathbf{x}}^t = \mathbf{z}$, i.e. the syndrome matches, but that codeword does not correspond with the transmitted one $\mathbf{x}$, $\mathbf{x} = \hat{\mathbf{x}}$.

It should be noted that different strategies for the original flooding scheduling demonstrate that undetected errors can be avoided using a particular decoding strategy [62]. Here, we show different strategies that can be used to reduce the probability of undetected errors to acceptable levels using iterative LDPC decoding, by means of a simple modification to the iterative decoder structure without redesigning the code.

6.2.1 Bounded Iterative Decoding

An original method to reduce the undetected error rate of short-length LDPC codes is proposed in Ref. [109]. The proposed method is based on calculating the Euclidean angle between the received word and the decoded codeword at the receiver. This codeword is then rejected if the calculated angle is greater than a threshold. With a judicious choice of the maximum decoding angle, the undetected error rate can be reduced while the overall error rate increases modestly. This modification in the decoding algorithm is called *bounded angle iterative decoding* by the authors. An upper bound on the performance of the proposed method is later analyzed in Ref. [110] for maximum-likelihood decoders.

Let $\mathbf{x}$ be the received word, and let $\mathbf{c}_i$ be the decoded codeword at the receiver (i.e. assuming that a valid codeword was found during the decoding process). The Euclidean angle θ_i is then calculated as:

$$\theta_i = \cos^{-1}\left(\frac{\langle \mathbf{x}, \mathbf{c}_i \rangle}{\|\mathbf{x}\|\|\mathbf{c}_i\|}\right) \tag{6.2}$$

The decoded codeword is only accepted if this angle θ_i is lower that a maximum decoding angle $\theta_{\max}$. A similar method is also commented by the authors where the Euclidean angle is replaced by the Euclidean distance.

6.2.2 Decoding Quasi-Cyclic Codes

An undesirable effect of digital communications is the insertion and deletion of symbols mainly due to synchronization errors. Some previous work in communication theory deal with this problem when using LDPC codes, e.g. see Ref. [107]. Symbol blocks are entirely shifted by the insertion or deletion of symbols, and it may result in decoding a wrong codeword. This effect, referred as *symbol slip*, is especially critical when using quasi-cyclic LDPC codes as analyzed in Ref. [111] since these decoding errors are the cause of undetected errors.

We discuss this problem here as quasi-cyclic LDPC codes are very common in standards and hardware implementations, e.g. see [102]. However, it should be noted that this kind of synchronization errors are not usual in quantum key distribution —according to the approach adopted in this work—, since each symbol (bits of the key) to reconcile is well-synchronized after the sifting procedure.

In Ref. [111], the authors analyze why symbol slips can produce undetected errors when using quasi-cyclic LDPC codes, and they propose several methods to prevent these errors. A simple way to avoid them is based on the use of a *pseudo-random number sequence*. On one side, the emitter performs an XOR operation between the codeword and a pseudo-random number sequence. On the other side, the receiver performs the same operation undoing the emitter changes and avoiding the symbol slip in quasi-cyclic codes.

6.3 Simulation Results

Simulation results over the binary symmetric channel were computed to analyze the undetected error probability of 2×10^3 bits length LDPC codes and coding rate one half. All simulations were computed using a sum-product algorithm, with serial scheduling, but different decoding parameters. For instance, simulation results are compared for a different number of maximum decoding iterations, and with or without codeword validation after each iteration. LDPC codes analyzed here were constructed using a PEG-based algorithm.

Figure 6.1 shows the frame error rate, P_w, and the undetected error probability, P_u, of a 2×10^3 bits length LDPC code with coding rate one half. Results are shown for a range of crossover probabilities ϵ. Simulation results for 20 and 200 maximum decoding iterations are compared. The figure shows that the undetected error probability remains low even for high crossover probabilities, a behavior that is crucial to the success of the previously proposed *blind* protocol.

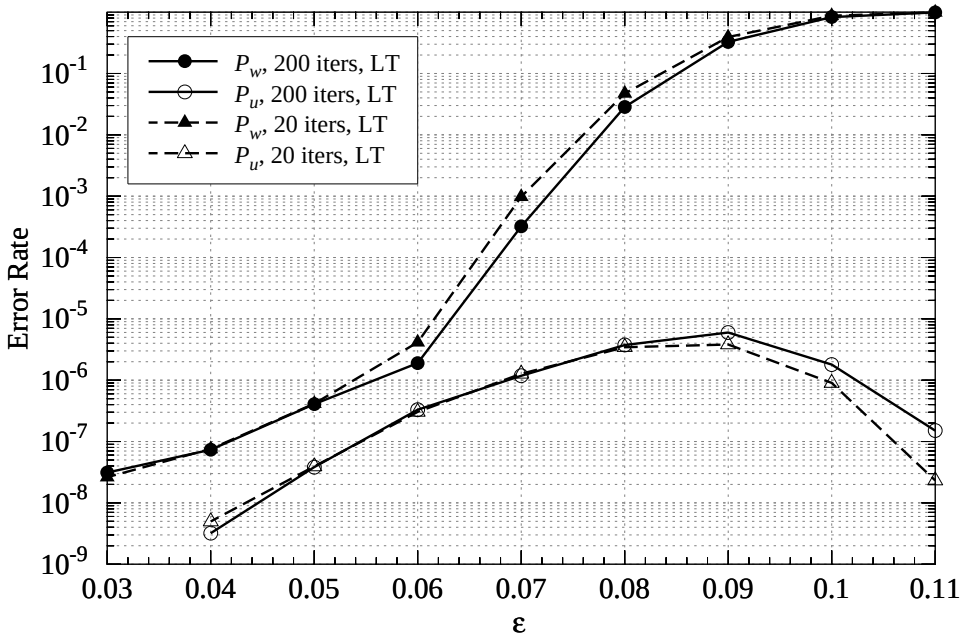

Figure 6.1: Performance and undetected error rate over the BSC with crossover probability ϵ of a PEG-based LDPC code.

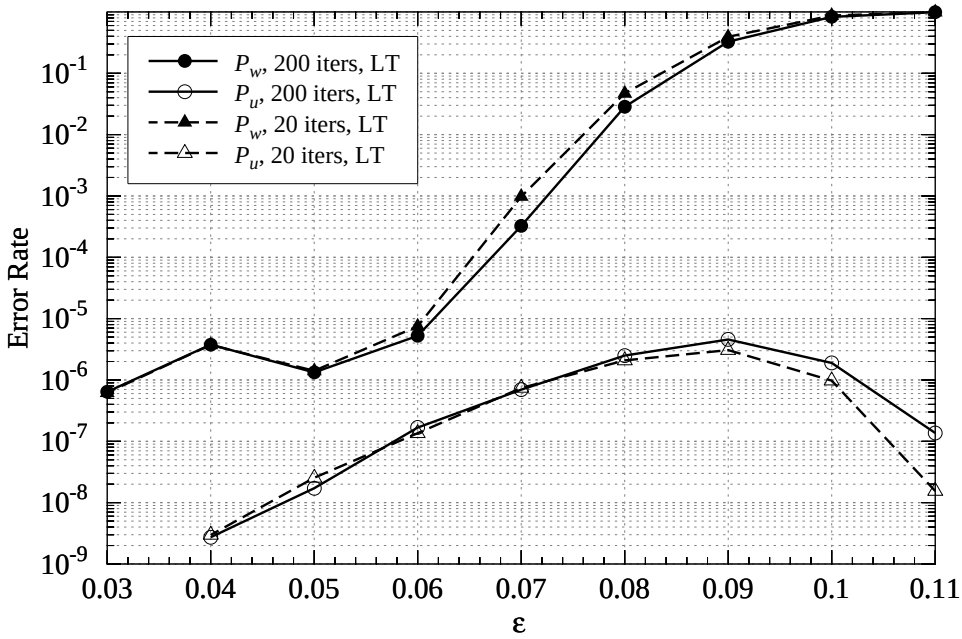

Figure 6.2: Performance and undetected error rate over the BSC with crossover probability ϵ of a PEG-based LDPC code.

