

Almost-Noiseless Joint Source-Channel Coding-Decoding of Sources with Memory

GIUSEPPE CAIRE¹, SHLOMO SHAMAI², SERGIO VERDÚ³

¹ Institute Eurecom, France Giuseppe.Caire@eurecom.fr

² Technion, Israel Sshlomo@ee.technion.ac.il

³ Princeton University, USA Verdu@princeton.edu

Abstract

The design of joint source-channel encoders and decoders to obtain low block error rate is considered in this paper. Linear encoders based on various low-density structures, and in particular the new class of Lotus codes are considered along with belief propagation decoders. We extend the schemes we recently introduced to design universal linear data compressors for sources with memory to the case of transmission through a noisy channel. The resulting encoders and decoders have linear complexity in the length of the source block and are particularly effective in the moderate blocklength regime relative to the conventional separate source-channel encoding approach. The availability of a modicum of feedback can be very beneficial to simplify the complexity and blocklength required to achieve a predetermined reliability level. Several new feedback schemes tailored to belief propagation decoding are proposed.

1 Introduction

In 1948, Shannon [1] showed that arbitrary reliability can be achieved if the source entropy is less than the channel capacity by separating the encoder into a (channel-independent) source encoder followed by a (source-independent) channel encoder and likewise for the decoder. The proof that separation of source compression and channel coding entails no loss of optimality while achieving arbitrary reliability for asymptotically long blocklength had to await the discovery of Fano's inequality. Known to hold in wide generality for sources and channels with memory, the "separation principle" is known to fail in certain situations when the source and the channel are nonstationary [2], as well as in multiterminal networks.

While important practical systems (such as facsimile and dialup modem standards) adhere to the separation principle, state-of-the-art packet-based high-speed data third-generation wireless systems do not. In fact, these systems incur in severe loss of efficiency as data is sent uncompressed. This pragmatic design choice is made because transmission in packets of moderate fixed length is ill-suited to existing data compression algorithms which are sensitive to error propagation.

Among the several motivations for adopting a joint approach to source-channel coding rather than the conventional separated approach is the hope that it may offer a more favorable performance-delay trade-off in the nonasymptotic regime. Despite this widely recognized target of opportunity, except for sources with very simple statistics, to date no approach is

known to effectively compete with the conventional separation-based approach. Existing data compression approaches based on either parsing (LZ) or arithmetic coding (PPM,CTW) require causal delay-free recovery of the data for decompression, which is cumbersome in the context of joint source/channel decoding. Works that build in some resilience to errors on one of those standard data compression algorithms include [3], [4], [5]. Instead our approach is different and follows the paradigm of *linear* encoding which is known to achieve the Shannon limit (e.g. [6]).

In many data transmission applications it is desired that the data be recovered almost noiselessly (i.e. with very low block error rate) despite the presence of a noisy channel. However, most of the work in the area of joint source/channel coding has focused on lossy recovery of the signal (where a nonvanishing per-letter distortion is allowed). In this work, our performance measure is block error rate. Note that in the lossy setting, joint linear encoding is no longer capable of achieving the Shannon limit asymptotically [7].

Maximum likelihood decoders do not give the most probable decisions when the messages fed to the channel encoder are not equiprobable. As noted by Shannon [1], if the input to the channel encoder has residual redundancy because data compression has been in-existent or incomplete, and its distribution is known at the decoder, then it is possible to take the source distribution into account in order to improve block error probability or bit error rate performance, and consequently achieve reliable communication exceeding the rate that the decoder would support for equiprobable messages. For some decoding algorithms it is straightforward to

³Partially supported by NF Grant CC-0312879

incorporate source distribution information:

- *Viterbi decoding*: Nonequiprobable marginals, or even Markov dependence in the source symbols, can be incorporated in the computation of the path metrics as noted by Forney [8] and Hagenauer [9] in the context of hard and soft decisions, respectively.
- *Turbo decoding*: The bias of a binary memoryless source is taken into account in [10], [11], [12], [13]. A nonbinary first-order Markov chain model is incorporated in the decoder in [14], [15] and [16]. Moreover, the iterative nature of the decoder enables the progressive refinement of the assumed source statistics [17] and marginal distributions [18].
- *Belief Propagation*: Prior information about the marginal distributions is easily incorporated in the Belief Propagation (BP) decoder. The prior information may derive from the availability at the decoder of a correlated source [19], [20], or from a model description communicated to the decoder by the encoder [21]. The incorporation of source memory in the algorithm is also possible (e.g. [22]). If the source has memory in the structure of a Markov chain, then it can be captured by a graphical model which can be adjoined to the Tanner graph of the code similarly to the case of Turbo decoding. A shortcoming of the latter approach is that the decoding complexity grows exponentially with the source Markov order.

