Low-Density Parity-Check Codes for Digital Subscriber Lines

E. Eleftheriou and S. Ölçer
IBM Research, Zurich Research Laboratory
8803 Rüschlikon, Switzerland

***Abstract*- The paper investigates the application of low-density parity-check (LDPC) codes to digital subscriber-line (DSL) transmission systems that employ discrete multitone modulation. A family of linear-time encodable binary LDPC codes that are well-suited for DSL transmission is introduced. Encoding and symbol mapping for multilevel modulation are described. Simulation results show that even under tight latency constraints good net coding gains can be achieved. Implementation complexity is analyzed and compared with that of trellis-coded modulation as employed in current asymmetric DSL transceivers. The incorporation of powerful LDPC coding techniques into next-generation DSL modems appears to be possible with reasonable increase in transceiver complexity.**

I. INTRODUCTION

Low-density parity-check (LDPC) codes [1,2] have mainly been considered for data transmission systems employing binary modulation. In many communication systems, however, multilevel modulation with more than two levels is employed to maximize the rate of information transfer under strict constraints on transmit signal bandwidth. An example is multicarrier digital-subscriber-line (DSL) transmission [3], where symbol constellations of possibly different sizes are used for quadrature amplitude modulation (QAM) on each subcarrier. The study of LDPC coding schemes that are suitable for bandwidth-efficient modulation represents, therefore, a topic of considerable practical interest.

In this paper, we describe an LDPC-coded multilevel modulation technique and investigate its application to DSL transmission. Binary LDPC codes are employed together with multilevel-symbol mapping based on set partitioning and so-called "double Gray-code labeling." Our approach differs from that in [4], where multilevel coding with binary LDPC component codes is proposed. We introduce a family of binary LDPC codes that offer good performance, are encodable in linear time, and do not suffer from error floors at significantly low bit-error rates. These LDPC codes can be constructed efficiently for any code rate and block size of interest for DSLs.

In current asymmetric DSL (ADSL) specifications [5], coding is achieved by a concatenated scheme that includes an outer Reed–Solomon (RS) code and an inner trellis code. Depending on the choice of code parameters and interleaving depth, this scheme can provide a net coding gain of up to ~5.5 dB with respect to uncoded modulation. LDPC coding, as described in this paper, is intended as a replacement of the inner trellis code with the objective of operating the ADSL link closer to its capacity limits than is currently possible. Our approach is applicable to both ADSL [6] and Very-high speed DSL (VDSL) systems.

In DSL transmission, overall delay, or latency, is a critical issue. "Voice" applications are known to demand rather low latency whereas other applications, such as video streaming, tolerate larger delays but need stronger error-correction capability. Thus, in studying new coding techniques for DSLs, trade-offs between coding gain and latency have to be well characterized. Another important issue is transceiver complexity. It is a critical parameter especially at the central-office access multiplexors or at remote terminals because it directly affects equipment cost and power consumption. LDPC coding is attractive for DSL transmission because it permits a wide range of trade-offs between latency, complexity, and system performance.

II. MULTICARRIER ADSL TRANSMISSION

The block diagram of Fig. 1 shows the components of a discrete-multitone (DMT)-based ADSL system that are relevant for the discussion in this paper.

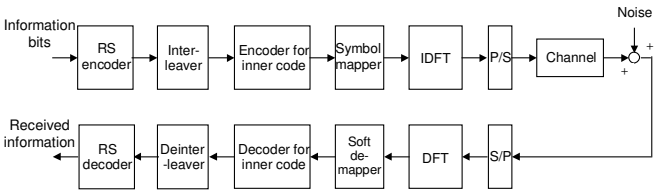

Fig. 1. DMT-based ADSL transmission system.

Information bits, representing user data and control messages, are encoded by an outer RS code with code symbols from GF(2^8), convolutionally interleaved, and further encoded by an inner-coding stage. In the current ADSL standard, the inner code is a four-dimensional 16-state trellis code. Here we investigate replacing the inner trellis code by an LDPC coding scheme[1]. In either case, the encoded data are mapped into frequency-domain modulation symbols and then transformed by an inverse discrete Fourier transform (IDFT) operation to yield a frame of time-domain signals. These signals are converted from parallel to serial (P/S) form and sent over the communication channel. At the receiver, the inverse of the transmit operations takes place to recover the information bits. The "soft demapper" block shown in Fig. 1, which is not needed in the case of trellis decoding by the Viterbi algorithm, computes soft information on LDPC code bits for subsequent soft iterative decoding.

