CHAPTER

CHANNEL CODING

8

Forward error correction (FEC) schemes play a fundamental role in every digital communication system, because they provide robustness against noise and other channel uncertainties (e.g., imperfect channel-state information). The large range of use cases envisaged for 5G NR (see Chapter 1) makes the problem of designing good FEC schemes particularly challenging. Indeed, together with the traditional *enhanced mobile broadband* (eMBB) use case, which involves providing a high data rate to mobile users, 5G NR will also support two new use cases: *massive machine-type communication* (mMTC), which aims at guaranteeing connectivity to a very large number of low-cost and low-energy devices, and *ultrareliable low-latency communications* (URLLC), which deals with providing connectivity with latency and reliability levels that are orders of magnitude more stringent than in eMBB and mMTC [28]. As we shall see, the FEC schemes that are optimal for data transmission in eMBB may be suboptimal in URLLC applications, where the latency constraint imposes severe restrictions on the code blocklength. Furthermore, whereas the problem of designing good FEC schemes for large blocklength values is well understood both from a theoretical and a practical implementation viewpoint, designing good FEC schemes for short blocklength is—as we shall see—still an area of research.

In this chapter, we shall provide an introduction to FEC schemes for 5G NR. Focusing first on a simple binary-input additive white Gaussian noise (AWGN) channel, we shall first characterize the fundamental trade-off between rate, packet error probability, and blocklength using recently developed nonasymptotic tools in information theory [29].

We shall then review the family of low-density parity-check (LDPC) and polar codes that have recently been selected for data and control channel, respectively, in 5G new-radio (NR) eMBB transmission [1], and benchmark their performance against the theoretical bounds given by nonasymptotic information theory. We will also review other FEC schemes that are competitive for short blocklengths and may be relevant for URLLC applications.

Finally, looking beyond what is currently standardized in NR, we shall consider the problem of communicating over multiantenna fading channels, and we shall use nonasymptotic information theory to shed light on the role of frequency and spatial diversity (see Chapter 1), with specific focus on the URLLC use case. We shall see that the use of space-frequency codes is necessary to achieve the required ultrahigh reliability. All the performance results reported in this chapter can be reproduced using the numerical routines described in [10,20], which are available online.

5G Physical Layer. https://doi.org/10.1016/B978-0-12-814578-4.00013-8

8.1 FUNDAMENTAL LIMITS OF FORWARD ERROR CORRECTION

8.1.1 THE BINARY AWGN CHANNEL

We shall focus for simplicity of exposition on the binary-input AWGN (bi-AWGN) channel, i.e., a memoryless discrete-time AWGN channel

$$y_k = \sqrt{\rho} x_k + w_k, \quad k = 1, \ldots, n, \tag{8.1}$$

whose input symbols $\{x_k\}$ belong to the binary alphabet $\{-1, 1\}$. We assume that the additive noise samples $\{w_k\}$ are independent and identically distributed zero mean, unit variance Gaussian random variables. Hence, ρ can be thought of as the signal-to-noise ratio (SNR). Finally, n in (8.1) denotes the number of discrete-time channel uses that can be employed to transmit a packet of information bits.

8.1.2 CODING SCHEMES FOR THE BINARY-AWGN CHANNELS

We shall next review the fundamental limits on the rate at which we can communicate over this channel for a given latency (expressed in terms of number of channel uses) and reliability constraint. To do so, we first introduce the concept of a channel coding scheme for the channel (8.1). We shall focus on coding schemes with codewords whose entries belong to the binary field $\mathbb{F}_2$ and assume that each binary coded symbol c_k is mapped into the binary phase-shift-keying (BPSK) symbol $x_k = 2c_k - 1$ in the Euclidean space.

Definition 1. A (n, M, ϵ) (binary) coding scheme for the channel (8.1) consists of:

- An encoder $f : \{1, 2, \ldots, M\} \mapsto \mathbb{F}_2^n$ that maps the information message $j \in \{1, 2, \ldots, M\}$, where $M = 2^k$ and k is the number of information bits, to a codeword in the set $\{\mathbf{c}_1, \ldots \mathbf{c}_M\}$, with $\mathbf{c}_m \in \mathbb{F}_2^n$, $m = 1, \ldots, M$. The set of M codewords is commonly referred to as the channel code or the codebook.
- A decoder $g : \mathbb{R}^n \mapsto \{1, 2, \ldots, M\}$ that maps the received sequence $\mathbf{y} \in \mathbb{R}^n$ into a message $\widehat{j} \in \{1, 2, \ldots, M\}$, or, possibly, it declares an error. This decoder satisfies the average packet error probability constraint

$$\mathbb{P}\left\{\widehat{j} \neq j\right\} \leq \epsilon. \tag{8.2}$$

□

The rate R of a (n, M, ϵ) coding scheme is $R = \log_2(M)/n = k/n$. A remark on terminology is at this point appropriate. Following [26], we differentiate between a *code* (i.e., the list of codewords) and a *coding scheme* (i.e., the code together with the encoder and the decoder), because a given code can be decoded using different decoding algorithms (often of drastically different complexity), yielding different coding schemes. We warn the reader that this distinction is frequently omitted in the coding-theory literature.

8.1.3 PERFORMANCE METRICS

We define the maximum coding rate $R^*(n, \epsilon)$ as the largest rate achievable with (n, M, ϵ) coding schemes. Mathematically,

$$R^*(n,\epsilon) = \sup\left\{\frac{\log_2(M)}{n} : \exists\,(n, M, \epsilon) \text{ coding scheme}\right\}. \tag{8.3}$$

Determining $R^*(n,\epsilon)$, which describes the fundamental trade-off between rate, blocklength, and packet error probability in the transmission of information, is a fundamental problem in information theory, with a long history. In a seminal and groundbreaking contribution, Shannon characterized the asymptotic behavior of $R^*(n,\epsilon)$ in the limit $n \to \infty$ for general memoryless channels [35]. Specifically, he showed that, for every $0 < \epsilon < 1$, the maximum coding rate $R^*(n,\epsilon)$ converges to a constant C—the so-called channel capacity—that depends on the characteristics of the channel. The consequence of this result is that, for every transmission rate R less than C, there exists a sequence of coding schemes with rate R and vanishing packet error probability as $n \to \infty$. Conversely, one can show that if $R > C$, then the packet error probability over most memoryless channels of practical relevance (including the bi-AWGN channel (8.1)) goes to 1. This means that reliable communication is possible only at rates less than C.

For the bi-AWGN channel in (8.1), the channel capacity is given by

$$C = \frac{1}{\sqrt{2\pi}} \int e^{-z^2/2}\Big(1 - \log_2\big(1 + e^{-2\rho + 2z\sqrt{\rho}}\big)\Big)\,\mathrm{d}z. \tag{8.4}$$

The proof of Shannon's coding theorem is based on a random-coding argument, and it does not provide a constructive way to approach the channel capacity. Indeed, it took about 50 years from the publication of Shannon's paper for the coding community to demonstrate near-capacity performance with practical coding schemes [11]. We shall review some of these coding schemes in Section 8.2.

It is worth stressing at this point that the channel capacity C is an asymptotic performance metric describing the behavior of the maximum coding rate $R^*(n,\epsilon)$ in the limit $n \to \infty$. This means that capacity cannot be used to benchmark the performance of coding schemes in which the blocklength n is short, as it is expected in some 5G use cases, due, for example, to a latency constraint.

This observation has renewed the interest in nonasymptotic characterizations of the maximum coding rate $R^*(n,\epsilon)$. The exact computation of such a quantity is a formidable task unless the number M of codewords is very small (e.g., $M = 4$; see [39]). However, tight upper (converse) and lower (achievability) bounds on $R^*(n,\epsilon)$ that can be computed efficiently can be obtained for a variety of channels for practical interest for 5G, including the bi-AWGN channel (8.1), using the finite-blocklength information-theoretic tools recently introduced in [29].

Such upper and lower bounds are depicted in Fig. 8.1 for the bi-AWGN channel. Specifically, Fig. 8.1A illustrates the tightest known bounds as a function of the blocklength n, for a target error probability of 10^{-4} and $\rho = 0$ dB. Equivalently, we can use the bounds to study the minimum packet error probability $\epsilon^*(n, R)$ achievable for a fixed blocklength n and rate R:

$$\epsilon^*(n, R) = \min\left\{\epsilon : \exists\,(n, \lceil 2^{nR} \rceil, \epsilon) \text{ coding scheme}\right\}. \tag{8.5}$$

This is illustrated in Fig. 8.1B where the bounds on $\epsilon^*(n, R)$ are plotted as a function of the minimum energy per bit E_b normalized with respect to the noise power spectral density N_0, which for the bi-AWGN channel, is given by

$$\frac{E_\mathrm{b}}{N_0} = \frac{\rho}{2R}. \tag{8.6}$$

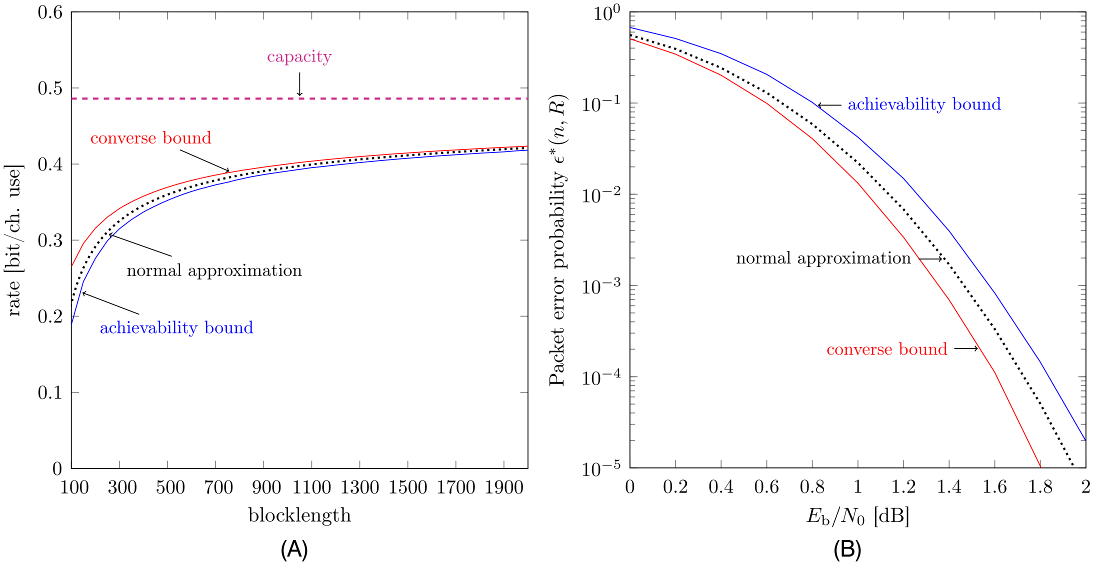

FIGURE 8.1

Bounds on the maximum coding rate $R^*(n,\epsilon)$ and on the minimum probability of error $\epsilon^*(n,R)$ achievable on the bi-AWGN channel (8.1). (A) $R^*(n,\epsilon)$ as a function of n for $\epsilon = 10^{-4}$, $\rho = 0$ dB. (B) $\epsilon^*(n,R)$ as a function of E_b/N_0 for $R = 0.5$, $n = 512$.