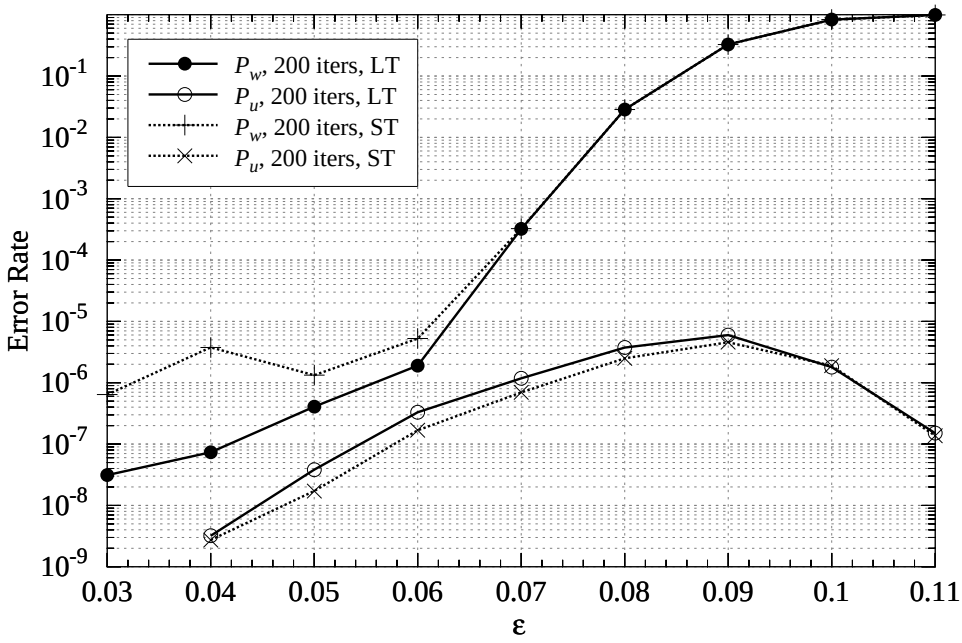

Figure 6.3: Performance and undetected error rate over the BSC with crossover probability ϵ using look-up tables of different sizes.

The LDPC code previously used in Figure 6.1 is again analyzed with a modified decoder. Figure 6.2 shows the performance and undetected error rate using an LDPC decoder with an smaller look-up table (ST), a probably desirable behavior in hardware or fast decoding implementations. Simulation results for 20 and 200 maximum decoding iterations are compared. The figure shows an strange behavior in the error floor region that has to be considered for the average efficiency of the interactive protocol proposed in Section 5.2.2. At first glance, this strange behavior does not cause a greater probability of undetected errors.

In Figure 6.3 we compare FER and undetected error rate of both LDPC decoders, i.e. using look-up tables of different sizes. Simulation results for two look-up tables of 10^6 and 10^7 values are compared. A number of 200 maximum decoding iterations is used in both simulations. Despite the noticeable difference in performance in the error floor region, the undetected error probability does not show any difference.

Part III

Concluding Remarks

Chapter 7

Conclusions

In this work we have shown how LDPC codes can be used to improve the performance of classical methods for information reconciliation. Based on the Wynner's idea of syndrome source coding, we studied how LDPC codes can be applied to the problem of information reconciliation. We have developed a protocol able to modulate the information rate of LDPC codes. The ability to adapt error correcting codes to different error rates is crucial to minimize the amount of information disclosed during the secret-key reconciliation. We also introduce the concept of efficiency as a measure for the quality of the reconciliation process. In the QKD context the efficiency is specially important, whenever it is used in long distance links or in noisy environments such as those arising in shared optical networks [27, 28, 112–115]. In these demanding environments high efficiency is necessary to distill a secret-key when maximum distance or absorption budget is required.

In practical QKD, throughput is also of paramount importance. Here, we analyzed the performance of LDPC codes using incremental redundancy HARQ schemes. These schemes are also used to improve the throughput in classical communications. We show how a similar scheme can be used to improve the average efficiency of an information reconciliation procedure. The results show that the average efficiency

may even be improved using short-length LDPC codes[1] and an slightly interactive protocol which is also good for a high throughput HW implementation. For instance, it is shown that a protocol with very low interactivity —i.e. with a maximum of three iterations— improves considerably the efficiency in high and low error rate regimes. This protocol can be easily implemented in hardware, since it uses short-length LDPC codes, and it can be also pipelined for a very high throughput of reconciled key.

The proposed interactive protocol also improves the final secret-key length. Up to now, in a QKD protocol part of the raw key has to be disclosed in order to estimate the channel parameter, i.e. the error rate. The new protocol does not need an a priori error rate estimate to work; it adapts automatically —hence the name *blind*—. As a result, the protocol also provides the *exact* error rate corrected, bringing interesting implications for the security analysis of finite length keys.

[1]Between 2×10^3 and 10^4 bits long.

Chapter 8

Future Work

QKD is an evolving and dynamic field but also with recognized long term goals. Two of the most cited ones: quantum repeaters networks and device independent QKD are among the last ones. These breakthroughs would allow a worldwide quantum network, either for security or computation purposes, but much basic research remains to be done. In the more technologically oriented short term view, one of the most important challenges of current developments in QKD is to ease the barrier among the lab prototypes and its actual deployment as industrial grade appliances. Towards this goal, significant advances should be pursued in: (1) the integration of QKD devices in commercial optical networks, (2) the development of high performance devices with high secret-key rates, and (3) the construction of more affordable devices. Moreover, we must also emphasize that all these challenges are faced from the standardization and miniaturization of QKD devices.

The research presented in this work was focused on the development of efficient protocols for reconciliation, that are also easy to implement in hardware and embedded devices, in line with the goals mentioned above. LDPC codes where selected for this work since they are capacity achieving for some communication channels, and there exist efficient decoding techniques that can be implemented in common hardware devices, such as the well-known field-programmable gate array (FPGA) or

the also common very-large-scale integration (VLSI) circuits. Short-length codes, for instance, were simulated with this purpose.