The telephone-twisted-pair channel introduces frequency-dependent signal distortion as well as several other forms of disturbance, of which crosstalk is the most important. If each DMT subchannel has sufficiently narrow bandwidth, then each one independently approximates an additive white Gaussian noise (AWGN) channel with a particular signal-to-

[1] Outer RS coding is included in the above description because this function is mandatory in current ADSL specifications. We focus on the inner coding scheme.

0-7803-7400-2/02/$17.00 © 2002 IEEE

noise ratio (SNR) [3]. Impulse noise represents a further source of disturbance that an ADSL system must be able to cope with. Finally, we note that narrowband interference of various origins, e.g., AM radio signals, also affects the reliability of communications in ADSL.

III. LDPC Parity-Check Matrix Construction

For ADSL transmission, LDPC codes with high code rates are desirable. Besides achieving high spectral efficiencies in a bandwidth-constrained transmission situation, such codes involve a smaller amount of parity checks than low-rate codes do, resulting in more tractable decoder implementations at the envisaged multi-megabit-per-second data rates. It is also desirable that the generation of the parity-check matrix involves a small amount of preprocessing operations, rendering "on-the-fly" construction of LDPC codes practical. Furthermore, linear-time encodable LDPC codes are attractive because low implementation complexity can also be achieved at the transmitter. Finally, the ability to specify the LDPC codes via a small number of parameters is critical because it minimizes the overhead during initialization, when the receiver must indicate to the transmitter which LDPC code to use for encoding. Codes that can be described by a small number of parameters are also well suited for standardization purposes.

The deterministic parity-check matrix construction presented in this section meets the above objectives. The construction is based on "array codes," which are two-dimensional codes that have been proposed for detecting and correcting burst errors [7]. When array codes are viewed as binary codes, their parity-check matrices exhibit sparseness, which can be exploited for decoding them as LDPC codes using the sum-product algorithm [8]. Therefore, array codes provide the framework for defining a family of LDPC codes that lend themselves to deterministic constructions.

The array-code parity-check matrix is specified by three parameters: a prime number p and two integers k and j such that $k, j \le p$. It has dimensions $jp \times kp$ and is given by [7]

$$\mathbf{H}_{\mathrm{A}} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ \mathbf{I} & \boldsymbol{\alpha} & \boldsymbol{\alpha}^2 & \cdots & \boldsymbol{\alpha}^{k-1} \\ \mathbf{I} & \boldsymbol{\alpha}^2 & \boldsymbol{\alpha}^4 & \cdots & \boldsymbol{\alpha}^{2(k-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{I} & \boldsymbol{\alpha}^{j-1} & \boldsymbol{\alpha}^{2(j-1)} & \cdots & \boldsymbol{\alpha}^{(j-1)(k-1)} \end{bmatrix}, \qquad (1)$$

where $\mathbf{I}$ is the $p \times p$ identity matrix and $\boldsymbol{\alpha}$ is a $p \times p$ permutation matrix representing a single left or right cyclic shift. For example, for $p = 5$,

$$\boldsymbol{\alpha} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{or} \quad \boldsymbol{\alpha} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \qquad (2)$$

The parameters j and k provide the column and row weight of $\mathbf{H}_{\mathrm{A}}$, respectively. By construction, the matrix $\mathbf{H}_{\mathrm{A}}$ is 4-cycle free because no two rows have overlapping "1"s in more than one position.

To achieve efficient encoding, a parity-check matrix in triangular form is desirable, see, e.g., [9]. Although Gaussian elimination could be used to this end, the resulting increase in processing complexity makes this approach inattractive. Instead, we define a new matrix $\mathbf{H}_{\mathrm{S}}$ by cyclically shifting the rows of the matrix $\mathbf{H}_{\mathrm{A}}$ in a blockwise manner. The amount of cyclic shift for each block row is such that the $jp \times jp$ leftmost subblock of $\mathbf{H}_{\mathrm{S}}$ contains the identity matrix $\mathbf{I}$ along its diagonal:

$$\mathbf{H}_{\mathrm{S}} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ \boldsymbol{\alpha}^{k-1} & \mathbf{I} & \boldsymbol{\alpha} & \cdots & \boldsymbol{\alpha}^{j-2} & \boldsymbol{\alpha}^{j-1} & & \boldsymbol{\alpha}^{k-2} \\ \boldsymbol{\alpha}^{2(k-2)} & \boldsymbol{\alpha}^{2(k-1)} & \mathbf{I} & \cdots & \boldsymbol{\alpha}^{2(j-3)} & \boldsymbol{\alpha}^{2(j-2)} & & \boldsymbol{\alpha}^{2(k-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ \boldsymbol{\alpha}^{(j-1)(k-j+1)} & \boldsymbol{\alpha}^{(j-1)(k-j+2)} & \cdots & \cdots & \mathbf{I} & \boldsymbol{\alpha}^{j-1} & \cdots & \boldsymbol{\alpha}^{(j-1)(k-1)} \end{bmatrix} \qquad (3)$$

The matrix $\mathbf{H}_{\mathrm{S}}$ is 4-cycle free and has same column and row weight as $\mathbf{H}_{\mathrm{A}}$.