In both figures, the converse bound (which is an upper bound on $R^*(n,\epsilon)$ and a lower bound on $\epsilon^*(n,R)$) is based on the minimax converse [29, Thm. 27] (see [14] for details). The achievability bound (which is a lower bound on $R^*(n,\epsilon)$ and an upper bound on $\epsilon^*(n,R)$) is the random-coding union bound with parameter s (RCUs) [23, Thm. 1], a relaxation of the RCU bound [29, Thm. 16] that lends itself to efficient numerical evaluation. The dotted curve in Fig. 8.1A is the so-called normal approximation [29, Eq. (223)] to the maximum coding rate $R^*(n,\epsilon)$, which is given by

$$R^*(n,\epsilon) \approx C - \sqrt{\frac{V}{n}}Q^{-1}(\epsilon) + \frac{1}{2n}\log_2 n. \tag{8.7}$$

Here, V is the channel dispersion, which, for the bi-AWGN channel, is

$$V = \frac{1}{\sqrt{2\pi}} \int e^{-z^2/2}\Big(1 - \log_2\big(1 + e^{-2\rho+2z\sqrt{\rho}}\big) - C\Big)^2 \mathrm{d}z, \tag{8.8}$$

and $Q^{-1}(\cdot)$ is the inverse of the Gaussian Q-function. In Fig. 8.1B, the normal approximation is

$$\epsilon^*(n,R) \approx Q\left(\frac{C - R + (2n)^{-1}\log_2 n}{\sqrt{V/n}}\right). \tag{8.9}$$

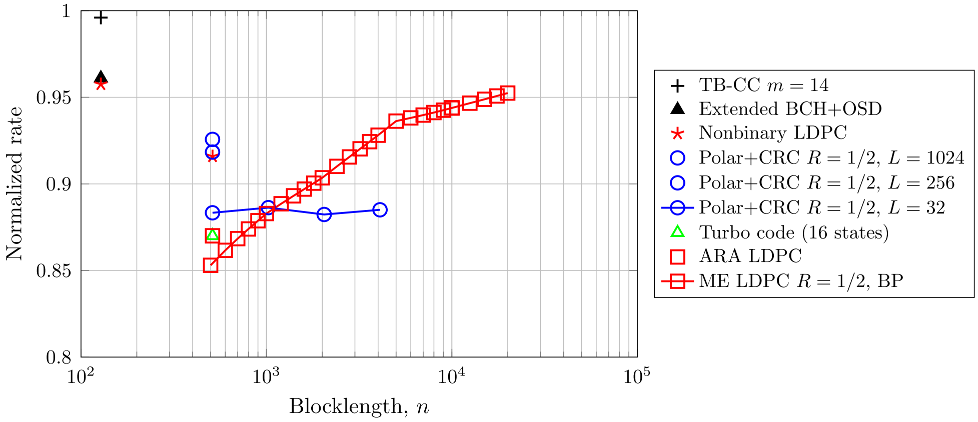

FIGURE 8.2

Normalized rate R_{norm} of some of the coding schemes reviewed in this chapter when used over the bi-AWGN channel (8.1). Here, $\epsilon = 10^{-4}$.

As can be seen from Fig. 8.1, the bounds characterize tightly the nonasymptotic performance metrics of interest, i.e., $R^*(n, \epsilon)$ and $\epsilon^*(n, R)$. Furthermore, the normal approximation, which is simple to evaluate numerically, turns out to be accurate in the parameter range considered in the figure. It is worth highlighting that this approximation is typically inaccurate in scenarios where both the required packet error probability and the SNR are low. In such scenarios, approximations based on saddle-point methods [23] should be used instead.

As we shall see in Section 8.3, the nonasymptotic bounds depicted in Fig. 8.1 can be generalized to include the presence of fading, and the use of pilot-aided transmission, and multiple antennas. Such generalizations provide valuable insights on how to optimally communicate over multiantenna fading channels.

The nonasymptotic bounds in Fig. 8.1 provide a natural way to benchmark the performance of practical coding schemes, which is more informative than the classic error probability vs. E_b/N_0 curves. Specifically, one fixes a code of a given blocklength n and rate R, and determines the minimum SNR $\rho_{\text{min}}(\epsilon)$ needed to achieve a target error probability ϵ. Then one defines the normalized rate

$$R_{\text{norm}} = \frac{R}{R^*(n, \epsilon)} \tag{8.10}$$

where the maximum coding rate $R^*(n, \epsilon)$ can be evaluated, for example, using the normal approximation (8.7) with ρ in (8.4) and (8.8) replaced by $\rho_{\text{min}}(\epsilon)$. Note that R_{norm} is a normalized performance metric that allows one to compare coding schemes of different blocklengths and different rates. The larger R_{norm}, the better the coding scheme.

Fig. 8.2 summarizes the performance over the bi-AWGN channel of some of the coding schemes we will review in the next sections of this chapter. As we shall see, there exist three different regimes in which different coding schemes are preferable: the short-blocklength regime, the moderate-blocklength regime, and the large-blocklength regime.

In the large-blocklength regime ($n \geq 1000$), modern codes that are decoded with belief propagation, such as multiedge-type LDPC codes and turbo codes, are the most competitive solutions. At moderate blocklengths ($400 \leq n \leq 1000$), good performance can be achieved using polar codes decoded with successive-cancellation decoding with a large list size and combined with an outer cyclic-redundancy check (CRC) code. Finally, in the short-blocklength regime ($n \leq 400$), some of the most promising solutions involve the use of short algebraic codes or linear block codes based on tail-biting trellises, decoded using near-maximum-likelihood (ML) decoding algorithms such as ordered-statistic decoding (OSD) [17], or LDPC codes over high-order finite fields. These insights have been taken into account in the 3GPP standardization activities. Indeed, LDPC codes will be used to protect the new-radio (NR) eMBB data channel and polar codes to protect the NR eMBB control channel [1].[1]

This confirms that nonasymptotic information-theoretic analyses provide concrete and useful guidelines on the design and the selection of actual coding schemes. We will see further examples of this principle in Section 8.3.

8.2 FEC SCHEMES FOR THE BI-AWGN CHANNEL

8.2.1 INTRODUCTION

Designing codes for large blocklengths is a well-investigated problem and effective solutions are available. Indeed, modern codes (e.g., turbo and LDPC codes) offer excellent performance under suboptimal but low-complexity iterative decoding algorithms such as belief propagation (BP). The design problem is more open for short blocklengths. On the one hand, the BP decoding performance becomes increasingly unsatisfactory when the blocklength decreases; on the other hand, a reduction in the blocklength makes it feasible to use near-maximum-likelihood decoding algorithms, which, when applied to, e.g., classical algebraic codes, yield performances that are sometimes superior to what can be achieved by modern codes with BP decoding.

In this section, we shall review some of the code constructions that are of interest for NR. Our emphasis will be mainly on the short- and moderate-blocklength regimes. For simplicity, we shall focus exclusively on the bi-AWGN channel. Also, in the spirit of the book, we shall highlight the general principles of each coding scheme, without delving too much into the many additional features (e.g., rate flexibility and suitability to hybrid automatic-repetition-request protocols) that are required for a coding scheme to be compatible with the requirements set in NR. Since our focus is on the bi-AWGN channel, we will not discuss coded-modulation techniques. We just highlight that the approach used in LTE to map coded bits into modulation symbols, which relies on bit-interleaved coded modulation (BICM), is suboptimal because it yields a well-known shaping loss for high-order constellations. This is particularly relevant for NR, which will support constellations belonging to sets of cardinality as large as 256. A different approach, which is becoming increasingly popular in fiber-optic applications, is to use the probabilistic shaping method proposed in [9]—an ingenious technique that provides rate adaptation and reduces significantly the shaping loss.

[1] The standardization of the coding schemes to be used in the other use cases is still ongoing.

8.2.2 SOME DEFINITIONS

We start our review by collecting here some standard definitions concerning linear block codes that will turn out to be useful in the remainder of this chapter (see, e.g., [31] for more details).

We say that the list of codewords $\mathcal{C}$ of an $(n, 2^k, \epsilon)$ binary coding scheme (see Definition 1) is a (n, k) *linear block code* if the 2^k codewords are a k-dimensional subspace of the vector space of all binary n-tuples. Here, addition and multiplication are the ones of the binary field $\mathbb{F}_2$. It follows by this definition that the codewords of a (n, k) linear block code can be expressed as a linear combination of k linearly independent codewords $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k$. In other words, the set $\{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_k\}$ forms a basis for $\mathcal{C}$. We can use this basis to perform encoding as follows. Let the so-called $k \times n$ *generator matrix* of the code be defined as[2]

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \\ \vdots \\ \mathbf{g}_k \end{bmatrix}. \tag{8.11}$$

Then the encoder output $\mathbf{c} = f(j)$ corresponding to the input message $j \in \{1, \ldots, 2^k\}$ is

$$\mathbf{c} = \mathbf{bG} \tag{8.12}$$

where $\mathbf{b}$ is the k-dimensional binary representation of j.

The $(n-k)$-dimensional dual space $\widetilde{\mathcal{C}}$ of $\mathcal{C}$ is the set of all binary n-tuples $\widetilde{\mathbf{c}}$ satisfying

$$\widetilde{\mathbf{c}}\mathbf{c}^T = 0, \quad \forall \mathbf{c} \in \mathcal{C}. \tag{8.13}$$

Let $\{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{n-k}\}$ be a basis of $\widetilde{\mathcal{C}}$. The parity check matrix (PCM) $\mathbf{H}$ of $\mathcal{C}$ is the $(n-k) \times n$ binary matrix

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \vdots \\ \mathbf{h}_{n-k} \end{bmatrix}. \tag{8.14}$$

Note that if $\mathbf{c} \in \mathcal{C}$ then

$$\mathbf{cH}^T = \mathbf{0}. \tag{8.15}$$

This highlights the important role of the PCM for error detection at the decoder. A linear block code is defined uniquely by the matrices $\mathbf{G}$ and $\mathbf{H}$.