We constructed LDPC codes from pre-designed ensembles of codes using PEG-based algorithms. As a future line of work, it may be interesting to analyze other codes, such as quasi-cyclic codes, for which there exist even more efficient decoding techniques. A tradeoff between efficiency and throughput must also be analyzed to maximize the final secret-key rate.

Also, towards the final application of the proposed protocol together with the privacy amplification phase, the effect of the protocol on the finite-key effects must still be studied, mainly when using short-length LDPC codes.

Bibliography

[1] C. H. Bennett and G. Brassard, "Quantum Cryptography: Public Key Distribution and Coin Tossing," in *IEEE International Conference on Computers, Systems, and Signal Processing*, 1984, pp. 175–179.

[2] S. Wiesner, "Conjugate Coding," *SIGACT News*, vol. 15, no. 1, pp. 78–88, 1983.

[3] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, "Quantum Cryptography," *Reviews of Modern Physics*, vol. 74, no. 1, pp. 145–195, March 2002.

[4] C. H. Bennett, F. Bessette, G. Brassard, L. Salvail, and J. Smolin, "Experimental Quantum Cryptography," *Journal of Cryptology*, vol. 5, no. 1, pp. 3–28, January 1992.

[5] G. Brassard and L. Salvail, "Secret-Key Reconciliation by Public Discussion," in *Eurocrypt'93, Workshop on the theory and application of cryptographic techniques on Advances in cryptology*, ser. Lecture Notes in Computer Science, vol. 765. Springer-Verlag, 1994, pp. 410–423.

[6] C. H. Bennett, G. Brassard, and J.-M. Roberts, "Privacy Amplification by Public Discussion," *SIAM Journal on Computing*, vol. 17, no. 2, pp. 210–229, 1988.

[7] C. H. Bennett, G. Brassard, C. Crepeau, and U. M. Maurer, "Generalized Privacy Amplification," *IEEE Transactions on Information Theory*, vol. 41, no. 6, pp. 1915–1923, November 1995.

[8] U. M. Maurer, "Secret Key Agreement by Public Discussion from Common Information," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 733–742, May 1993.

[9] M. Van Dijk and H. Van Tilborg, "The Art of Distilling," in *ITW 1998, Information Theory Workshop*, June 1998, pp. 158–159.

[10] G. Van Assche, "Information-Theoretic Aspects of Quantum Key Distribution," Université Libre de Bruxelles, 2005, ph. D. Thesis.

[11] C. Crépeau, "Réconciliation et Distillation Publiques de Secret," Unpublished Manuscript, 1995, available at http://www.cs.mcgill.ca/˜crepeau/.

[12] M. Van Dijk and A. Koppelaar, "High Rate Reconciliation," in *ISIT 1997, IEEE International Symposium on Information Theory*, June 1997, p. 92.

[13] K. Yamazaki, M. Osaki, and O. Hirota, "On Reconciliation of Discrepant Sequences Shared Through Quantum Mechanical Channels," in *Information Security*, ser. Lecture Notes in Computer Science, E. Okamoto, G. Davida, and M. Mambo, Eds. Springer Berlin / Heidelberg, 1998, vol. 1396, pp. 345–356.

[14] T. Sugimoto and K. Yamazaki, "A Study on Secret Key Reconciliation Protocol "Cascade"," *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences*, vol. E83-A, no. 10, pp. 1987–1991, October 2000.

[15] E. Furukawa and K. Yamazaki, "Application of Existing Perfect Code to Secret Key Reconciliation," in *ISCIT 2001, International Symposium on Communications and Information Technologies*, 2001, pp. 397–400.

[16] A. Yamamura and H. Ishizuka, "Error Detection and Authentication in Quantum Key Distribution," in *Information Security and Privacy*, ser. Lecture Notes in Computer Science, 2001, vol. 2119, pp. 260–273.

[17] K. Chen, "Reconciliation by Public Discussion: Throughput and Residue Error Rate," Unpublished Manuscript, 2001.

[18] S. Liu, "Information-Theoretic Secret Key Agreement," Ph.D. dissertation, Technische Universiteit Eindhoven, 2002.

[19] S. Liu, H. C. A. Van Tilborg, and M. Van Dijk, "A Practical Protocol for Advantage Distillation and Information Reconciliation," *Designs, Codes and Cryptography*, vol. 30, no. 1, pp. 39–62, 2003.

[20] A. Nakassis, J. C. Bienfang, and C. J. Williams, "Expeditious Reconciliation for Practical Quantum Key Distribution," in *SPIE Conference Proceedings*, vol. 5436, no. 1, 2004, pp. 28–35.

[21] W. T. Buttler, S. K. Lamoreaux, J. R. Torgerson, G. H. Nickel, C. H. Donahue, and C. G. Peterson, "Fast, efficient error reconciliation for quantum cryptography," *Physical Review A*, vol. 67, no. 5, p. 052303, May 2003.

[22] C. Elliott, D. Pearson, and G. Troxel, "Quantum Cryptography in Practice," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '03, 2003, pp. 227–238.

[23] D. Pearson, "High-speed QKD Reconciliation using Forward Error Correction," in *7th International Conference on Quantum Communication, Measurement and Computing*, vol. 734, no. 1, November 2004, pp. 299–302.

[24] C. Elliott, A. Colvin, D. Pearson, O. Pikalo, J. Schlafer, and H. Yeh, "Current status of the DARPA Quantum Network," arXiv:quant-ph/0503058v2.

[25] T. J. Richardson and R. L. Urbanke, *Modern Coding Theory*. New York, NY, USA: Cambridge University Press, 2008.

[26] D. Elkouss, A. Leverrier, R. Alleaume, and J. J. Boutros, "Efficient reconciliation protocol for discrete-variable quantum key distribution," in *ISIT 2009, IEEE International Symposium on Information Theory*, July 2009, pp. 1879–1883.

[27] D. Lancho, J. Martinez, D. Elkouss, M. Soto, and V. Martin, "QKD in Standard Optical Telecommunications Networks," in *QuantumComm 2009, First International Conference on Quantum Communication and Quantum Networking*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 36, 2009, pp. 142–149.

[28] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka, K. Yoshino, Y. Nambu, S. Takahashi, A. Tajima, A. Tomita, T. Domeki, T. Hasegawa, Y. Sakai, H. Kobayashi, T. Asai, K. Shimizu, T. Tokura, T. Tsurumaru, M. Matsui, T. Honjo, K. Tamaki, H. Takesue, Y. Tokura, J. F. Dynes, A. R. Dixon, A. W. Sharpe, Z. L. Yuan, A. J. Shields, S. Uchikoga, M. Legré, S. Robyr, P. Trinkler, L. Monat, J.-B. Page, G. Ribordy, A. Poppe, A. Allacher, O. Maurhart, T. Länger, M. Peev, and A. Zeilinger, "Field test of quantum key distribution in the Tokyo QKD Network," *Optics Express*, vol. 19, no. 11, pp. 10 387–10 409, May 2011.

[29] R. Gallager, "Low-Density Parity-Check Codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, January 1962.