To obtain the parity-check matrix in the desired form, the lower-triangular elements of the $jp \times jp$ leftmost subblock of $\mathbf{H}_{\mathrm{S}}$ are set to zero, yielding

$$\mathbf{H} = \begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} & \mathbf{I} & \cdots & \mathbf{I} \\ \mathbf{O} & \mathbf{I} & \boldsymbol{\alpha} & \cdots & \boldsymbol{\alpha}^{j-2} & \boldsymbol{\alpha}^{j-1} & & \boldsymbol{\alpha}^{k-2} \\ \mathbf{O} & \mathbf{O} & \mathbf{I} & \cdots & \boldsymbol{\alpha}^{2(j-3)} & \boldsymbol{\alpha}^{2(j-2)} & & \boldsymbol{\alpha}^{2(k-3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & & & \vdots \\ \mathbf{O} & \mathbf{O} & \cdots & \mathbf{O} & \mathbf{I} & \boldsymbol{\alpha}^{j-1} & \cdots & \boldsymbol{\alpha}^{(j-1)(k-1)} \end{bmatrix}, \qquad (4)$$

where $\mathbf{O}$ is the $p \times p$ null matrix.

By successive row and column permutations, $\mathbf{H}_{\mathrm{A}}$ and $\mathbf{H}_{\mathrm{S}}$ can brought be to a form similar to that defined in [10]. Therefore, using similar counting arguments as in [10], it can be shown that $\mathbf{H}_{\mathrm{A}}$ and $\mathbf{H}_{\mathrm{S}}$ both lead to a minimum Hamming distance of $d_{\min} = 6$ for $j = 3$ and $d_{\min} \ge 8$ for $j = 4$ (these are the values of j of most practical interest). Furthermore, for $j = 3$, it can be shown that forcing $\mathbf{H}_{\mathrm{S}}$ to the triangular form $\mathbf{H}$ does not decrease the minimum distance of the code. This property is conjectured to also hold for $j = 4$, but could only be verified via an exhaustive search for codes employed in the simulations.

The LDPC code defined by $\mathbf{H}$ has code-word length $N = kp$, number of parity checks $M = jp$, and information block length $K = (k - j)p$. An LDPC code with $N' < N$ is easily obtained by discarding the $N - N'$ rightmost columns of $\mathbf{H}$. The parity-check matrix $\mathbf{H}$ is fully determined by p, j, k, and the code length N'.

Efficient encoding is achieved directly from $\mathbf{H}$ without the need to compute the generator matrix of the code. Recall that

because LDPC codes are linear block codes, an N-tuple $\mathbf{x}$ is an LDPC code word if and only if $\mathbf{H} \cdot \mathbf{x}^{\mathrm{T}} = \mathbf{0}^{\mathrm{T}}$. Let us express the vector $\mathbf{x}$ in the form $\mathbf{x}^{\mathrm{T}} = [\mathbf{p}^{\mathrm{T}}\, \mathbf{s}^{\mathrm{T}}]$, where the $jp \times 1$ vector $\mathbf{p}$ represents the *parity* part and the $(k - j)p \times 1$ vector $\mathbf{s}$ represents the *systematic* part of the code word $\mathbf{x}$. Then, it is easy to see that the jp parity bits in $\mathbf{p}$ can be obtained in a recursive manner by employing $\mathbf{H} \cdot [\mathbf{p}^{\mathrm{T}}\ \mathbf{s}^{\mathrm{T}}] = \mathbf{0}^{\mathrm{T}}$ and exploiting the upper-triangular form of $\mathbf{H}$. This encoding process can be shown to require $(N/2)\,[r\,(j + 3) + (j - 3)]$ XOR operations, where r is the rate of the code. Hence, the code is linear-time encodable.

IV. Bandwidth-Efficient LDPC-Coded Modulation

Soft demapping at the receiver is greatly simplified if square-shaped constellations are employed, because the real and imaginary parts of the received noisy complex signals can then be demapped independently. We thus assume that transmit symbols are chosen from a 2^b-QAM symbol set, where the integer $b = 1$ or $b > 1$ and even. The block diagram of the multilevel encoding and symbol mapping functions is shown in Fig. 2.