For a given (n, k) linear block code, let $\mathcal{X} = \{\mathbf{x}(1), \mathbf{x}(2), \ldots, \mathbf{x}(2^k)\}$ be the set of BPSK coded sequences in the n-dimensional Euclidean space corresponding to the 2^k codewords. Under the assumption that the information message j is drawn uniformly from $\{1, 2, \ldots, 2^k\}$, the decoding rule that minimizes the packet error probability ϵ is the maximum-likelihood (ML) decoding rule

[2] Following the standard convention in coding theory, all vectors in the remainder of the chapter are row vectors.

$$\widehat{\jmath} = \underset{m \in \{1,\ldots,2^k\}}{\arg\max}\; p(\mathbf{y} \,|\, \mathbf{x}(m)) \tag{8.16}$$

where $p(\mathbf{y} \,|\, \mathbf{x})$ denotes the channel law (8.1),

$$p(\mathbf{y} \,|\, \mathbf{x}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\|\mathbf{y} - \sqrt{\rho}\mathbf{x}\|^2}{2}\right). \tag{8.17}$$

It follows from (8.17) that the ML decoding rule (8.16) can be equivalently expressed as

$$\widehat{\jmath} = \underset{m \in \{1,\ldots,2^k\}}{\arg\min}\; \|\mathbf{y} - \sqrt{\rho}\mathbf{x}(m)\|^2. \tag{8.18}$$

Stated explicitly, the ML decoder selects the message whose corresponding BPSK coded sequence is closest to the received signal $\mathbf{y}$ in the Euclidean space.

Assume that the coded sequence $\mathbf{x}(1)$, corresponding to $j = 1$, is transmitted. The probability that a different coded sequence $\mathbf{x}(\ell)$ with $\ell \neq 1$ is closer to the received signal $\mathbf{y}$ than $\mathbf{x}(1)$ depends on the Euclidean distance between $\mathbf{x}(\ell)$ and $\mathbf{x}(1)$. Specifically,

$$\mathbb{P}\{\widehat{\jmath} = \ell \,|\, j = 1\} = Q\left(\frac{\sqrt{\rho}\|\mathbf{x}(\ell) - \mathbf{x}(1)\|}{2}\right). \tag{8.19}$$

Let now K_d be the average number of coded sequences that have a neighbor at distance d. It follows from (8.19) and from an application of the union bound that [16, Eq. (2.32)]

$$\epsilon \leq \sum_d K_d\, Q\left(\frac{\sqrt{\rho}d}{2}\right). \tag{8.20}$$

The upper bound in (8.20) is typically dominated by the first term

$$K_{d_{\min}}\, Q\left(\frac{\sqrt{\rho}d_{\min}}{2}\right) \tag{8.21}$$

where $d_{\min}$ is the minimum Euclidean distance between any two coded sequences. This quantity is equal to twice the minimum Hamming weight of the nonzero codewords in $\mathcal{C}$. The bound (8.20) highlights the dependence of the packet error probability ϵ on the distance spectrum of the code. More sophisticated and tighter bounds on the packet error probability of linear block codes under ML decoding are described in [33]. It is worth highlighting that the evaluation of the ML rule (8.18) is in practice unfeasible already for values of k larger than a few tens of bits because of the complexity, unless the code possesses structures that facilitate it.

8.2.3 LDPC CODES

8.2.3.1 Fundamentals of LDPC Codes

LDPC codes are a class of linear block codes characterized by a PCM that is sparse, i.e., it contains only few nonzero entries. Originally proposed by Gallager [18] in the 1960s, and later rediscovered and generalized in the 1990s [22,5], LDPC codes provide a performance close to capacity for a large set of communication channels. These codes are currently deployed in several standards, including

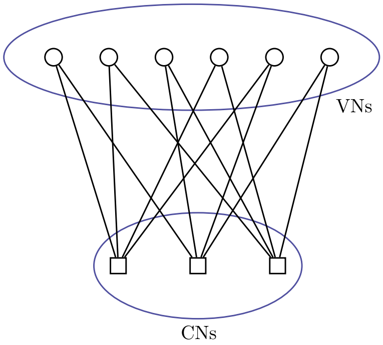

FIGURE 8.3

Tanner graph of the linear code with parity-check matrix given in (8.22).

IEEE802.11n, IEEE802.16e (WiMAX), IEEE 802.11ad (WiGig) and DVB-S2. As we shall briefly review, the sparseness of the PCM enables the use of low-complexity iterative decoding algorithms, which provide often near-ML performance. Throughout this section, we shall focus on binary LDPC codes. Extensions to higher fields are discussed in Section 8.2.5.3.

An (n, k) binary LDPC code is defined in terms of a $m \times n$ sparse PCM. Here, $m \geq n - k$, which implies that the PCM may be rank deficient. A convenient way to represent the PCM is through its Tanner graph representation [37]. A Tanner graph of an LDPC code is a bipartite graph, i.e., a graph in which the nodes are of two different types, and the edges connect only nodes of different types. These two types of nodes are commonly referred to by variable nodes (VNs), which are as many as the codeword length n, and by check node (CN), which are as many as the number of rows m in the PCM, i.e., as many as the parity-check equations. The Tanner graph is constructed from the PCM by drawing an edge between the ith check node and the jth variable node whenever the entry $[\mathbf{H}]_{ij}$ of the PCM contains a 1. As an example, the linear block code with PCM given by

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{8.22}$$

is equivalently described by the Tanner graph depicted in Fig. 8.3.

We say that a LDPC code is regular if all VNs have the same degree, i.e., they are connected to the same number of edges, and the same occurs for the CNs. In the example of Fig. 8.3, all VNs have degree 2 and all CNs have degree 4. Allowing for VNs and CNs of different degrees, which results in irregular LDPC codes, turns out to be beneficial from a performance viewpoint [30]. Irregular LDPC codes are typically described in terms of their degree distributions, which provide the fraction of all edges connected to VNs/CNs of each degree.

Coarsely speaking, the decoding of LDPC codes is an iterative process, often referred to as BP, in which log-likelihood ratios (LLRs) about the coded bits are exchanged along the edges of the Tanner graph. Each decoding iteration consists of two phases: a first phase in which, at each VN, the LLRs from the channel and from the upcoming edges are processed and transformed in updated LLRs sent to the neighboring CNs; and a second phase in which the LLRs arriving at each CN from the neighboring VNs is processed and updated LLRs are sent back. This process is repeated until a codeword is found or the maximum number of iterations is exceeded. The key observation is that the processing at the CNs and VNs depends only on locally available information, which allows for an efficient and parallelizable decoding process. However, since the decoding process is local, the globally optimal ML solution may not be found through this procedure, especially in the presence of short cycles in the Tanner graph. Indeed, such cycles constrain the decoding process to remain local. This observation also shows the importance of the low-density assumption, which facilitates the design of Tanner graphs free of short cycles.

One way to design LDPC codes is to use a pseudorandom algorithm that constructs a PCM with given degree distributions and avoids short cycles. Such an approach, despite yielding LDPC codes with extremely good performance [30], is impractical from an implementation viewpoint, because the absence of further structures in the PCM makes both the encoding and the decoding complexity too high for practically relevant blocklengths and rates.

A more practically appealing approach is to design structured LDPC codes constructed from a smaller protograph. The LDPC codes that one obtains through this construction form a subset of the more general class of MET-LDPC codes, whose performances are illustrated in Fig. 8.2. The PCM of protograph-based LDPC codes can be specified in terms of a small base matrix. The actual PCS is constructed from the base matrix by replacing each entry in the base matrix by a $Q \times Q$ binary matrix whose rows and columns have a weight equal to the corresponding entry in the base matrix. Here, Q is the so-called *lifting factor*. It is particularly convenient to pick as binary matrix a $Q \times Q$ cyclic permutation matrix, whose row and column weights are one. The resulting code is quasi-cyclic, a property that allows for simplified encoding and decoding, with negligible performance loss [21,31]. As an example, consider the following base matrix $\mathbf{H}_\mathrm{b}$:

$$\mathbf{H}_\mathrm{b} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \tag{8.23}$$

and assume that 3×3 cyclic permutation matrices are used to extend the base matrix to a PCM. Such a PCM may have the following structure:

$$\mathbf{H} = \left[\begin{array}{ccc|ccc|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array}\right]. \tag{8.24}$$

8.2.3.2 The LDPC-Code Solution Chosen for 5G NR

The LDPC codes chosen for the data channel in 5G NR are quasi-cyclic and have a rate-compatible structure that facilitates their use in hybrid automatic-repetition-request (HARQ) protocols.[3]

[3] This section is largely based on [32], where further information about the LDPC-code family chosen for 5G NR, including the rate-matching procedure and their use in combination with HARQ, can be found.

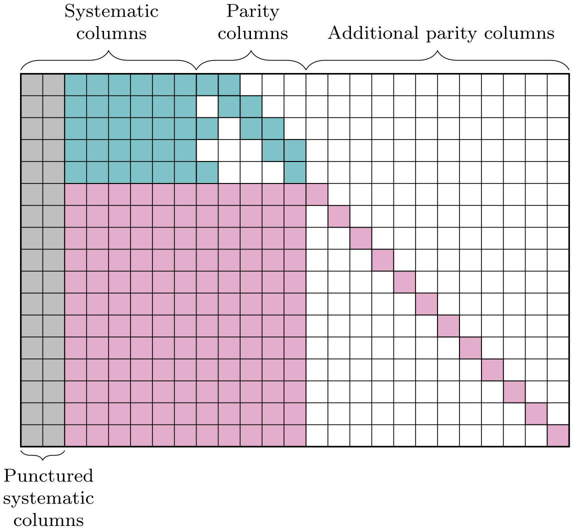

FIGURE 8.4

The general structure of the base matrix used in the quasi-cyclic LDPC codes selected for the data channel in NR.