[30] R. G. Gallager, *Low-density parity-check codes*. MIT Press, Cambridge,, 1963.

[31] T. J. Richardson and R. L. Urbanke, "The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, February 2001.

[32] D. Elkouss, J. Martínez, D. Lancho, and V. Martín, "Rate Compatible Protocol for Information Reconciliation: An application to QKD," in *ITW 2010, IEEE Information Theory Workshop*, January 2010, pp. 145–149.

[33] J. Martinez-Mateo, D. Elkouss, and V. Martin, "Interactive Reconciliation with Low-Density Parity-Check Codes," in *Proceedings of the 6th International Symposium on Turbo Codes & Iterative Information Processing*, September 2010, pp. 270–274.

[34] D. Elkouss, J. Martinez-Mateo, and V. Martin, "Information Reconciliation for Quantum Key Distribution," *Quantum Information and Computation*, vol. 11, no. 3&4, pp. 226–238, April-May 2011.

[35] ——, "Untainted Puncturing for Irregular Low-Density Parity-Check Codes," *IEEE Communications Letters*, 2011, submitted.

[36] J. Martinez-Mateo, D. Elkouss, and V. Martin, "Blind Reconciliation," *Quantum Information & Computation*, 2011, submitted to.

[37] D. Elkouss, "Information Reconciliation Methods in Secret-Key Distribution," Ph. D. Thesis, Universidad Politéctica de Madrid, 2011, submitted for the degree of Doctor of Philosophy.

[38] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, July and October 1948.

[39] ——, "Communication Theory of Secrecy Systems," *The Bell System Technical Journal*, vol. 28, pp. 656–715, 1949.

[40] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley-Interscience, 1991.

[41] D. J. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003, available at http://www.inference.phy.cam.ac.uk/mackay/itila/.

[42] G. Van Assche, *Quantum Cryptography and Secret-Key Distillation*. Cambridge University Pres, 2006.

[43] S. Loepp and W. K. Wootters, *Protecting Information: From Classical Error Correction to Quantum Cryptography*. Cambridge University Press, 2006.

[44] R. Tanner, "A Recursive Approach to Low Complexity Codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, September 1981.

[45] D. Slepian and J. Wolf, "Noiseless Coding of Correlated Information Sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, July 1973.

[46] A. D. Wyner, "Recent Results in the Shannon Theory," *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 2–10, January 1974.

[47] A. Liveris, Zixiang Xiong, and C. Georghiades, "Compression of Binary Sources With Side Information at the Decoder Using LDPC Codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, October 2002.

[48] J. Muramatsu, T. Uyematsu, and T. Wadayama, "Low-Density Parity-Check Matrices for Coding of Correlated Sources," *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3645–3654, October 2005.

[49] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1276, 2002.

[50] D. J. MacKay and R. Neal, "Near Shannon Limit Performance of Low Density Parity Check Codes," *Electronics Letters*, vol. 32, no. 18, pp. 1645–1646, August 1996.

[51] D. J. MacKay, "Good Error-Correcting Codes Based on Very Sparse Matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, March 1999.

[52] T. J. Richardson, M. Shokrollahi, and R. L. Urbanke, "Design of Capacity-Approaching Irregular Low-Density Parity-Check Codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, February 2001.

[53] Sae-Young Chung, T. J. Richardson, and R. L. Urbanke, "Analysis of Sum-Product Decoding of Low-Density Parity-Check Codes Using a Gaussian Approximation," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 657–670, February 2001.

[54] Sae-Young Chung, J. Forney, G.D., T. J. Richardson, and R. L. Urbanke, "On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, February 2001.

[55] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Improved Low-Density Parity-Check Codes Using Irregular Graphs," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 585–598, February 2001.

[56] K. Kasai, R. Matsumoto, and K. Sakaniwa, "Information Reconciliation for QKD with Rate-Compatible Non-Binary LDPC Codes," in *ISITA 2010, International Symposium on Information Theory and its Applications*, October 2010, pp. 922–927.

[57] R. Matsumoto, "Problems in application of LDPC codes to information reconciliation in quantum key distribution protocols," arXiv:0908.2042v2 [cs.IT], 2009.

[58] A. Shokrollahi and R. Storn, "Design of Efficient Erasure Codes with Differential Evolution," in *ISIT 2000, IEEE International Symposium on Information Theory*, June 2000, p. 5.

[59] E. Berlekamp, R. McEliece, and H. Van Tilborg, "On the Inherent Intractability of Certain Coding Problems," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, May 1978.

[60] C. Di, D. Proietti, I. E. Telatar, T. J. Richardson, and R. L. Urbanke, "Finite-Length Analysis of Low-Density Parity-Check Codes on the Binary Erasure Channel," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1570–1579, June 2002.

[61] T. J. Richardson, "Error Floors of LDPC Codes," in *Proceedings of the Forty-First Annual Allerton Conference on Communication, Control, and Computing*, vol. 41, no. 3, October 2003, pp. 1426–1435.

[62] E. Sharon, S. Litsyn, and J. Goldberger, "An Efficient Message-Passing Schedule for LDPC Decoding," in *2004 23rd IEEE Convention of Electrical and Electronics Engineers*, September 2004, pp. 223–226.

[63] ——, "Efficient Serial Message-Passing Schedules for LDPC Decoding," *IEEE Transactions on Information Theory*, vol. 53, no. 11, pp. 4076–4091, November 2007.

[64] C. Roth, A. Cevrero, C. Studer, Y. Leblebici, and A. Burg, "Area, Throughput, and Energy-Efficiency Trade-offs in the VLSI Implementation of LDPC Decoders," in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, May 2011, pp. 1772–1775.

[65] Jun Chen, Da-ke He, and A. Jagmohan, "The equivalence between slepian-wolf coding and channel coding under density evolution," *IEEE Transactions on Communications*, vol. 57, no. 9, pp. 2534–2540, September 2009.

[66] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold, "Regular and Irregular Progressive Edge-Growth Tanner Graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, January 2005.

[67] Xiao-Yu Hu, E. Eleftheriou, and D.-M. Arnold, "Irregular Progressive Edge-Growth (PEG) Tanner Graphs," in *ISIT 2002, IEEE International Symposium on Information Theory*, July 2002, p. 480.

[68] Tao Tian, C. Jones, J. Villasenor, and R. Wesel, "Construction of Irregular LDPC Codes with Low Error Floors," in *ICC 2003, IEEE International Conference on Communications*, vol. 5, May 2003, pp. 3125–3129.

[69] Hua Xiao and A. Banihashemi, "Improved Progressive-Edge-Growth (PEG) Construction of Irregular LDPC Codes," *IEEE Communications Letters*, vol. 8, no. 12, pp. 715–717, December 2004.

[70] Sung-Ha Kim, Joon-Sung Kim, and Dae-Son Kim, "LDPC Code Construction with Low Error Floor Based on the IPEG Algorithm," *IEEE Communications Letters*, vol. 11, no. 7, pp. 607–609, July 2007.