When $b > 1$ and even, the two binary $b/2$-tuples ($v_{b/2-1}$, $v_{b/2-2}$, …, v_1, v_0) and ($w_{b/2-1}$, $w_{b/2-2}$, …, w_1, w_0) independently select two L-ary real symbols, $L = 2^{b/2}$, representing the real and imaginary parts, respectively, of the complex QAM symbol to be transmitted. The L-ary symbols belong to the set

$$\mathbf{A} = \{A_\ell = 2\ell - (L-1),\quad \ell = 0, 1, ..., L-1\}. \tag{5}$$

Each 2^b-QAM symbol conveys b_{cv} and b_{cw} LDPC code bits on its real and imaginary parts, respectively; the remaining bits are uncoded. It is generally sufficient to allow up to six code bits per QAM symbol for best trade-off in terms of spectral efficiency, performance, and implementation complexity.

Symbol mapping relies on the partition of the set $\mathbf{A}$ into $2^{b_{\mathrm{cv}}}$ $[2^{b_{\mathrm{cw}}}]$ subsets such that the minimum Euclidean distance between the symbols within each subset is maximized.

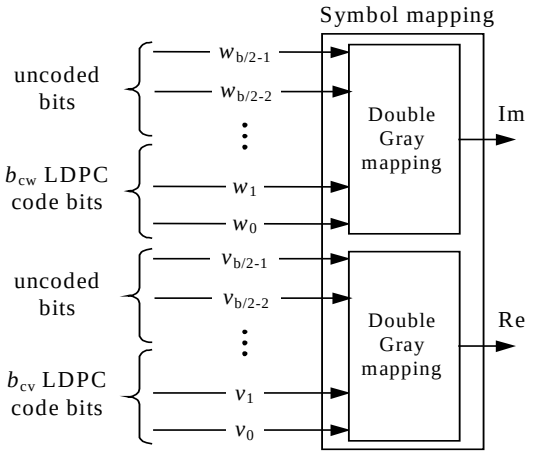

Fig. 2. Multilevel LDPC encoding and symbol mapping for 2^b QAM.

TABLE I. Example of symbol labeling for the case of 256 QAM (L = 16), $b_{\mathrm{cv}} = b_{\mathrm{cw}} = 2$.

L-ary symbol	v_3 (w_3)	v_2 (w_2)	v_1 (w_1)	v_0 (w_0)
+15	1	0	1	0
+13	1	0	1	1
+11	1	0	0	1
+9	1	0	0	0
+7	1	1	1	0
+5	1	1	1	1
+3	1	1	0	1
+1	1	1	0	0
–1	0	1	1	0
–3	0	1	1	1
–5	0	1	0	1
–7	0	1	0	0
–9	0	0	1	0
–11	0	0	1	1
–13	0	0	0	1
–15	0	0	0	0

The b_{cv} $[b_{\mathrm{cw}}]$ least-significant bits (LSBs) of v $[w]$ label the subsets of $\mathbf{A}$ following a Gray-coding rule. The remaining most-significant bits (MSBs) label symbols within a subset following a separate Gray-coding rule. Table I gives an example of this *double Gray-code labeling* technique for the case of 256 QAM (L = 16), and $b_{\mathrm{cv}} = b_{\mathrm{cw}} = 2$. When $b = 1$, only the code bit v_0 is employed. This case corresponds to BPSK modulation.

We note that Gray-code labeling is optimum in an information-theoretic sense as it leads to largest capacity for bit-interleaved coded modulation [11,12]. Intuitively, from the observation of a noisy symbol, the most reliable soft information on each underlying bit can be generated if Gray-code labeling is used because here the variation of a symbol value between two adjacent levels corresponds to flipping a single bit only. Gray-code labeling is thus adopted for the LSBs on which the soft demapper needs to generate reliability information. Furthermore, as the uncoded MSBs are obtained via simple thresholding at the receiver, labeling those bits with a (separate) Gray code within each subset allows lowering the bit-error rate on the MSBs.