To cover the large range of information payloads and rates that need to be supported in 5G NR, two different base matrices are specified. The general structure of these base matrices is provided in Fig. 8.4. In the figure, each white square represents a zero in the base matrix and each nonwhite square represents a one. The first two columns in gray correspond to punctured systematic bits that are actually not transmitted. Their addition is known to improve the threshold of the resulting code, i.e., its minimum SNR operating point [13]. The blue (dark gray in print version) part constitutes the kernel of the base matrix, and it defines a high-rate code. The dual-diagonal structure of the parity subsection of the kernel enables efficient encoding. Transmission at lower code rates is achieved by adding additional parity bits, i.e., by including an appropriately chosen subset of rows and columns in the base matrix containing entries marked in pink (light gray in print version). To enable maximum parallelism, the rows of the base matrix outside the kernel are designed so as to be orthogonal or quasi-orthogonal. The maximum lift factor $Q_{\max}$ is 384. This number is chosen to trade optimally between the parallel processing opportunities enabled by a large Q and the performance loss in terms of threshold due to the resulting higher amount of structure.

The base matrix #1, which is optimized for high rates and long blocklengths, supports LDPC codes of a nominal rate between $1/3$ and $8/9$. This matrix is of dimension 46×68 and has 22 systematic columns. Together with a lift factor of 384, this yields a maximum information payload of $k = 8448$ bits (including CRC).

The base matrix #2 is optimized for shorter blocklengths and smaller rates. It enables transmissions at a nominal rate between $1/5$ and $2/3$, it is of dimension 42×52, and it has 10 systematic columns. This implies that the maximum information payload is $k = 3840$.

The choice of each of the $Q \times Q$ circulant matrices to be substituted into the entries of the base matrix to form the full PCM is specified in [1]. These circulant matrices are selected so as to ensure efficient encoding and to obtain, at the same time, a good performance in terms of both error floor and threshold.

It is worth pointing out that, as observed in [32], the base matrix #2 tends to yield lower-complexity decoding and should in general be used whenever the information payload k is less than 3840 and the rate is less than $2/3$, whereas base matrix #1 should be used in the rest of the parameter range. Two exceptions are the case $k \leq 308$ for which base matrix #2 should be used for all rates, and the case $R \leq 1/4$, for which the base matrix #2 should be used for all information-payload sizes k.

8.2.4 POLAR CODES

8.2.4.1 Fundamentals of Polar Codes

Polar codes, introduced by Arıkan [6], are a class of linear block codes that provably achieve the capacity of memoryless symmetric channels, such as the bi-AWGN, with low encoding and decoding complexity, and a recursive structure that facilitates their hardware implementation.

To introduce the core idea behind polar codes, i.e., the so-called *channel polarization*, we start by observing that, among all bi-AWGN channels (8.1), there are two extreme types, for which the communication problem is trivial[4]:

- The perfect (noiseless) channel $y_k = \sqrt{\rho} x_k$.
- The useless channel $y_k = w_k$.

Uncoded transmission is sufficient to achieve the capacity of the first channel, whereas no information can be transmitted on the second channel. Arıkan's polarization technique is a lossless and low-complexity method to convert any binary-input symmetric channel into a mixture of extremal binary-input channels.

Polarization is achieved through the *polar transform*, which operates as follows: given two copies of a binary-input channel $W : \mathbb{F}_2 \to \mathcal{Y}$, where $\mathcal{Y}$ denotes the output set ($\mathcal{Y} = \mathbb{R}$ for the bi-AWGN channel), the polar transform creates, under the assumption of uniformly distributed inputs, two new *synthetic channels* $W^- : \mathbb{F}_2 \to \mathcal{Y} \times \mathcal{Y}$ and $W^+ : \mathbb{F}_2 \to \mathcal{Y} \times \mathcal{Y} \times \mathbb{F}_2$, defined as follows:

$$W^-(y_1, y_2 \,|\, u_1) = \frac{1}{2} \sum_{u_2 \in \mathbb{F}_2} W(y_1 \,|\, u_1 \oplus u_2) W(y_2 \,|\, u_2), \tag{8.25}$$

$$W^+(y_1, y_2, u_1 \,|\, u_2) = \frac{1}{2} W(y_1 \,|\, u_1 \oplus u_2) W(y_2 \,|\, u_2). \tag{8.26}$$

Here, $\oplus$ is the addition in $\mathbb{F}_2$. Such a transform is illustrated in Fig. 8.5. Note that the channel W^+, which has input u_2 and output (y_1, y_2, u_1), contains the channel W. Hence, its capacity under uniform inputs (which we shall assume throughout this section) is no smaller than the capacity of the channel W. Since the transformation

$$x_1 = u_1 \oplus u_2, \tag{8.27}$$

$$x_2 = u_2, \tag{8.28}$$

[4] See, e.g., [26, Ch. 14] and Telatar's plenary talk at the 2017 International Symposium on Information Theory https://goo.gl/zQz6nB for a more comprehensive introduction to polar codes.

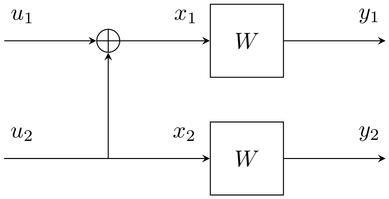

FIGURE 8.5

The polar transform.

is invertible, the sum of the capacities of W^+ and W^- must be equal to twice the capacity of W. This implies that the capacity of W^- must be smaller than that of the original channel W. To summarize, we started from two identical copies of W. Through the application of the polar transform we obtained two new synthetic channels: W^-, whose capacity is smaller than W, and W^+, whose capacity is larger than W.

Example. Consider a binary erasure channel (BEC), i.e., a discrete memoryless channel with input–output relation

$$y = \begin{cases} x & \text{with prob. } 1-p, \\ ? & \text{with prob. } p, \end{cases} \tag{8.29}$$

where x denotes the binary input and the symbol "?" denotes an erasure. In words, with probability $(1-p)$ the input symbol x is received correctly, and with probability p it is erased. One can verify that the synthetic channels W^- and W^+ induced by the polar transform are also BECs, with erasure probability $p^- = p(2-p) \geq p$ and $p^+ = p^2 \leq p$, respectively.

Some comments on the synthetic channels W^- and W^+ are in order. The channel W^-, which has input u_1 and output (y_1, y_2), is indeed a genuine channel, because the receiver has access to both y_1 and y_2. The channel W^+, however, has u_1 as output, which is not available at the receiver. However, W^+ can be synthesized by imposing a decoding order. Namely, we first decode u_1 using y_1 and y_2. Then we use y_1, y_2, and the estimate $\hat{u}_1$ of u_1 to decode u_2. This corresponds to successive-cancellation decoding. One can show that the block-error probability $\Pr\{(\hat{u}_1, \hat{u}_2) \neq (u_1, u_2)\}$ achievable with successive-cancellation decoding coincides with the one attainable by a genie-aided successive-cancellation decoder that uses u_1 instead of $\hat{u}_1$ in the second step of the decoding procedure. In other words, error propagation is not an issue if we measure performance in terms of block-error probability.

The polarization transform can now be applied again to the inputs of W^- and W^+. This results in the four channels W^{--}, W^{-+}, W^{+-}, and W^{++} illustrated in Fig. 8.6. This process can be applied recursively N times to synthesize 2^N channels out of 2^N copies of W. Applying this process to a BEC with erasure probability $p = 0.3$ for the case $N = 10$ yields 1024 synthetic channels, whose capacities, which are equal to one minus the corresponding erasure probability, are illustrated in Fig. 8.7. This figure exemplifies the polarization phenomenon: roughly 30% of the synthetic channels have capacity close to zero, i.e., they are useless, whereas 70% have capacity close to 1, i.e., they are almost perfect. Observe now that the capacity of the underlying BEC is exactly $1 - p = 0.7$. Consequently, Fig. 8.7

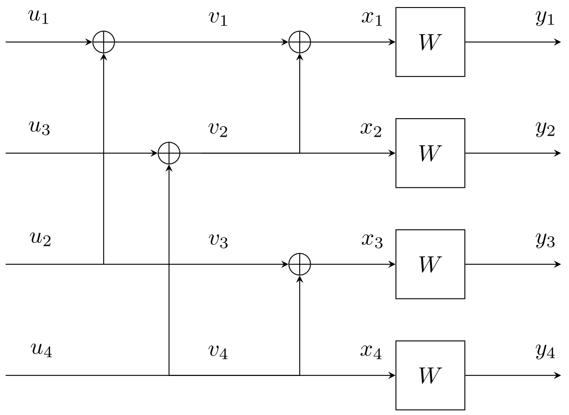

FIGURE 8.6

The four synthetic channels obtained by applying twice the polar transform.

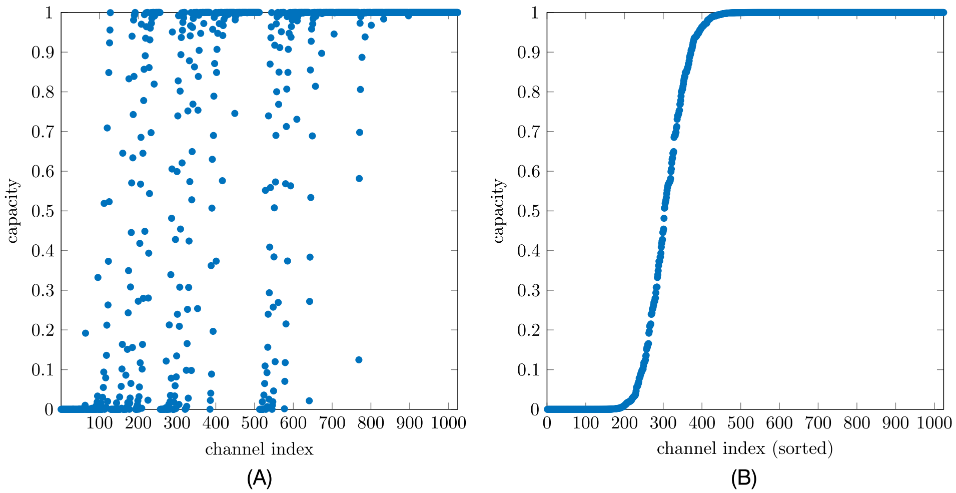

FIGURE 8.7

Capacity of the 1024 synthetic channels obtained by applying recursively the polar transform $N = 10$ times to a BEC with erasure probability p. (A) Unsorted. (B) Sorted.

suggests that, to achieve capacity, one can use a simple rate 0.7 binary code in which the k information bits are mapped to the almost perfect channels; the codeword entries corresponding to the remaining almost useless channels are frozen, i.e., they contain symbols known to the decoder (i.e., zeros).