[71] G. Richter, "An Improvement of the PEG Algorithm for LDPC Codes in the Waterfall Region," in *EUROCON 2005, International Conference on Computer as a Tool*, vol. 2, November 2005, pp. 1044–1047.

[72] J. Martinez-Mateo, D. Elkouss, and V. Martin, "Improved Construction of Irregular Progressive Edge-Growth Tanner Graphs," *IEEE Communications Letters*, vol. 14, no. 12, pp. 1155–1157, December 2010.

[73] M. R. Yazdani and A. H. Banihashemi, "On Construction of Rate-Compatible Low-Density Parity-Check Codes," *IEEE Communications Letters*, vol. 8, no. 3, pp. 159–161, March 2004.

[74] W. Matsumoto and H. Imai, "A Study on Rate-Compatible LDPC Codes," in *ISITA 2004, International Symposium on Information Theory and its Applications*, October 2004, pp. 529–534.

[75] Jeongseok Ha, Jaehong Kim, and S. W. McLaughlin, "Rate-Compatible Puncturing of Low-Density Parity-Check Codes," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2824–2836, November 2004.

[76] H. Pishro-Nik and F. Fekri, "Results on Punctured Low-Density Parity-Check Codes and Improved Iterative Decoding Techniques," *IEEE Transactions on Information Theory*, vol. 53, no. 2, pp. 599–614, February 2007.

[77] Chun-Hao Hsu and A. Anastasopoulos, "Capacity Achieving LDPC Codes Through Puncturing," *IEEE Transactions on Information Theory*, vol. 54, no. 10, pp. 4698–4706, October 2008.

[78] G. Richter, S. Stiglmayr, and M. Bossert, "Optimized Asymptotic Puncturing Distributions for Different LDPC Code Constructions," in *ISIT 2006, IEEE International Symposium on Information Theory*, July 2006, pp. 831–835.

[79] G. Richter, A. Hof, and C. Huppert, "Optimized Puncturing Distributions for long LDPC Codes and Different Channels," in *ISWCS 2006, 3rd International Symposium on Wireless Communication Systems*, September 2006, pp. 749–753.

[80] I. Andriyanova and R. L. Urbanke, "Waterfall Region Performance of Punctured LDPC Codes over the BEC," in *ISIT 2009, IEEE International Symposium on Information Theory*, July 2009, pp. 2644–2648.

[81] N. Bonello, Sheng Chen, and L. Hanzo, "Low-Density Parity-Check Codes and Their Rateless Relatives," *IEEE Communications Surveys Tutorials*, vol. 13, no. 1, pp. 3–26, First Quarter 2011.

[82] Jaehong Kim, A. Ramamoorthy, and S. McLaughlin, "Design of Efficiently-Encodable Rate-Compatible Irregular LDPC Codes," in *2006. ICC 2006, IEEE International Conference on Communications*, vol. 3, June 2006, pp. 1131–1136.

[83] Jaehong Kim, Woonhaing Hur, A. Ramamoorthy, and S. McLaughlin, "Design of Rate-Compatible Irregular LDPC Codes for Incremental Redundancy Hybrid ARQ Systems," in *ISIT 2006, IEEE International Symposium on Information Theory*, July 2006, pp. 1139–1143.

[84] Cuizhu Shi and A. Ramamoorthy, "Design and Analysis of E^2RC Codes," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 6, pp. 889–898, August 2009.

[85] Jeongseok Ha, Jaehong Kim, D. Klinc, and S. W. McLaughlin, "Rate-Compatible Punctured Low-Density Parity-Check Codes With Short Block Lengths," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 728–738, February 2006.

[86] B. N. Vellambi and F. Fekri, "Rate-Compatible Puncturing of Finite-Length Low-Density Parity-Check Codes," in *IEEE International Symposium on Information Theory*, July 2006, pp. 1129–1133.

[87] Hyo Yol Park, Jae Won Kang, Kwang Soon Kim, and Keum Chan Whang, "Efficient Puncturing Method for Rate-Compatible Low-Density Parity-Check Codes," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 3914–3919, November 2007.

[88] B. N. Vellambi and F. Fekri, "Finite-Length Rate-Compatible LDPC Codes: A Novel Puncturing Scheme," *IEEE Transactions on Communications*, vol. 57, no. 2, pp. 297–301, February 2009.

[89] M. El-Khamy, J. Hou, and N. Bhushan, "Design of Rate-Compatible Structured LDPC Codes for Hybrid ARQ Applications," *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 6, pp. 965–973, August 2009.

[90] Jeongseok Ha, D. Klinc, J. Kwon, and S. W. McLaughlin, "Layered BP Decoding for Rate-Compatible Punctured LDPC Codes," *IEEE Communication Letters*, vol. 11, no. 5, pp. 440–442, May 2007.

[91] Tao Tian and C. R. Jones, "Construction of Rate-Compatible LDPC Codes Utilizing Information Shortening and Parity Puncturing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2005, no. 5, pp. 789–795, 2005.

[92] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, "The security of practical quantum key distribution," *Reviews of Modern Physics*, vol. 81, no. 3, pp. 1301–1350, September 2009.

[93] R. Mantha, "Hybrid Automatic Repeat Request Schemes Using Turbo Codes and Low Density Parity Check Codes," Master's thesis, Department of Electrical and Computer Engineering, University of Toronto, 1999.

[94] C. Lott, O. Milenkovic, and E. Soljanin, "Hybrid ARQ: Theory, State of the Art and Future Directions," in *IEEE Information Theory Workshop on Information Theory for Wireless Networks*, July 2007, pp. 1–5.

[95] Jing Li and K. R. Narayanan, "Rate-Compatible Low Density Parity Check Codes for Capacity-Approaching ARQ Schemes in Packet Data Communications," in *The IASTED International Conference on Communications, Internet, and Information Technology*, November 2002.

[96] S. Sesia, G. Caire, and G. Vivier, "Incremental Redundancy Hybrid ARQ Schemes Based on Low-Density Parity-Check Codes," *IEEE Transactions on Communications*, vol. 52, no. 8, pp. 1311–1321, August 2004.

[97] E. Soljanin, R. Liu, and P. Spasojevic, "Hybrid ARQ with Random Transmission Assignments," in *Advances in Network Information Theory*. American Mathematical Society Publications, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 2004, Volume 66.

[98] N. Varnica, E. Soljanin, and P. Whiting, "LDPC Code Ensembles for Incremental Redundancy Hybrid ARQ," in *ISIT 2005, International Symposium on Information Theory*, September 2005, pp. 995–999.

[99] E. Soljanin, N. Varnica, and P. Whiting, "Punctured vs Rateless Codes for Hybrid ARQ," in *ITW 2006, IEEE Information Theory Workshop*, March 2006, pp. 155–159.

[100] M. Levorato and M. Zorzi, "Performance Analysis of Type II Hybrid ARQ with Low-Density Parity-Check Codes," in *ISCCSP 2008, 3rd International Symposium on Communications, Control and Signal Processing*, March 2008, pp. 804–809.