Another scenario where the dependence among the bits fed to the channel encoder has been studied is when the encoder follows a variable-length source encoder. The fact that the outputs of the variable-length encoder are not equally likely (since some variable-length output sequences are forbidden) can be taken care at the decoder (of, typically, a convolutional or turbo channel code) by modelling the source encoder with a finite-state machine [23], [24], [25], [26], [27].

When channel-encoding redundant sources, it is natural to ask how the source statistics should drive the choice of the encoder. A necessary condition that can be directly imported from information theory is that the empirical distributions of the code outputs should look like the mutual-information maximizing inputs [28]. Thus, systematic error-correcting codes are usually poor choices in the presence of source redundancy. Most binary nonsystematic codes achieve nearly equiprobable first-order marginals that are optimal for symmetric channels. The choice of constituent encoders in Turbo codes to improve performance for biased memoryless sources has been addressed in [11], [13]. Taking into account the redundancy in the source at the decoder enables to achieve higher rates by simply puncturing the encoder outputs. In [10] both the systematic bits and a fixed fraction of the parity-check bits of a turbo encoder are discarded.

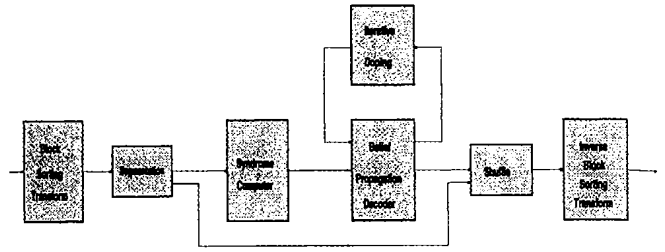


Fig. 1. Compression/Decompression Scheme for Noiseless Channel

Our approach to exploiting the memory redundancy of the source at the decoder is rather different from the previous approaches, and was presented in the context of data compression (noiseless channel) in [29], [30]. With a noiseless channel, the scheme is given in Figure 1. The block-sorting transform shown in Figure 1 is a one-to-one transformation, also called the Burrows-Wheeler transform (BWT) [31] which performs the following operation: after adding a special End-of-file symbol, it generates all cyclic shifts of the given data string and sorts them lexicographically. The last column of the resulting matrix is the BWT output from which the original data string can be recovered.

The BWT shifts redundancy in the memory to redundancy in the marginal distributions. The redundancy in the marginal distributions is then much easier to exploit at the decoder as the decoding complexity is independent of the complexity of the source model (in particular, the number of states for Markov sources). It is shown in [32] that the output of the BWT (as the blocklength grows) is asymptotically piecewise i.i.d. For stationary ergodic tree sources the length, location, and distribution of the i.i.d. segments depend on the statistics of the source. The universal BWT-based methods for data compression all hinge on the idea of compression for a memoryless source with an adaptive procedure which learns implicitly the local distribution of the piecewise i.i.d. segments, while forgetting the effect of distant symbols.

In the data compression algorithm of [29], [30], the compression is carried out by multiplication of the Burrows-Wheeler Transform of the source string with the parity-check matrix of an error correcting code. Of particular interest are LDPC codes whose Belief Propagation decoder is able to incorporate the time-varying marginals at the output of the BWT in a very natural way. The unequal marginals produced at the output of the BWT have a synergistic effect with the Belief Propagation algorithm which is able to iteratively exploit imbalances in the reliability of variable nodes. The universal implementation of the algorithm where the encoder identifies the source segmentation and describes it to the decompressor is discussed in [21].

An important ingredient in the compression scheme of [29], [30], [21] is the ability to do decompression at the compressor. This enables to tune the choice of the

codebook to the source realization and more importantly it enables the use of the Closed-Loop Iterative Doping (CLID) algorithm of [29]. This is an effective algorithm which enables zero-error data compression with performance which is very competitive with that of standard data compression algorithms.

The current availability of channel codes that achieve rates close to channel capacity while using the linear-complexity BP decoder, coupled with the availability of universal data compressors/decompressors following the same paradigm [29], [30], [21] present a synergistic opportunity to design joint source-channel encoders and decoders with favorable performance-complexity. In this paper we emphasize the new ingredients that come into play because of the presence of the noisy channel. Aspects such as universality to the source statistics and dealing with the source memory through the BWT are not emphasized since they are identical to the way we handled them in the pure data compression scenario [29], [30], [21].

The rest of the paper is organized as follows. Section 2 proposes three different linear coding schemes for joint source-channel encoding. Section 3 is devoted to the Quenched Belief Propagation (QBP) scheme which is effective in order to increase the resilience to channel errors/erasures for low-noise channels. Section 4 examines the capabilities of joint source-channel encoding/decoding when some degree of feedback is available. Because of space limitations the paper focuses on the description of the new codes rather than on numerical experimentation.