More formally, we define a polar code as follows. Fix an integer N and let the blocklength be $n = 2^N$. Denote by $\mathbf{G} = \mathbf{F}^{\otimes n}$ the $n \times n$ polar transform matrix, where $\otimes$ stands for the Kronecker product, and

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \tag{8.30}$$

is the polar transform. Let also $\mathcal{U} \subset \{0, 1, \ldots 2^N - 1\}$ be a set of indices of cardinality k, where k is the size of the payload, and let $\mathcal{F}$ be the complementary set of size $n-k$. The k information bits are mapped to the entries belonging to $\mathcal{U}$ of a vector $\mathbf{u}$ of size n. The remaining entries, i.e., the ones with indices belonging to the set $\mathcal{F}$, are frozen to zero. In the remainder of the section, we shall refer to $\mathcal{F}$ as the frozen set. This set is constructed so as to contain the $n-k$ synthetic channels that have the smallest capacity. Finally, the resulting binary codeword $\mathbf{c}$ is obtained by performing the polarization mapping $\mathbf{c} = \mathbf{uG}$.

The receiver uses successive-cancellation decoding; the decoding order is obtained by applying a bit-reversal permutation to the channel index set: specifically, we number the channels from 0 to $n-1$, we obtain a new index for each channel by reversing the binary representation of its index, and we decode following the ordering induced by the new indices.

One can formally show [7] that the block-error probability ϵ achievable with the polar-coding scheme just described decays roughly as $2^{-\sqrt{n}}$, where $n = 2^N$ is the blocklength, for every rate below capacity. This proves that polar codes are indeed capacity achieving. Furthermore, they have the additional practical benefit over modern codes decoded with BP of not suffering from an error floor [24].

As far as complexity is concerned, the recursive nature of the polar transform allows one to perform encoding and decoding with a complexity that scales as $n \log_2 n$. Furthermore, these operations are naturally parallelizable.

One disadvantage of polar codes is that, when combined with successive-cancellation decoding, they offer mediocre performance in the short- and moderate-blocklength regimes that are of interest for control-channel applications. The performance of polar codes in this regime can be improved significantly by using instead a successive-cancellation list decoder [36]. This decoder considers simultaneously, at each decoding stage, L alternative paths, where L is the list size. The complexity of such a decoder is of order $Ln \log_2 n$. Performance can be further improved by concatenating the polar code with a high-rate CRC code. This code is used at the end of the list-decoding process to select the final codeword among the ones contained in the list. Concatenating a polar code with a CRC code turns out to be beneficial, because it improves the minimum distance of the polar code. As shown in Fig. 8.2, such an approach yields a state-of-the-art performance at moderate-blocklength values.

8.2.4.2 The Polar-Code Solution Chosen for 5G NR

Polar codes in 5G NR will be used to protect control signaling, with the exception of the transmission of payloads of up to $k = 11$ bits, for which Reed–Muller codes will be used, similar to 4G. Polar codes of different lengths and rates are supported. Such codes are obtained from an underlying parent code of length $2^{10} = 1024$. As recently summarized in [40], the specific polar code adopted in 5G NR relies on three innovations compared to what was described in the previous section, which allow the resulting coding scheme to satisfy the flexibility, processing latency, and complexity desiderata in 5G NR. Such innovations are:

- the offline computation of a deterministic reliability ordering of the synthetic channels;

- the use of parity-check bits to assist successive-cancellation list decoding also during its intermediate stages;
- a low-complexity rate-matching algorithm providing the needed rate and blocklength flexibility.

Deterministic Reliability Ordering

As explained in the previous section, one crucial step in the design of polar codes is the construction of the frozen-bit set $\mathcal{F}$. As shown in Fig. 8.7, the useless channels do not seem to follow a regular pattern. It turns out that, apart from the BEC, no efficient algorithm is known to rank the synthetic channels according to their reliability.[5] Since the reliability of the synthetic channels and their relative ordering depends on time-varying parameters such as the SNR, adaptive offline and online algorithms, which are able to order the synthetic channels as a function of the current values of channel parameters, appear to be unfeasible because of latency and/or memory requirements. The solution adopted in 5G NR is to assign to each synthetic channel of the parent code a deterministic, i.e., channel-parameter-independent, polarization weight that expresses its reliability. The polarization weights induce an ordering on the subchannels that is prestored so as to avoid online computations. Using techniques such as the β-expansion in number theory [19,40], these polarization weights can be chosen so as to satisfy the natural universal partial ordering existing among the synthetic channels [34].

To save complexity, the polarization weights of the parent code are used also for transmission involving shorter codes of length $n = 2^N$, $N < 10$. Specifically, let $\mathbf{q}^{n_{\max}}$ be the vector containing the ordered sequence of the 1024 synthetic-channel indices of the parent code, ordered according to increasing polarization weights. Then a reliability index list $\mathbf{q}^n$ for the shorter code is obtained by removing from $\mathbf{q}^{n_{\max}}$ all indices larger or equal to n. Although this choice is suboptimal, the significant complexity reduction justifies the resulting moderate performance loss. The polarization weights in 5G NR have been chosen through extensive simulations to guarantee good performance for all blocklengths and code rates.

Parity-Check Coding

The polar-code structure selected in NR relies on the addition of a more general, yet hardware friendly, outer code than just a CRC. Specifically, n_{pc} parity-check bits are appended to the information payload. Some of them are assigned to the $k + n_{\mathrm{pc}}$ unfrozen synthetic channels with lowest polarization weight. This improves the error performance of the code. Some of them are assigned to the unfrozen-bit positions in the vector $\mathbf{u}$ that correspond to the rows in the polar transform matrix $\mathbf{G}$ with smallest Hamming weights. This improves the distance spectrum of the resulting concatenated code.

To aid the intermediate steps of the successive-cancellation list-decoding algorithm, each parity-check bit is designed to depend only on preceding information bits. Specifically, the parity bits are computed through a length 5 shift register that evaluates the $\mathbb{F}_2$-sum of information bits that are five positions apart; see [40, Algorithm 3] for details. Empirical evidence suggests that enforcing such spacing makes the resulting scheme robust against error propagation. The same shift-register architecture can be used by the successive-cancellation list decoder to prune all paths that result in an erroneous parity-check bit.

[5] See, however, [34,25,19] for recent progress on this problem, based on universal partial ordering, and the β-expansion in number theory.

Rate Adaptation

The polar-code construction reviewed so far allows one to generate codes of blocklength $n = 2^N$ for some integer $N \leq 10$. Additional blocklength values can be obtained by *puncturing* or *shortening* [8].

In puncturing, one transforms the original (n, k) polar code into a $(n - p, k)$ polar code by removing $p < n - k$ bits from each codeword **c**. At the receiver, the channel LLRs corresponding to the punctured coded bits are set to 0 (the bits are assumed erased), and then the decoder of the original (n, k) polar code is applied. The presence of p zero-valued channel LLRs induces, through the decoding process, zero-valued LLRs for p of the entries of the n-dimensional input vector **u**. To avoid poor performance, these entries must be set to frozen bits. This can be achieved by puncturing the bits of **c** with indices corresponding to the reverse-bit permutations of $\{1, 2, \dots, p\}$ and by freezing the corresponding bits of **u**. The remaining frozen bits are chosen among the ones with lowest polarization weight, as usual.

We next discuss shortening. Let us assume that the original (n, k) code is systematic. Shortening allows one to obtain a $(n - p, k - p)$ code, $p < k$, by setting p systematic bits to zero and by not transmitting them. At the decoder side, the channel LLRs corresponding to the punctured systematic bits are set to infinity, and the decoder of the original code is applied. For nonsystematic polar codes, a natural way to select the coded bits to shorten is by requiring them to be linear combinations of frozen bits. This can be achieved by shortening the bits of **c** with indices corresponding to the reverse-bit permutations of $\{n - p + 1, \dots, n\}$ and by freezing the corresponding bits of **u**. As before, the remaining frozen bits are chosen among the ones with lowest polarization weight.

8.2.5 OTHER CODING SCHEMES FOR THE SHORT-BLOCKLENGTH REGIME

To conclude our overview, we shall next present a selection of additional coding schemes that exhibit a favorable performance/complexity trade-off in the short-blocklength regime ($n < 400$). Even though these schemes are not standardized in 5G, some of them may enter future releases due to their suitability for URLLC.

8.2.5.1 Short Algebraic Linear Block Codes With Ordered-Statistics Decoding

As already mentioned, when the blocklength is short, classic algebraic codes such as BCH and extended BCH codes can be decoded using near ML decoding algorithms. OSD, which we shall review next, is one such algorithm.

Recall that the evaluation of the ML decoding rule (8.16), which reduces to (8.18) in the bi-AWGN case, has in general a prohibitive complexity, already for small values of the information-payload size k. The idea behind OSD is to replace (8.18) with

$$\widehat{\jmath} = \arg\min_{m \in \mathcal{L}} \|\mathbf{y} - \sqrt{\rho}\mathbf{x}(m)\|^2, \tag{8.31}$$

where the optimization is performed over a list $\mathcal{L}$ of much smaller cardinality than 2^k. OSD constructs such a list through the following steps.

Let us consider the problem of decoding a (n, k) linear block code $\mathcal{C}$ that is used over the bi-AWGN channel (8.1). Let **G** be the generator matrix of the code, **u** the k-dimensional information vector, **c** the corresponding codeword, and **x** its vector representation in the Euclidean space after BPSK modulation.

For a given received vector **y**, we let $\mathbf{r} = [|y_1|, \dots, |y_n|]$; furthermore, we construct the additional vector $\mathbf{r}'$ by ordering the entries of **r** in decreasing order. Note that the scalars $\{y_\ell\}_{\ell=1}^{n}$ are proportional to the channel LLRs. Hence, the vector $\mathbf{r}'$ contains a scaled version of the channel LLRs ordered according to their reliability. Let π_1 be the permutation that maps $\mathbf{r}$ into $\mathbf{r}'$. If we apply this permutation to the columns of the generator matrix $\mathbf{G}$, we obtain a new generator matrix:

$$\mathbf{G}' = \pi_1(\mathbf{G}). \tag{8.32}$$

It turns out to be convenient to associate to each column of $\mathbf{G}'$ a reliability value, which is given by the corresponding entry of $\mathbf{r}'$. Next we rearrange the columns of $\mathbf{G}'$ so that the first k columns of this matrix are the k linear independent columns with the highest reliability, ordered in decreasing order of reliability, and the remaining $n-k$ columns are also ordered in decreasing order of reliability. We denote the resulting matrix by $\mathbf{G}''$ and the corresponding column permutation by π_2. Finally, we put $\mathbf{G}''$ in the systematic form $\mathbf{G}''' = \left[\mathbf{I}_k\ \mathbf{P}\right]$ by performing standard row operations. Here, $\mathbf{P}$ is of size $k \times n-k$. Note that the codes generated by $\mathbf{G}$ and by $\mathbf{G}'''$ are equivalent.