[101] P. Eraerds, N. Walenta, M. Legré, N. Gisin, and H. Zbinden, "Quantum key distribution and 1 Gbit/s data encryption over a single fibre," *New Journal of Physics*, vol. 10, p. 063027, 2010.

[102] 802.11 Working Group of the 802 Committee, "IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput," IEEE Std 802.11n-2009, 2009.

[103] J. Wolf and I. Blakeney, R.D., "An Exact Evaluation of the Probability of Undetected Error for Certain Shortened Binary CRC Codes," in *MILCOM 1988, IEEE Military Communications Conference*, vol. 1, October 1988, pp. 287–292.

[104] T. Klove and V. Korzhik, *Error Detecting Codes, General Theory and their Application in Feedback Communication Systems*. Kluwer Academic, 1995.

[105] T. Klove, *Codes for Error Detection*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2007.

[106] D. J. MacKay and M. C. Davey, "Evaluation of Gallager Codes for Short Block Length and High Rate Applications," in *Workshop on Codes, Systems and Graphical Models*, 1999, pp. 113–130.

[107] M. C. Davey and D. J. Mackay, "Reliable Communication over Channels with Insertions, Deletions, and Substitutions," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 687–698, February 2001.

[108] T. Wadayama, "On Undetected Error Probability of Binary Matrix Ensembles," in *ISIT 2008, IEEE International Symposium on Information Theory*, July 2008, pp. 1045–1049.

[109] S. Dolinar, K. Andrews, F. Pollara, and D. Divsalar, "Bounded Angle Iterative Decoding of LDPC Codes," in *MILCOM 2008, IEEE Military Communications Conference*, November 2008, pp. 1–6.

[110] ——, "The Limits of Coding with Joint Constraints on Detected and Undetected Error Rates," in *ISIT 2008, IEEE International Symposium on Information Theory*, July 2008, pp. 970–974.

[111] A. Kaiser, S. Dolinar, and M. Cheng, "Undetected Errors in Quasi-Cyclic LDPC Codes Caused by Receiver Symbol Slips," in *GLOBECOM 2009, IEEE Global Telecommunications Conference*, December 2009, pp. 1–6.

[112] C. Elliott, "Building the Quantum Network," *New Journal of Physics*, vol. 4, no. 1, pp. 46.1–46.12, 2002.

[113] M. Peev, C. Pacher, R. Alléaume, C. Barreiro, J. Bouda, W. Boxleitner, T. Debuisschert, E. Diamanti, M. Dianati, J. F. Dynes, S. Fasel, S. Fossier, M. Fúrst, J.-D. Gautier, O. Gay, N. Gisin, P. Grangier, A. Happe, Y. Hasani, M. Hentschel, H. Hübel, G. Humer, T. Länger, M. Legré, R. Lieger, J. Lodewyck, T. Lorünser, N. Lütkenhaus, A. Marhold, T. Matyus, O. Maurhart, L. Monat, S. Nauerth, J.-B. Page, A. Poppe, E. Querasser, G. Ribordy, S. Robyr, L. Salvail, A. W. Sharpe, A. J. Shields, D. Stucki, M. Suda, C. Tamas, T. Themel, R. T. Thew, Y. Thoma, A. Treiber, P. Trinkler, R. Tualle-Brouri, F. Vannel, N. Walenta, H. Weier, H. Weinfurter, I. Wimberger, Z. L. Yuan, H. Zbinden, and A. Zeilinger, "The secoqc quantum key distribution network in vienna," *New Journal of Physics*, vol. 11, no. 7, p. 075001, 2009.

[114] R. Alléaume, F. Roueff, E. Diamanti, and N. Lütkenhaus, "Topological optimization of quantum key distribution networks," *New Journal of Physics*, vol. 11, no. 7, p. 075002, 2009.

[115] T. E. Chapuran, P. Toliver, N. A. Peters, J. Jackel, M. S. Goodman, R. J. Runser, S. R. McNown, N. Dallmann, R. J. Hughes, K. P. McCabe, J. E. Nordholt, C. G. Peterson, K. T. Tyagi, L. Mercer, and H. Dardy, "Optical networking for quantum key distribution and quantum communications," *New Journal of Physics*, vol. 11, no. 10, p. 105001, 2009.

[116] A. Amraoui, A. Montarani, T. J. Richardson, and R. L. Urbanke, "Finite-Length Scaling for Iteratively Decoded LDPC Ensembles," *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 473–498, February 2009.

[117] R. Yazdani and M. Ardakani, "Waterfall Performance Analysis of Finite-Length LDPC Codes on Symmetric Channels," *IEEE Transactions on Communications*, vol. 57, no. 11, pp. 3183–3187, November 2009.

Acronyms

ACE Approximate Cycle EMD

ACK Acknowledgment

ARQ Automatic Repeat reQuest

AWGN Additive White Gaussian Noise

BB84 1984 Bennett and Brassard QKD protocol

BEC Binary Erasure Channel

BER Bit Error Rate

BP Belief Propagation

BSC Binary Symmetric Channel

CRC Cyclic Redundancy Check

DMC Discrete Memoryless Channel

EMD Extrinsic Message Degree

FCD Free Check-node Degree

FEC Forward Error Correction

FER Frame Error Rate

GF Galois Field

HARQ Hybrid Automatic Repeat reQuest

HW Hardware

IR Incremental Redundancy

LLR Log-Likelihood Ratio

LPDC Low-Density Parity-Check

LUT Look-Up Table

MPA Message-Passing Algorithm

NAK Negative Acknowledgment

OTP One-Time Pad

PEG Progressive Edge-Growth

QBER Quantum Bit Error Rate

QC Quasi Cyclic

QKD Quantum Key Distribution

RC Rate Compatible

RSA Rivest, Shamir and Adleman protocol

SNR Signal-to-Noise Ratio

SPA Sum Product Algorithm

Index

Part IV

Appendices

Appendix A

Theoretical Thresholds

In the *δ-modulated* method proposed in Section 4.3 for the construction of rate-adaptive LDPC codes, there is a tradeoff between the covered error range, increasing with δ, and the efficiency of the procedure, decreasing with higher δ values. The higher δ value, the greater covered error range, but the threshold if getting far from the un-modulated threshold for δ values higher that 0.1. This behavior was originally shown in Ref. [34], and it is also depicted here in Figure A.1.

Figure A.1 shows the theoretical efficiency (threshold) of the proposed protocol for the asymptotic case, i.e. assuming that infinite length LDPC codes are used for the reconciliation. Efficiency was computed for different proportions of punctured and shortened symbols. These theoretical values were computed using a discretized version of the density evolution algorithm [54] to estimate the theoretical threshold of each code for the proposed proportion of punctured and shortened symbols.