Let now y denote the real part of a noisy received signal:

$$y = A + n, \tag{6}$$

with $A \in \mathbf{A}$ and n an AWGN sample with variance σ_n^2 (the imaginary part of the received signal is processed similarly). The soft demapper shown in Fig. 1 computes the *a posteriori* probability (APP) of code bit i_m being equal to $x = 0$, 1 as

$$\Pr(i_m = x \mid y) = \frac{\displaystyle\sum_{\mathrm{A}_\ell \in \mathbf{A}_{m,x}} e^{-\frac{(y-\mathrm{A}_\ell)^2}{2\sigma_{\mathrm{n}}^2}}}{\displaystyle\sum_{\mathrm{A}_\ell \in \mathbf{A}} e^{-\frac{(y-\mathrm{A}_\ell)^2}{2\sigma_{\mathrm{n}}^2}}},\ m = 0,1,\ldots,b_{cv}-1, \tag{7}$$

where $\mathbf{A}_{m,x}$ denotes the set of symbols A for which $i_m = x$. If sum-product decoding is based on the computation of log-likelihood ratios, then $\ln\left[\Pr(i_m = 0 \mid y)/\Pr(i_m = 1 \mid y)\right]$ can be employed. For a practical implementation, it is not necessary to include all the terms in the summations in Eq. (7). Given a received signal $|y| < L-1$, it is usually sufficient to determine the two closest nominal symbols A_ℓ, and include only those in the summation terms. If the received signal does not fall within the constellation boundaries, i.e., $|y| \geq L-1$, then the APP is set to 1 or to 0, depending on the symbol found at the constellation edge. In this way, the computational effort for soft demapping is not only reduced but also made essentially independent of the constellation size L.

The (approximate) channel APPs generated in this manner are finally used in the sum-product algorithm (SPA) [1,2] for soft iterative decoding. Note that various simplifications of the SPA have been proposed in the literature. For example, the simplified algorithm presented in [13] operates entirely in the log-likelihood-ratio domain and offers a substantial reduction in complexity with essentially the same performance as the full SPA.

V. Implementation Complexity

In this section, we compare, for a particular example, the implementation complexity of the proposed LDPC coding scheme with that of trellis-coded modulation (TCM) as specified in [5].

A. Encoding Complexity

Let us consider a DMT system with 200 tones (subchannels) and 16 QAM on each tone. As computational complexity for each trellis-encoding step amounts to 7 XOR operations for TCM, a total complexity of $100 \times 7 = 700$ XOR operations per DMT symbol is obtained. For LDPC coding, we need a code of length $200 \times 4 = 800$ bits in this case. An appropriate choice is the code with $j = 3$, $k = 25$ and rate $r = K/N \simeq 0.8863$, resulting in a complexity of 2127 XOR operations per DMT symbol. Therefore, the complexity of LDPC encoding is about three times that of TCM encoding.

B. Decoding Complexity

Consider again the above example. The computational complexity of the trellis decoding approximately amounts to 119 additions and 4 multiplications per trellis step, not accounting for the complexity of subset decoding and the updating of survivor sequences (backtracing).

Using the algorithm in [13], the complexity for LDPC decoding amounts to $3(k-2)+2j$ additions per iteration, assuming a block-parallel implementation. Furthermore for soft demapping, 8 multiplications and 6 additions are required per QAM symbol. The complexity of decoding for uncoded bits, which we did not account for, can be assumed to be similar to that of subset decoding in TCM. Using the same LDPC code parameters as above, assuming 20 iterations for soft decoding and a DMT-symbol rate of 4000 Hz, the results of Table II are obtained.

TABLE II. Decoding complexity for LDPC-coded modulation and TCM.

	Additions/s	Multiplications/s
TCM	47.6 M	1.6 M
Soft demapping + LDPC decoding	4.8 M + 6.0 M	6.4 M + 0

Note that if an LDPC code spanning more than one DMT symbol is used, complexity due to sum-product decoding will grow (which, fortunately, represents the less intensive part of the decoding process) whereas soft-demapping complexity will remain fixed because soft demapping is performed on a DMT-symbol basis.

C. Memory Requirements

For TCM, a memory size of about $20 \times 2 \times 16 = 640$ words, where the factor 20 accounts for 5 constraint lengths, is needed to store the survivor sequences for 16-state Viterbi decoding.

For the above LDPC code example, the parity-check matrix has $800 \times 3 = 2400$ nonzero entries, the locations of which have to be stored for encoding purposes. Assume, for decoding with a fully parallel and pipelined structure, that each memory block is implemented as two buffers alternating between read and write. Then, the required memory for sum-product decoding is $4 \times 2400 = 9600$ words. Clearly, longer codes will lead to more stringent memory requirements.