To construct the list, we now apply the second permutation π_2 to $\mathbf{r}'$ and obtain $\mathbf{r}'' = \pi_2(\mathbf{r}')$. Next, we perform a hard decision on the first k entries of $\mathbf{r}''$, to obtain the k-dimensional vector $\hat{\mathbf{u}}$. Specifically, we set $\hat{u}_i = 0$ if $z_i > 0$ and $\hat{u}_i = 1$ otherwise. It is worth remarking that the permutations ensure that the hard decision is performed on the most reliable linear-independent set of channel outputs.

For a given integer t, the list $\mathcal{L}$ is finally constructed by considering all codewords,

$$\hat{\mathbf{c}} = \pi_1^{-1}(\pi_2^{-1}(\tilde{\mathbf{u}}\mathbf{G}''')) \tag{8.33}$$

where $\tilde{\mathbf{u}}$ spans all k-dimensional vectors whose Hamming distance from $\hat{\mathbf{u}}$ is smaller or equal to t. The final decision is taken by computing the Euclidean distance between the BPSK vectors corresponding to each codeword in $\mathcal{L}$ and $\mathbf{y}$, and by selecting the codeword closest to $\mathbf{y}$.

The complexity of this procedure grows with the size of the list

$$|\mathcal{L}| = \sum_{m=0}^{t} \binom{k}{m}. \tag{8.34}$$

Clearly, the larger t, the larger the list and the better the performance of OSD; but also the greater the decoding complexity. Indeed, in the extreme case $t = k$, OSD coincides with ML decoding. In general, the value of t needed to approach ML decoding performance grows with the blocklength n. The decoding complexity can be reduced if the minimum distance $d_{\min}$ of the code is known. In such a case, one can stop the list construction procedure as soon as one finds a codeword whose corresponding BPSK vector has a Euclidean distance from $\mathbf{y}$ less than $\sqrt{\rho d_{\min}}$, since there cannot be a codeword that is closer to $\mathbf{y}$ than this.

8.2.5.2 Linear Block Codes With Tail-Biting Trellises

Short linear block codes can sometimes be represented efficiently by finite-length trellises with multiple initial and final states, in which each codeword corresponds to a tail-biting (TB) path with the same initial and final state. One example of codes that have this property and have also good distance spectra are linear block codes obtained through a tail-biting termination of suitably chosen convolutional codes.

The Viterbi algorithm can be used to decode such codes. However, since the initial state is not known, ML decoding requires running the Viterbi algorithm as many times as the number of initial states.

A suboptimal but lower-complexity decoding method is to use the so-called wrap-around Viterbi algorithm (WAVA). One assumes that all initial states are equiprobable and then runs the Viterbi algorithm one time. If the most likely path returned by the algorithm is a tail-biting path, decoding stops and this path is given as output. If the returned path is not tail-biting, the final state of the trellis is copied to the initial state and the Viterbi algorithm is run again.

The process is repeated until a tail-biting path is returned, or a maximum number of iterations is exceeded, in which case the decoder declares an error.

8.2.5.3 Nonbinary LDPC Codes

The performance under iterative decoding of LDPC codes in the short-blocklength regime can be significantly improved by constructing such codes on higher-order fields than $\mathbb{F}_2$ [12]. Low-complexity implementations of the required decoding algorithm are, however, still an active area of research.

8.2.5.4 Performance

We illustrate next the performance of the coding schemes described in Sections 8.2.5.1–8.2.5.3. We consider a short code of length 128 and rate $R = 1/2$. Hence, $k = 64$.

Fig. 8.8A, illustrates the performance of a $(128, 64)$ extended BCH code with minimum distance 22 [17]. The t parameter in the OSD decoder is set to 4, which results in a list of size 679121. In Fig. 8.8B, we consider three tail-biting convolutional codes with different memory m. Specifically, we analyze a memory 8 convolutional code with generator polynomial (given in octal form) $[515, 677]$, a memory 11 convolutional code with generator polynomial $[5537, 6131]$ and a memory 14 convolutional code with generator polynomial $[75063, 56711]$. We see from the figure that both the eBCH and the tail-biting convolutional codes with memory 11 and 14 operate remarkably close to the finite-blocklength normal approximation limit. The memory-8 convolutional code exhibits a loss of about 0.5 dB at 10^{-4} packet error probability. In Fig. 8.8C, we consider the performance of a nonbinary LDPC code constructed over a finite field of order 256. The PCM of the code has a constant row weight of 4 and a constant column weight of 2. We consider both iterative decoding with 200 iterations, and OSD with $t = 4$. One sees that the performances with OSD are close to the normal approximation, whereas the performance gap with iterative decoding is about 0.6 dB at 10^{-4}. Finally, in Fig. 8.8D we present the performance of a $(128, 71)$ polar code, combined with a $(71, 64)$ shorted cyclic code that serves as CRC. The list size is limited to 32. The performance gap to the normal approximation is about 0.4 dB at 10^{-4}.

8.3 CODING SCHEMES FOR FADING CHANNELS

So far, we have focused on the problem of transmitting information over the bi-AWGN channel (8.1). In this final section, we shall instead consider the more practically relevant scenario of communications over a multiantenna fading channel. The purpose is to illustrate the additional design challenges brought about by fading. Our focus will be on the short-packet regime and on the URLLC use case. We shall first discuss the single-input single-output (SISO) case and then move to multiple-input multiple-output (MIMO) transmissions.

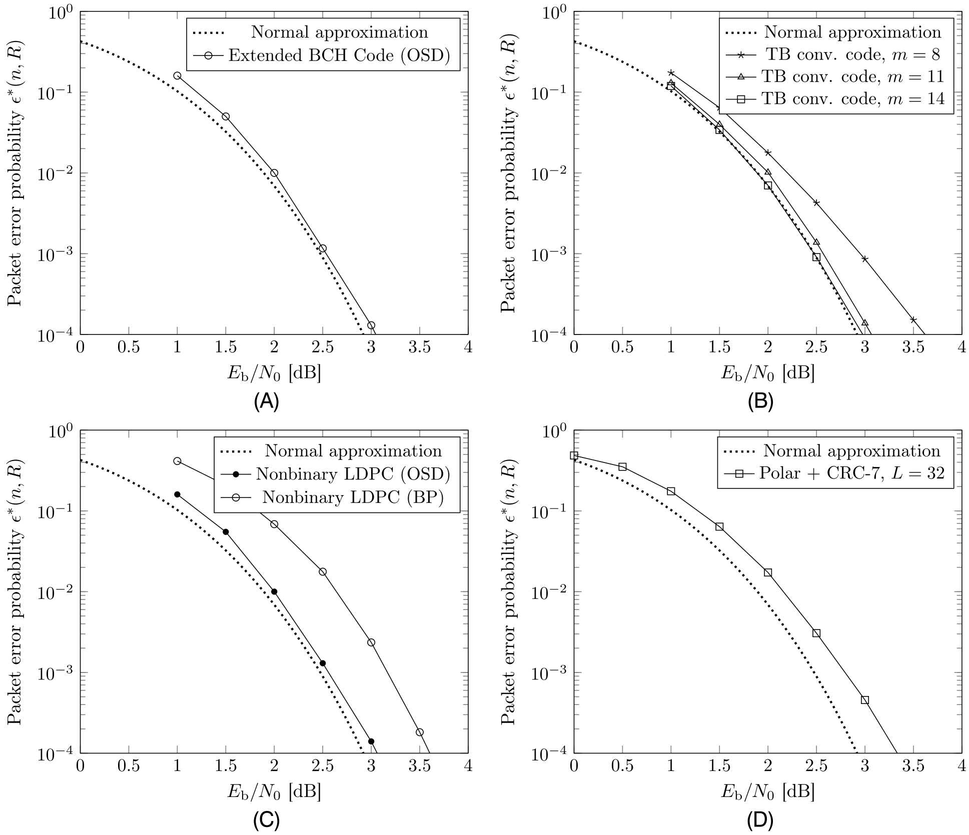

FIGURE 8.8

Performance of the coding schemes described in this section. Here, $R = 1/2$ and $n = 128$.

8.3.1 THE SISO CASE

We assume orthogonal frequency-division multiplexing (OFDM) operations and assume that each codeword spans multiple resource blocks (RBs), which are transmitted in the same time slot but at different frequencies. We assume that each RB contains d OFDM symbols consisting of u subcarriers. Hence, a RB conveys $n_c = d \times u$ complex-valued symbols. Low latency is achieved by selecting a sufficiently small value for d. For example, in downlink control channels, d may be chosen from the set $\{1, 2, 3\}$.

We assume that the coherence time T_c of the channel is larger than the transmission duration, and we let $L_{\max} = \lfloor B/B_c \rfloor$ be the ratio between the transmission bandwidth B and the coherence bandwidth of the channel B_c. Hence, $L_{\max}$ corresponds to the maximum number of frequency diversity branches offered by the channel. As we shall see, exploiting diversity is fundamental to achieve the reliability constraints set in URLLC.

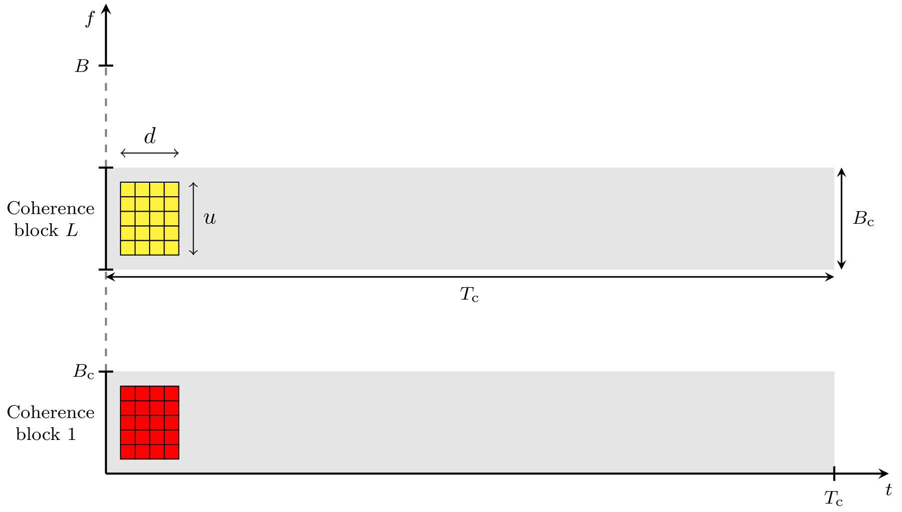

FIGURE 8.9

Signaling strategy in the time-frequency plane.