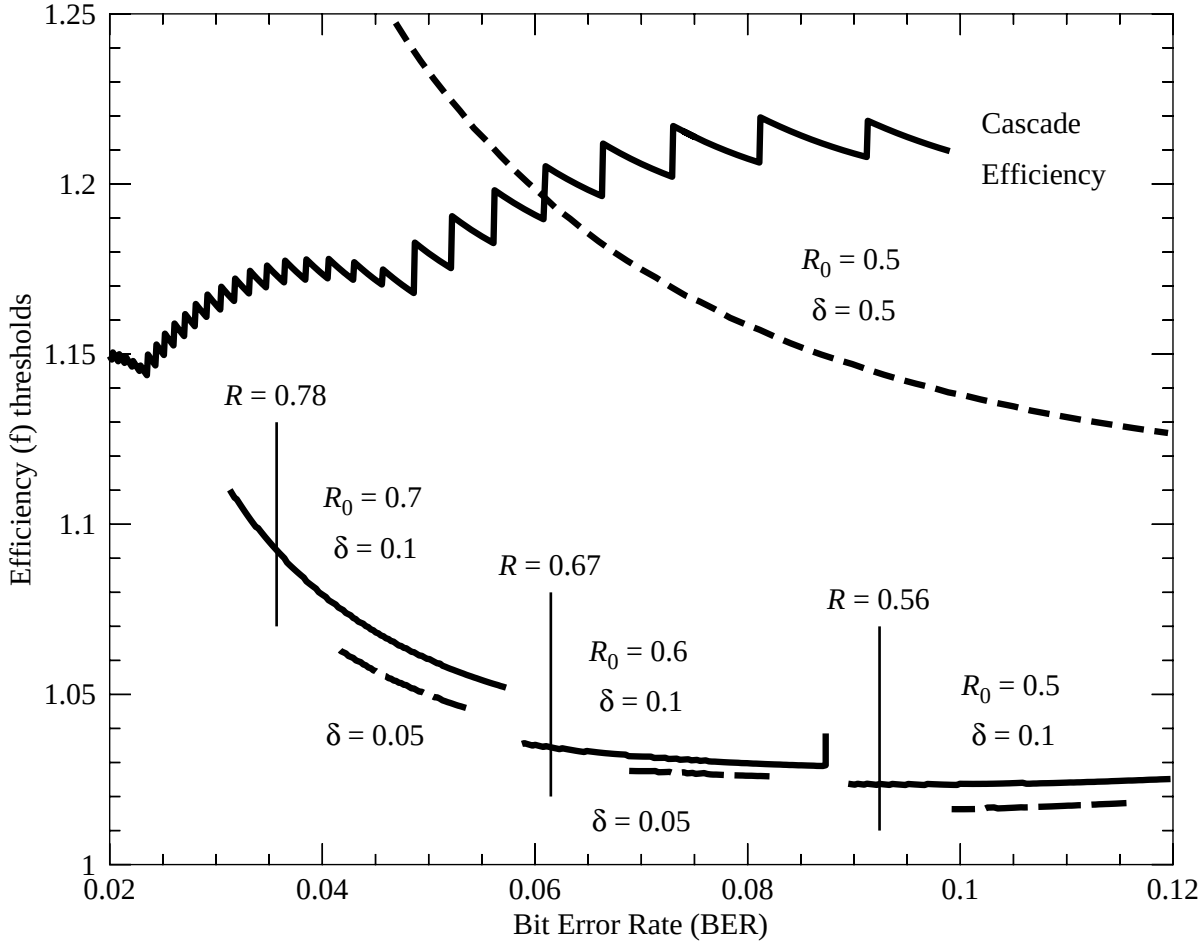

Figure A.1: Efficiency thresholds computed for the proposed construction of δ-modulated rate-adaptive LDPC codes using different proportion of punctured and shortened symbols.

Appendix B

Analysis of Finite Length Low-Density Parity-Check Codes

Nowadays, there exist some methods to analyze the asymptotic behavior of LDPC codes. One of those methods is the *density evolution* proposed in Ref. [31]. This is a well-known procedure able for determining the capacity of LDPC codes (i.e. the theoretical threshold of a family of LDPC codes) under message-passing decoding. However, the analytical behavior of finite length LDPC codes in the mostly used channels is, to date, one of the most important uncovered problems in the area. An initial approach was carried out for the binary erasure channel in Ref. [60], but considering only a given regular ensemble of LDPC codes. Recently, new studies have been proposed for the analytical study of finite length codes in the waterfall region. These new approaches were motivated by the study of a physical phenomenon (commonly used in statistical physics) described by the *scaling law*, an observed phenomenon in most systems when go through a phase transition state. This transition state occurs in LDPC decoding when the channel crossover probability, ϵ, is achieving the threshold of the code, ϵ^*. This phenomenon was firstly applied for the study of finite length LDPC codes in the binary erasure channel in Ref. [116], and extended for the study of punctured codes in Ref. [80]. In this work we use the approximation of this

phenomenon for finite length communications using the concept of *observed* channel introduced in Ref. [117], a simpler approach but with an acceptable accuracy.

B.1 Observed Channel

Any communication channel (discrete memoryless channel) is stochastically modeled by a set of parameters. For instance, the binary symmetric channel (BSC) is parameterized by its error rate ϵ, BSC(ϵ), as shown in Figure 2.2. However, these parameters accurately describe the behavior of the modeled channel only in the asymptotic case, i.e. assuming infinite length communications. In the BSC(ϵ) we define the *observed* bit error rate in a communication, P_{obs}, as the number of errors divided by the length of this communication, N. This observed value is constant only in the asymptotic case, i.e. $P_{\text{obs}} = \epsilon \ \forall N$ only when $N \to \infty$. The distribution of errors in our *observed* BSC channel is then described by the following probability mass function (pmf):

$$f_{P_{\text{obs}}}(\epsilon, N, x) = \binom{N}{Nx} \epsilon^{Nx} (1-\epsilon)^{N-Nx} \tag{B.1}$$

where Nx is the number of errors in the communication of length N, such that it is an integer in the range $[0, N]$, i.e. $0 \leq Nx \leq N$. For convenience we will omit the parameters for the channel and the communication length, $\mathcal{C}(\theta)$ and N respectively, when these are understood without complicating the notation. Notice that since $f_{P_{\text{obs}}}(x)$ is a pmf, we have that $\sum_x f_{P_{\text{obs}}}(x) = 1$.