VI. Simulation Results

A. Performance in AWGN

In the simulations, the full SPA is employed with the number of iterations limited to 20. We represent bit and block-error rates as a function of E_b/N_0, the ratio of energy per-bit to noise power-spectral-density, and symbol-error rates as a function of the normalized SNR. Recall that for a modulation and coding scheme transmitting η bit/symbol, the normalized SNR is defined as [14]

$$\mathrm{SNR}_{\mathrm{norm}} = \frac{\eta}{2^\eta - 1}\,\frac{E_b}{N_0}\ . \tag{8}$$

For uncoded QAM, $\mathrm{SNR}_{\mathrm{norm}} \approx 9.8$ dB at a symbol-error rate of 10^{-7}.

Figs. 3 to 5 show the bit-error rate (BER) and block-error rate (BLER) performance of three LDPC codes for binary transmission over the AWGN channel. The codes have lengths $N = 529$, 2209, and 4489, and assume $j = 3$, $j = 4$, and $j = 4$, respectively. Uncoded performance and capacity are also plotted in these figures.

The performance achieved is as good as or better than the performance of the randomly constructed LDPC codes [2] of comparable lengths and rates. Note also the absence of error floors at error rates of 10^{-7}, which are of interest for ADSL. It is therefore expected that good performance will also be achieved for multilevel modulation. Figs. 6 to 8 show the symbol-error rate performance for 16, 256, and 4096 QAM over the AWGN channel using the three LDPC codes.

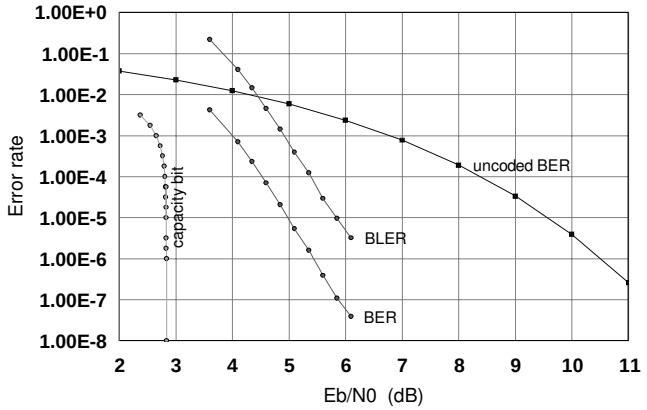

Fig. 3. Performance of LDPC code (529, 460) for binary transmission over the AWGN channel.

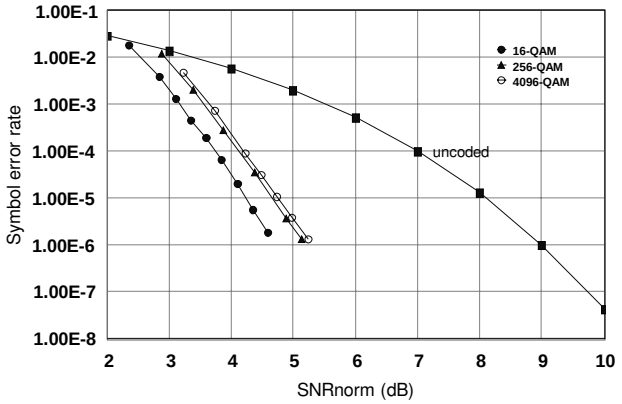

Fig. 6: Performance of LDPC code (529, 460) for transmission over the AWGN channel using 16, 256, and 4096 QAM.

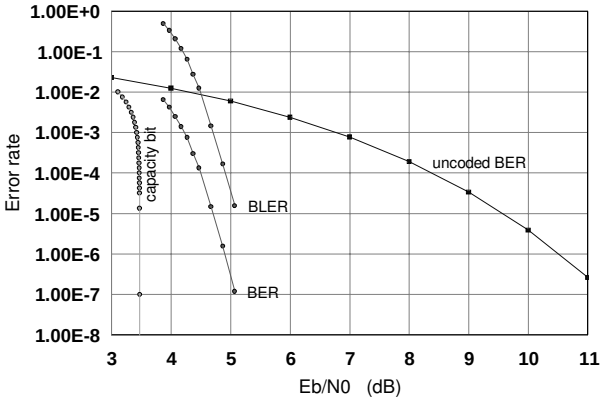

Fig. 4. Performance of LDPC code (2209, 2021) for binary transmission over an AWGN channel.