For simplicity, we model channel variations in frequency using the block-memoryless fading assumption. According to this assumption, the channel stays constant over each coherence interval and changes independently across diversity branches. This means that the fading channel is fully characterized by L_{max} complex channel coefficients.

As illustrated in Fig. 8.9, we assume that each RB fits within a coherence interval, i.e., all the complex symbols contained in the RB experience the same fading gain. We also assume that different RBs are allocated at different frequency branches. Finally, we assume that each codeword consists of $L \leq L_{\text{max}}$ RBs.

An example is in order. Assuming $B = 20$ MHz, as in LTE, we obtain $L_{\text{max}} = 4$ for the extended pedestrian type-A (EPA) 5 Hz channel model [2], whose coherence bandwidth is 4.4 MHz, whereas $L_{\text{max}} = 30$ for the tapped-delay-line type-C (TDL-C) 300 ms–3 km/h channel model [3], whose bandwidth is 0.66 MHz. In both cases, the coherence interval is about 85 ms, which exceeds by far the duration of a RB in all practically relevant scenarios. Indeed, recall that, with 15 kHz subspacing, the duration of an OFDM symbol is just 66.7 µs. We shall denote by $[\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_L] \in \mathbb{C}^{Ln_c}$ the vector of the transmitted complex symbols, where $n = L_c$ is the blocklength. The corresponding received vector is $[\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L]$ where $\mathbf{y}_\ell$ is the received signal corresponding to the ℓth RB; we have

$$\mathbf{y}_\ell = \sqrt{\rho} h_\ell \mathbf{x}_\ell + \mathbf{w}_\ell, \quad \ell = 1, \dots, L. \tag{8.35}$$

Here, $\{h_\ell\}$ are i.i.d. fading coefficients, in agreement with the memoryless block-fading assumption. We shall assume that the variance of the $\{h_\ell\}$ is normalized to one. Furthermore, $\mathbf{w}_\ell$ is the additive white complex-Gaussian noise whose entries are i.i.d. and have zero mean and unit variance.

We shall focus on pilot-assisted transmission [38] according to which n_p out of the n_c entries of a RB are allocated to pilot symbols known to the receiver, and the remaining entries are reserved for

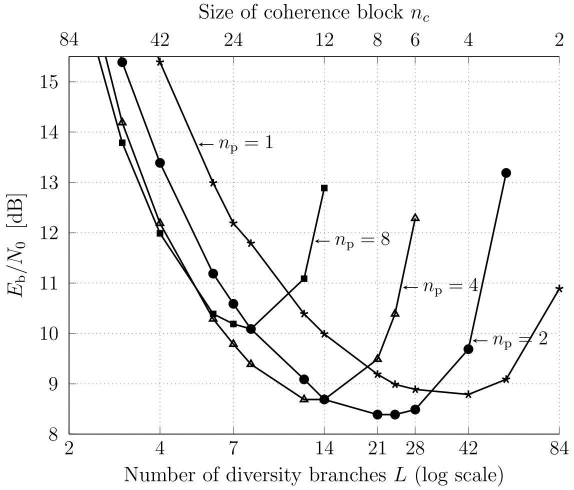

FIGURE 8.10

Minimum energy per bit E_b/N_0 to achieve $\epsilon = 10^{-3}$ for the Rayleigh-fading case, as a function of the number of diversity branches L. Here, $k = 81$ and $n = 168$.

coded data symbols. Specifically, we assume that $\mathbf{x}_\ell = [\mathbf{x}_\ell^{(p)}\mathbf{x}_\ell^{(d)}]$ where $\mathbf{x}_\ell^{(p)}$ is the n_p-dimensional pilot-symbol vector and $\mathbf{x}_\ell^{(d)}$ is the $n_c - n_p$ data vector. Throughout, we will focus on the scenario in which both pilot and data symbols are transmitted at the same power. More precisely, we assume that the entries of $\mathbf{x}_\ell^{(p)}$ and $\mathbf{x}_\ell^{(d)}$ are QPSK symbols with unit energy. QPSK modulation is indeed suitable for the low-rate low-power scenarios that are relevant for URLLC.

At the receiver side, we assume a practically relevant mismatch-decoding structure, in which the n_p-dimensional received vector $\mathbf{y}_\ell^{(p)}$ that corresponds to the pilot symbols is used to obtain a ML estimated $\hat{h}_\ell$ of the fading channel h_ℓ according to

$$\hat{h}_\ell = \mathbf{y}_\ell^{(p)} \frac{(\mathbf{x}_\ell^{(p)})^H}{n_p\sqrt{\rho}}. \tag{8.36}$$

Then the channel estimate is fed to a scaled minimum-distance decoder that treats it as perfect and produces a message estimate as follows:

$$\hat{j} = \underset{m\in\{1,2,\dots,2^k\}}{\arg\min} \sum_{\ell=1}^{L} \|\mathbf{y}_\ell^{(d)} - \hat{h}_\ell \mathbf{x}_\ell^{(d)}(m)\|^2. \tag{8.37}$$

The performance of this transceiver architecture has been recently studied in [27,15] using finite-blocklength information-theoretic methods similar to the ones described in Section 8.1. This kind of theoretical analyses provides useful insights on the optimal number of diversity branches L one should code over for a given fixed blocklength n, and on the optimal number of pilot symbols n_p that should be allocated in each resource block. This is illustrated in Fig. 8.10 where we have depicted

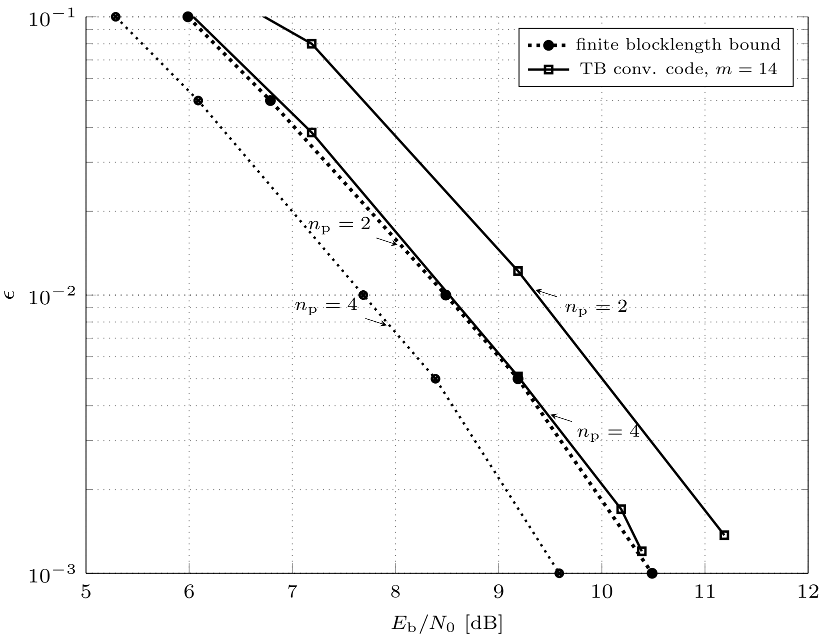

FIGURE 8.11

Packet error probability versus energy per bit for the case of Rayleigh fading, $k = 81$, $n = 186$, $L = 7$. Information-theoretic bounds [27, Th. 3] and performance of an actual coding scheme based on tail-biting convolutional codes and OSD.

the information-theoretic upper bound [27, Th. 3] on the minimum energy per bit E_{b}/N_0 required to achieve $\epsilon = 10^{-3}$ when transmitting $k = 81$ information bits. The blocklength $n = Ln_{\mathrm{c}}$ is 168 and a different number of diversity branches L is considered (the larger L, the smaller n_{c}).

We see that there exists an optimum number of diversity branches $L = 21$ that minimizes the energy per bit. When $L < 21$, the system is penalized by the insufficient amount of frequency diversity available, which makes reliable transmission costly from an energy viewpoint. When $L > 21$, and, hence $n_{\mathrm{c}} = n/\ell < 8$, the system suffers from the large pilot-symbol overhead, which is required to track the fast channel variations. We also observe that the number n_{p} of pilot symbols per resource block needs to be chosen carefully as a function of the number L of frequency diversity branches. Indeed, setting n_{p} to a suboptimal value yields a significant performance loss.

We next consider the performance of an actual coding scheme and benchmark it against the information-theoretic bounds. Specifically, we choose a $(324, 81)$ binary quasi-cyclic code obtained by tail-biting termination of a rate $1/4$ convolutional code with memory $m = 14$. The output of the encoder is passed through a pseudorandom interleaver and then some of the coded symbols are punctured to accommodate the desired number of pilot symbols per resource block (RB) after QPSK modulation. Decoding is performed via OSD with $t = 3$. The performance of this coding scheme when $L = 7$ is illustrated in Fig. 8.11 for the case $n_{\mathrm{p}} = 2$ and $n_{\mathrm{p}} = 4$. In both cases, the gap to the information-theoretic bound is about 1 dB.

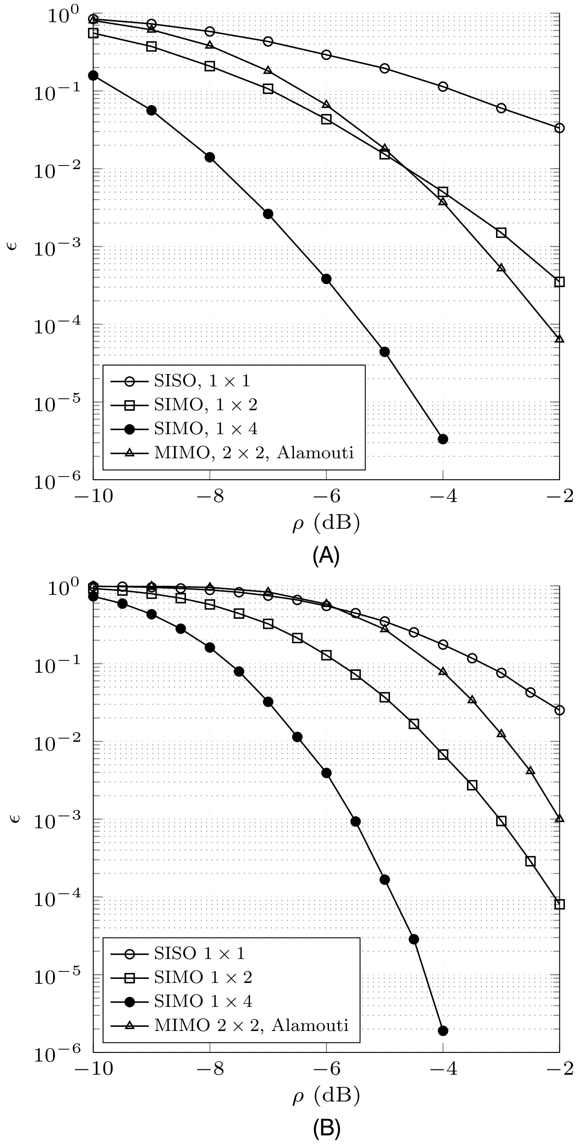

FIGURE 8.12

Packet error probability versus E_b/N_0; $k = 30$, $n = 288$, spatially white Rayleigh fading. (A) $L = 4$. (B) $L = 12$.