2 Joint Source-Channel Encoding

We consider three classes of encoding structures, for all of which the source is first passed through a BWT and joint BP decoding is performed prior to inverse BWT. These codes are universal in that they are not tuned to the source in any way other than they use a rate (ratio of input to output bits) which is known to be below the ratio of channel capacity to source entropy.

2.1 Two LDPCs

The first approach builds an n -to- m encoder by adjoining the Tanner graphs of two LDPC codes as shown in Figure 2.

The block source encoder outputs a block of ℓ bits using an $(\ell \times n)$ matrix \mathbf{H}_s . Then we apply the resulting ℓ -block to a $(\ell \times m)$ generator matrix \mathbf{G}_c of an LDPC with a low-density parity check matrix $((m-\ell) \times m)$ \mathbf{H}_c . To that end, we can use the (almost) linear encoding algorithm [33], which takes an arbitrary parity check matrix and generates an associated systematic generator matrix. As a byproduct of the algorithm, it identifies a linearly-independent subset of ℓ variable nodes in the original $((m-\ell) \times m)$ \mathbf{H}_c parity-check matrix. Now \mathbf{H}_c s becomes the systematic input,

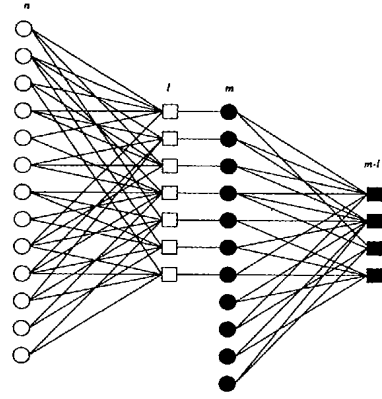


Fig. 2. LDPC for separate source-channel encoding.

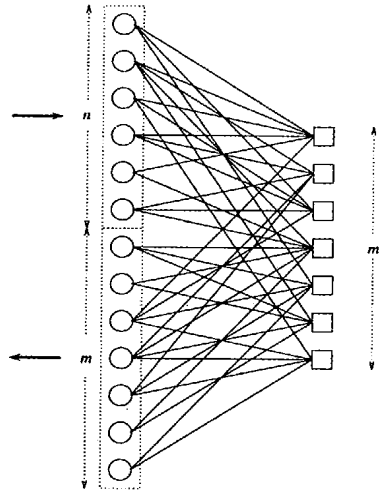


Fig. 3. LDPC for joint source-channel encoding.

and the remaining $m - \ell$ channel inputs are simply the nonsystematic symbols generated by the generator matrix of the channel code. The graph of the channel parity matrix is just a permuted version of the graph that would be obtained with the original parity check matrix \mathbf{H}_c . The m black variable nodes correspond to the channel outputs. The choice of ℓ (i.e. the source and channel coding rates) is a free design parameter. Note that with a joint belief-propagation decoder there is indeed some performance advantage in not letting ℓ/n come down to close to the entropy since the residual redundancy of the source can be very beneficial for the convergence of the iterations.

2.2 Single LDPC

The second class of encoding designs we analyze consist of a single systematic LDPC of rate $\frac{n}{n+m} \approx \frac{C}{C+H}$; we apply the BWT output to the systematic part and send through the channel the nonsystematic part of the codeword. See Figure 3 where the variable nodes have been rearranged so that the nodes $\{1, \dots, n\}$ correspond

to the systematic part. In the source coding problem the approach we introduced in [29], [30] in which we encoded by multiplying the source vector with an $m \times n$ parity-check matrix \mathbf{H} is a special case of the current approach where the parity check matrix of the rate $n/(n+m)$ code is the matrix $[\mathbf{H} \quad -\mathbf{I}]$.

2.3 Lotus Codes

In the classical parallel concatenated convolutional code (turbo-code) the source n -sequence s is fed in the natural ordering to a recursive convolutional encoder RSCC1 and in the *interleaved* order ($s\Pi$, where Π denotes the permutation matrix corresponding to the interleaver) to the recursive convolutional encoder RSCC2. Denoting by \mathbf{x}_1 and \mathbf{x}_2 the outputs of RSCC1 and RSCC2, the turbo-encoded codeword is formed by the concatenation $\mathbf{x} = [s, \mathbf{x}_1, \mathbf{x}_2]$ (we include possible puncturing of the symbols output by RSCC1 and RSCC2 as part of the convolutional encoders). The Tanner graph of the corresponding *regular* turbo-code is shown in Fig. 4. We call this structure “regular” since every information bit-node has the same degree (two, in this case). For simplicity, we assume that the encoders RSCC1 and RSCC2 are identical.