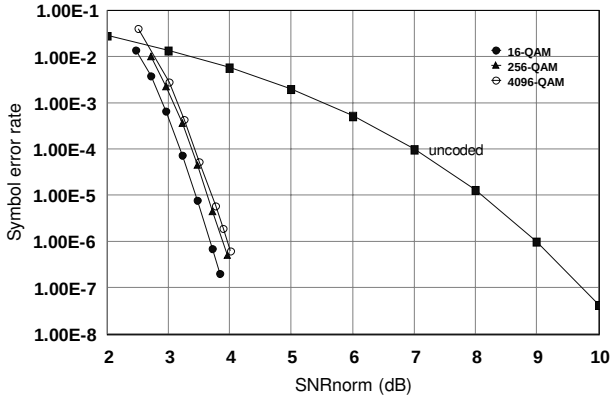

Fig 7: Performance of LDPC code (2209, 2021) for transmission over the AWGN channel using 16, 256, and 4096 QAM.

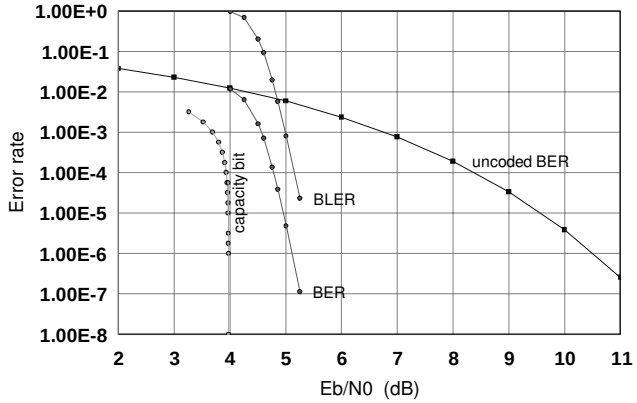

Fig. 5. Performance of LDPC code (4489, 4221) for binary transmission over an AWGN channel.

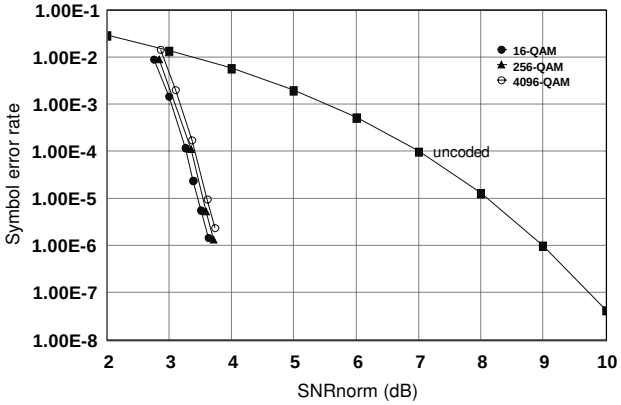

Fig. 8: Performance of LDPC code (4489, 4221) for transmission over the AWGN channel using 16, 256, and 4096 QAM.

Fig. 9 shows the performance of the (2209, 2021) LDPC code in the spectral-efficiency versus power-efficiency plane, at a BER of 10^{-7} (triangles). The figures incorporate the capacity of the employed signal sets (squares), shedding light onto the effectiveness of the proposed LDPC coding scheme. The gap between the capacity limit and power efficiency of the LDPC schemes remains fairly constant, nearly independently of the spectral efficiency. Similar results have been obtained for the other LDPC codes and other spectral efficiencies but, for space reason, are not shown here.

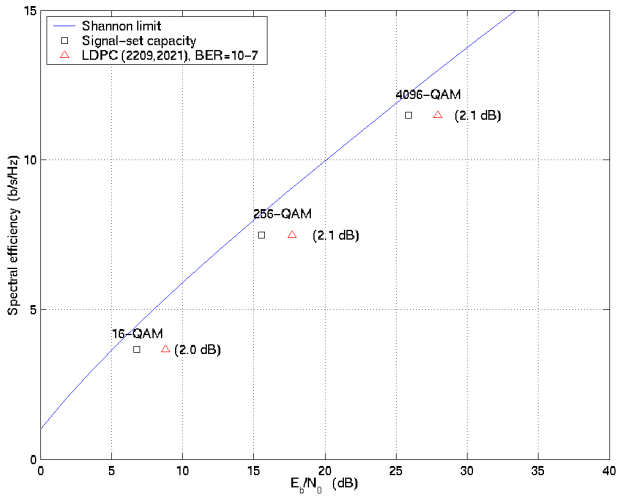

Fig. 9: Performance of LDPC code (2209, 2021) for various spectral efficiencies. The numbers in parentheses indicate the gap in E_b/N_0 between the coded scheme and the signal-set capacity at BER = 10^{-7}.

B. Performance in AWGN with Latency Constraints

To determine the net coding gain as function of coding latency, we consider a DMT system with a total number of 100 or 200 tones. DMT symbols are assumed to be sent at the nominal rate of 4000 Hz, as is the case for ADSL.