Assuming that the length of the communication is large enough, i.e. when it is higher than a few thousand bits, this pmf can be approximated with high precision by using a (continuous) Gaussian probability density function centered around the error rate ϵ and with variance $\sigma^2_{P_{\text{obs}}} = \epsilon(1-\epsilon)/N$:

$$f_{P_{\text{obs}}}(\epsilon, N, x) \approx \mathcal{N}(\mu_{P_{\text{obs}}}, \sigma^2_{P_{\text{obs}}}) \tag{B.2}$$

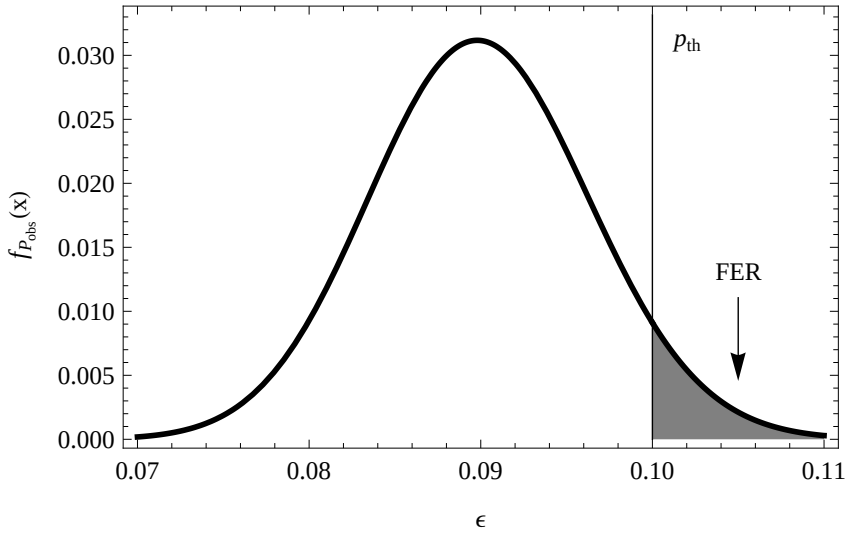

Figure B.1: Graphical interpretation of frame error rate.

B.2 Frame Error Rate

Let us now consider that we are using a finite length linear code to correct any error occurred during the communication, then we can estimate the ratio of codewords that cannot be corrected by calculating the probability that the observed channel behaves worse than the decoding threshold of our code, ϵ^* (see Ref. [31]). Figure B.1 shows a graphical interpretation of this ratio —ratio of codewords that cannot be corrected— according to the error distribution in an observed channel, i.e. assuming a finite length communication.

Using an error correction code of length N with a theoretical threshold of ϵ^*, the FER for our BSC(ϵ) channel can be reasonably approximated by:

$$F_{P_{\text{obs}}}(\epsilon, N, \epsilon^*) = 1 - \Pr(P_{\text{obs}} \leq \epsilon^*) \tag{B.3}$$

$$= \Pr(P_{\text{obs}} > \epsilon^*) = \int_{\epsilon^*}^{1} f_{P_{\text{obs}}}(\epsilon, N, x)dx \tag{B.4}$$

Using the Gaussian approximation:

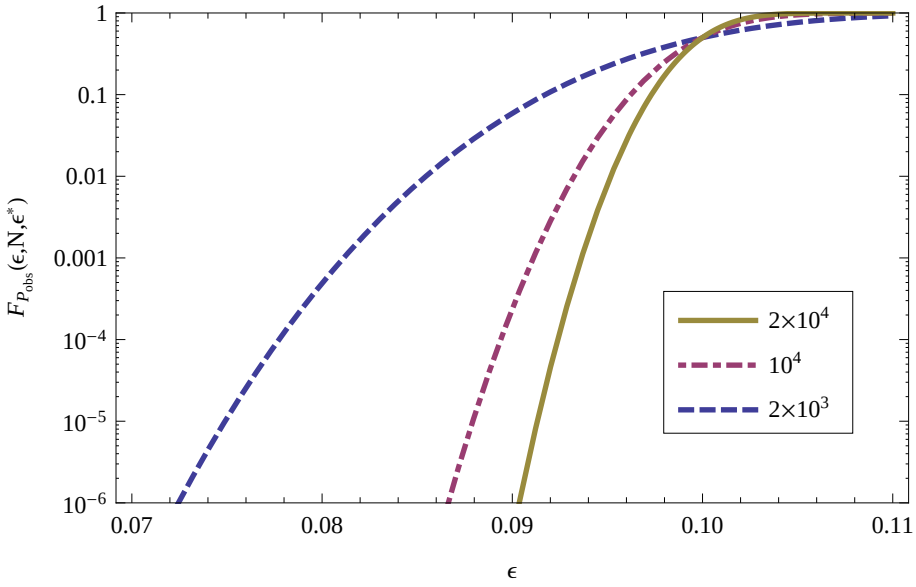

Figure B.2: Finite length analysis for different communication lengths.

$$F_{P_{\text{obs}}}(\epsilon, N, \epsilon^*) \approx \frac{1}{\sqrt{2\pi\epsilon(1-\epsilon)/N}} \int_{\epsilon^*}^{1} e^{-\frac{N(x-\epsilon)^2}{2\epsilon(1-\epsilon)}} dx \tag{B.5}$$

Note that, for convenience, we have used the term F instead of $F_{P_{\text{obs}}}(\mathcal{C}(\theta), N, \epsilon^*)$ in the main body of the paper.

Note also that this analytical approximation is only valid for the behavior in the *waterfall* region of an error correction code, since it does not include information about the performance in the error floor regime. Figure B.2 shows the performance of this approach for a correcting code with threshold of 0.10 and different codeword lengths of 2×10^3, 10^4 and 2×10^4 bits.

Appendix C

Ensembles of Low-Density Parity-Check Codes

Table C.1 shows the symbol node distribution, $\lambda(x)$, of those ensembles of LDPC codes that have used in this work. Other ensembles of LDPC codes have been extracted from Refs. [26,52].

Table C.1: λ-distribution of LDPC code ensembles for different coding rates.

Rate:	$R = 0.3$	$R = 0.4$	$R = 0.5$	$R = 0.6$	$R = 0.7$	$R = 0.8$
λ_1	0.247205	0.181749	0.159673	0.116530	0.091699	0.066795
λ_2	0.225225	0.147329	0.121875	0.125646	0.171401	0.194832
λ_3	0.054374	0.054427	0.112610	0.108507	0.068388	0.057052
λ_4		0.070728	0.190871	0.053422	0.120523	0.064502
λ_6		0.068692		0.072723		
λ_7				0.034796		
λ_8	0.153518	0.135139		0.072999		0.204606
λ_9	0.168646		0.077062			
λ_{10}					0.187471	
λ_{14}						0.096441
λ_{17}				0.075261		
λ_{24}			0.337909			
λ_{27}					0.208278	
λ_{28}						0.238720
λ_{29}					0.152239	
λ_{31}				0.117103		
λ_{34}		0.159581				0.077052
λ_{39}	0.151032	0.182355				
λ_{44}				0.223013		
Threshold:	0.180247	0.140508	0.102592	0.074526	0.050187	0.028941
Shannon:	0.189298	0.146102	0.110028	0.079383	0.053239	0.031124
Channel:	BSC	BSC	BSC	BSC	BSC	BSC

Vitae

Jesus Martinez-Mateo is a doctoral researcher at the Technical University of Madrid (UPM). He has a bachelor degree in Computer Science Engineering and master of science in Computational Mathematics by the UPM. He is member of the research group on Quantum Information and Computation. He is contributing to several research projects for the study and design of a metropolitan quantum key distribution network. Outstanding projects: CENIT Segur@. funded by the Ministry of Trade and Industry of Spain, and QUITEMAD among others.

He reconciles his research with the development of free and open source software (e.g. Lan Core). He is also member of an university cooperation for development group, TEDECO (Technology for Development and Cooperation) at UPM, and he has been part of two field missions to the University of Ngozi, in Burundi.