8.3.2 THE MIMO CASE

Exploiting the additional spatial diversity provided by MIMO transmission and reception is crucial to achieve the reliability level targeted by URLLC. The information-theoretic bounds depicted in Fig. 8.10

can be extended to MIMO communications, which allows one to explore the benefit of multiple antennas. As discussed in [15], the information-theoretic bounds can be extended to cover the case in which a space-frequency code is used at the transmitter to let the available antennas provide spatial diversity, in the absence of channel-state information at the transmitter.

In Fig. 8.12, we report the performance of different MIMO configurations for the case $k = 30$, which is relevant for downlink control-information transmission, and $n = 288$. We compare the performance of single-input single-output (SISO), 1 × 2 and 1 × 4 SIMO, and 2 × 2 MIMO with Alamouti encoding [4]. In Fig. 8.12A, we consider the case $n_c = 72$ and $L = 4$ (EPA 5 Hz channel model), whereas in Fig. 8.12B we set $n_c = 24$ and $L = 12$ (TDL-C channel model). In both cases, the number of pilot symbols is optimized to minimize the error probability. Furthermore, QPSK transmission is assumed.

We observe that, within the range of SNR values considered in the figure, only the 1 × 4 SIMO configuration is able to achieve an error probability below 10^{-5}, a common requirement in URLLC. Although the 2 × 2 MIMO Alamouti configuration offers the same amount of spatial diversity as 1 × 4 single-input multiple-output (SIMO), it is more sensitive to channel-estimation errors. This is particularly evident in Fig. 8.12B, where the small value of n_c results in a noisy channel estimate. As a consequence, the 2 × 2 MIMO Alamouti configuration performs worse than 1 × 2 SIMO over the range of SNR values considered in the figure.

REFERENCES

[1] 3GPP, TS 38.212 V15.0.0: multiplexing, and channel coding, 2017, Dec.

[2] 3GPP, TS 36.104: Technical specification group radio access network, 3GPP, 2012.

[3] 3GPP, TR 38.901: Study on channel model for frequencies from 0.5 to 100 GHz, 3GPP, 2017.

[4] S. Alamouti, A simple transmit diversity technique for wireless communications, IEEE Journal on Selected Areas in Communications (ISSN 0733-8716) 16 (8) (1998, Oct.) 1451–1458, https://doi.org/10.1109/49.730453.

[5] N. Alon, M. Luby, A linear time erasure-resilient code with nearly optimal recovery, IEEE Transactions on Information Theory 42 (11) (1996, Nov.) 1732–1736.

[6] E. Arıkan, Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels, IEEE Transactions on Information Theory 55 (7) (2009, Jul.) 3051–3073.

[7] E.E. Arıkan, I. Telatar, On the rate of channel polarization, in: Proc. IEEE Int. Symp. Inf. Theory (ISIT), Seoul, Korea, 2009, Jul., pp. 1493–1495, https://arxiv.org/abs/0807.3806.

[8] V. Bioglio, F. Gabry, I. Land, Low-complexity puncturing and shortening of polar codes, in: Proc. IEEE Wireless Commun. Netw. Conf., San Francisco, CA, U.S.A., 2017, Mar.

[9] G. Böcherer, F. Steiner, P. Schulte, Bandwidth efficient and rate-matched low-density parity-check coded modulation, IEEE Transactions on Communications 63 (12) (2015, Dec.) 4651–4665.

[10] A. Collins, G. Durisi, T. Erseghe, V. Kostina, J. Östman, Y. Polyanskiy, I. Tal, W. Yang, SPECTRE: short-packet communication toolbox, v2.0, https://github.com/yp-mit/spectre, 2016, Sep.

[11] D.J. Costello Jr., G.D. Forney Jr., Channel coding: the road to channel capacity, Proceedings of the IEEE 95 (6) (2007, Jun.) 1150–1177.

[12] M.C. Davey, D. MacKay, Low density parity-check codes over GF(q), IEEE Communications Letters 2 (6) (1998, Jun.) 165–167.

[13] D. Divsalar, S. Dolinar, C.R. Jones, K. Andrews, Capacity approaching protograph codes, IEEE Journal on Selected Areas in Communications 27 (6) (2009, Aug.) 876–888.

[14] T. Erseghe, Coding in the finite-blocklength regime: bounds based on Laplace integrals and their asymptotic approximations, IEEE Transactions on Information Theory 62 (12) (2016, Dec.) 6854–6883.

[15] G.C. Ferrante, J. Östman, G. Durisi, K. Kittichokechai, Pilot-assisted short-packet transmission over multiantenna fading channels: a 5G case study, in: Conf. Inf. Sci. Sys. (CISS), Princeton, NJ, 2018, Mar.

[16] G.D. Forney Jr., G. Ungerboeck, Modulation and coding for the linear Gaussian channels, IEEE Transactions on Information Theory 44 (6) (1998, Oct.) 2384–2415.

[17] M. Fossorier, S. Lin, Soft-decision decoding of linear block codes based on ordered statistics, IEEE Transactions on Information Theory (ISSN 0018-9448) 41 (5) (1995, Sep.) 1379–1396, https://doi.org/10.1109/18.412683.

[18] R. Gallager, Low-density parity-check codes, IRE Transactions on Information Theory 8 (1) (1962, Jan.) 21–28.

[19] G. He, J.C. Belfiore, X. Liu, Y. Ge, R. Zhang, I. Land, Y. Chen, R. Li, J. Wang, G. Yang, W. Tong, β-expansion: a theoretical framework for fast and recursive construction of polar codes, in: Proc. IEEE Global Telecommun. Conf. (GLOBECOM), Singapore, 2017, Dec., https://arxiv.org/abs/1704.05709.

[20] G. Liva, F. Steiner, pretty-good-codes.org: online library of good channel codes, http://pretty-good-codes.org/, 2017.

[21] G. Liva, W.E. Ryan, M. Chiani, Quasi-cyclic generalized LDPC codes with low error floors, IEEE Transactions on Communications 56 (1) (2008, Jan.) 49–57.

[22] D. MacKay, R. Neal, Good codes based on very sparse matrices, in: C. Boyd (Ed.), IMA Conf. Cryptography and Coding, Springer-Verlag, 1995, Oct.

[23] A. Martinez, A. Guillén i Fàbregas, Saddlepoint approximation of random-coding bounds, in: Proc. Inf. Theory Applicat. Workshop (ITA), San Diego, CA, U.S.A., 2011, Feb.

[24] M. Mondelli, S.H. Hassani, R. Urbanke, Unified scaling of polar codes: error exponent, scaling exponent, moderate deviations, and error floors, IEEE Transactions on Information Theory 62 (12) (2016, Dec.) 6698–6712.

[25] M. Mondelli, S.H. Hassani, R. Urbanke, Construction of polar codes with sublinear complexity, https://arxiv.org/abs/1612.05295, 2017, Jul.

[26] S.M. Moser, Information Theory (Lecture Notes), fifth ed., ETH Zurich/National Chiao Tung University, Switzerland/Taiwan, 2017, Mar.

[27] J. Östman, G. Durisi, E.G. Ström, M.C. Coşkun, G. Liva, Short packets over block-memoryless fading channels: pilot-assisted or noncoherent transmission?, https://arxiv.org/abs/1712.06387, 2017, Dec.

[28] S. Parkvall, E. Dahlman, A. Furuskär, M. Frenne, NR: the new 5G radio access technology, IEEE Communications Standards Magazine (2017, Dec.) 24–30.

[29] Y. Polyanskiy, H.V. Poor, S. Verdú, Channel coding rate in the finite blocklength regime, IEEE Transactions on Information Theory 56 (5) (2010, May) 2307–2359.

[30] T.J. Richardson, R. Urbanke, Modern Coding Theory, Cambridge Univ. Press, Cambridge, U.K., 2008.

[31] W.E. Ryan, S. Lin, Channel Codes: Classical and Modern, Cambridge Univ. Press, 2009.

[32] S. Sandberg, M. Andersson, A. Shirazinia, Y. Blankenship, LDPC Codes for 5G New Radio, 2018.

[33] I. Sason, S. Shamai (Shitz), Performance analysis of linear codes under maximum-likelihood decoding: a tutorial, Foundations and Trends in Communications and Information Theory (2006), https://doi.org/10.1561/0100000009.

[34] C. Schürch, A partial order for the synthesized channels of a polar code, in: Proc. IEEE Int. Symp. Inf. Theory (ISIT), Barcelona, Spain, 2016, Jul., pp. 220–224.

[35] C.E. Shannon, A mathematical theory of communication, The Bell System Technical Journal 27 (1948, July and October) 379–423, 623–656.

[36] I. Tal, A. Vardy, List decoding of polar codes, IEEE Transactions on Information Theory (ISSN 0018-9448) 61 (5) (2015, May) 2213–2226, https://doi.org/10.1109/TIT.2015.2410251.

[37] R.M. Tanner, A recursive approach to low complexity codes, IEEE Transactions on Information Theory 27 (9) (1981, Sep.) 533–547.

[38] L. Tong, B.M. Sadler, M. Dong, Pilot-assisted wireless transmissions, IEEE Signal Processing Magazine 21 (6) (2004, Nov.) 12–25, https://doi.org/10.1109/MSP.2004.1359139.

[39] G. Vazquez-Vilar, A.T. Campo, A.G. i Fàbregas, A. Martinez, Bayesian M-ary hypothesis testing: the meta-converse and Verdú-Han bounds are tight, IEEE Transactions on Information Theory 62 (5) (2016, May) 2324–2333.

[40] H. Zhang, R. Li, J. Wang, S. Dai, G. Zhang, Y. Chen, H. Luo, J. Wang, Parity-check polar coding for 5G and beyond, in: IEEE Int. Conf. Commun. (ICC), 2018, 01, https://arxiv.org/abs/1801.03616.