The results summarized in Table III show the net coding gains in dB at a symbol-error rate of 10^{-7} for different values of coding latency (no outer RS code). We did not run simulations for codes longer than 7200 bits, hence some entries in the table are not provided. The code rates were chosen in the range of 0.82 to 0.95. It can be seen that good coding gains can be achieved even for very tight latency constraints.

TABLE III. NET CODING GAINS (IN DB) ACHIEVED BY SELECTED LDPC CODES AS A FUNCTION OF LATENCY.

Modulation	**# of tones**	**Latency (in ms)**					
		0.5	**1**	**2**	**4**	**6**	**8**
16 QAM	100	4.5	4.8	5.2	6.0	6.1	6.2
	200	4.8	5.2	6.0	6.2	–	–
4096 QAM	100	4.1	4.6	5.2	5.9	6.1	–
	200	4.6	5.2	5.9	–	–	–

VII. CONCLUSIONS

LDPC codes are finding their way into a number of applications, e.g., for wireless communications and storage channels. They also offer unique advantages for DSL transmission.

The simulation results presented here show that, even under tight latency constraints, good net coding gains can be achieved by LDPC coding. Furthermore, LDPC codes do not exhibit "error floors" at the low bit-error rates of interest for DSL transmission. Another advantage is their low implementation complexity as compared, for example, to turbo codes. In fact, many implementation trade-offs are possible owing to the inherent parallelism in the sum-product algorithm, opening the way for very-low-power VLSI realizations.

Clearly, further study is needed to fully characterize the benefits of LDPC coding for DSLs, including VDSL, and to assess performance with actual loop and noise characteristics. However, the incorporation of powerful LDPC coding techniques into next-generation DSL modems appears to be attractive in terms of performance gains and also possible at only a reasonable increase in transceiver complexity.

REFERENCES

[1] R. G. Gallager, "Low-density parity-check codes," *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21-28, Jan. 1962.

[2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inform. Theory*, vol. 45, No. 2, pp. 399-431, Mar. 1999.

[3] T. Starr, J. M. Cioffi, and P. J. Silverman, *Digital Subscriber Line Technology*, Upper Saddle River, NJ: Prentice Hall, 1999.

[4] K. Narayan and J. Li, "Bandwidth efficient low density parity check coding using multilevel coding and iterative multistage decoding," in *Proc. 2nd Int'l Symposium on Turbo Codes and Related Topics*, Brest, France, pp. 165-168, Sept. 2000.

[5] "Asymmetric digital subscriber line (ADSL) transceivers," ITU-T Recommendation G.992.1, June 1999.

[6] E. Eleftheriou and S. Ölçer, "Proposed text on LDPC coding for inclusion in Draft Recommendation," Document SC-065, ITU-Telecommunication Standardization Sector, Study Group 15, San Francisco, CA, Aug. 2001.

[7] M. Blaum, P. Farrell, and H. van Tilborg, "Array codes," in *Handbook of Coding Theory*, V. S. Pless and W. C. Huffman Eds., Elsevier 1998.

[8] J. L. Fan, "Array codes as low-density parity-check codes," in *Proc. 2nd Int'l Symposium on Turbo Codes and Related Topics*, Brest, France, pp. 543-546, Sept. 2000.

[9] T. J. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, No. 2, pp. 638-656, Feb. 2001.

[10] D. Hösli, E. Svensson, and D. Arnold, "High-rate low-density parity-check codes: construction and application," in *Proc. 2nd Int'l. Symposium on Turbo Codes and Related Topics*, Brest, France, pp. 447-450, Sept. 2000.

[11] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, No. 3, pp. 927-946, May 1998.

[12] E. Eleftheriou, X. Hu, and S. Ölçer, "An information-theoretic framework for comparing the coding schemes proposed for G.dmt.bis and G.lite.bis," Temporary Document IC-070, ITU-T, Study Group 15, Question 4, Irvine, CA, Apr. 9-13, 2001.

[13] E. Eleftheriou, T. Mittelholzer, and A. Dholakia, "Reduced-complexity decoding algorithm for low-density parity-check codes," *Electron. Lett*., vol. 37, no 2, pp. 102-104, Jan. 2001.

[14] M. V. Eyuboglu and G. D. Forney, "Trellis precoding: combined coding, precoding and shaping for intersymbol interference channels," *IEEE Trans. Inform. Theory*, vol. 38, No. 2, pp. 301-314, Mar. 1992.