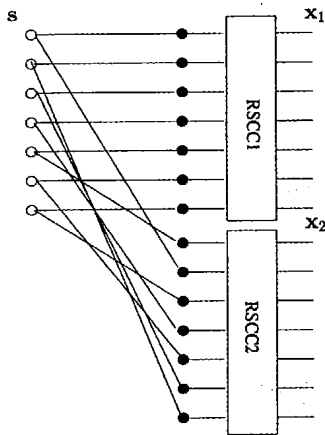


Fig. 4. Tanner graph of a classical turbo-code.

We derive the Tanner graph for the family of Lotus codes¹ by modifying the Tanner graph of Fig. 4. First, we join the two trellis sections corresponding to RSCC1 and RSCC2 by letting the initial state of the trellis of RSCC2 coincide with the final state of the trellis of RSCC1. Then, we notice that the sequence $c = [s, s\Pi]$ at the input of the common recursive convolutional encoder (from now on denoted by RSCC) is obtained via a linear encoding operation, with generator matrix $\mathbf{G} = [\mathbf{I}, \Pi]$ of rate $1/2$, where the corresponding linear code is isomorphic to the Cartesian product of n repetition

¹We christen this family of codes as LOTUS codes as a reflection that they include both LOW-density parity check and TURBO codes as special cases. In mythology, lotus is an all-encompassing symbol of creation, birth and dawn.

codes of length 2. We generalize the encoder by letting \mathbf{G} be an arbitrary generator matrix. In particular, we are interested in the class of low-density generator matrices defined by the degree sequences $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_i \rho_i x^{i-1}$, where λ_i (resp., ρ_i) denote the fractions of edges of left (resp., right) degree i in the Tanner graph representation of the code defined by \mathbf{G} . The Tanner graph of a Lotus code is shown in Fig. 5. For future use, the nodes corresponding to information bits are referred to as the “source nodes”, the nodes corresponding to the modulo-2 sums are referred to as the “check nodes” (even if strictly speaking they are not parity-check constraints) and the nodes corresponding to the inputs of the RSCC encoder are referred to as the “parity bit nodes”.

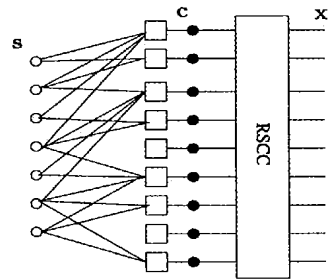


Fig. 5. Tanner graph of a Lotus code.

The ensemble of Lotus codes with input block length n , degree sequences $\lambda(x), \rho(x)$ and convolutional component RSCC is formed by all Tanner graphs of the type represented in Fig. 5 where the edge left and right connections are defined by a permutation π of $e = n / \int_0^1 \lambda(x) dx$ elements² selected with uniform probability over the set of all e -elements permutations. A Lotus ensemble with given degree sequences and convolutional component will be denoted by $\mathcal{C}(\lambda, \rho, \text{RSCC})$ in the following, where “RSCC” is a placeholder indicating the recursive convolutional code components (e.g., we can use the generators of the binary encoder in octal notation in conjunction with some ad-hoc notation indicating the puncturing pattern). Note that in the special case where the RSCC consists of two autonomous copies of the same recursive convolutional encoder, the left degree is equal to 2 and the right degree is equal to 1, we obtain the classical turbo codes [34]; since LDPC codes have low density generator matrices, the Gallager codes [35] and the irregular LDPC codes [36] are also special cases of Lotus codes taking the RSCC to be the identity; when the left and right degrees are constant and the RSCC is an accumulator the structure reduces to the Repeat-Accumulate codes of [37]; dropping the restriction that the left degree is constant (while keeping a constant right degree) leads to the Irregular Repeat-Accumulate codes of [38]; and if in that structure

²Here e denotes the number of edges, assumed to be an integer. In practice, for finite n some integer rounding is needed.

we fix a right degree of one and allow a general recursive systematic convolutional code (instead of an accumulator) we obtain the irregular Turbo codes of [39]. Even though here we have limited the description of the class of Lotus codes to a front-end ensemble which is characterized like a standard irregular LDPC ensemble with a left and right degree distribution, we note that as far as complexity-performance is concerned it is advantageous to consider yet a more general structure encompassing the so-called *multi-edge type* LDPCs.³ In that case edges are colored and a variable-node vector distribution dictates the number of edges of each color emanating from each variable node, and likewise for the check nodes. This generalization is particularly useful in order to incorporate knowledge at the encoder of time-varying source distributions such as we would have in a non-universal setting at the output of the BWT. It also can encompass the use of precoding which is beneficial to bootstrap good block error rate performance from good bit error rate performance (see Figure 8.)

2.3.1 Design of Lotus codes based on EXIT functions

In our joint source-channel coding scheme we apply the source sequence s to the Lotus encoder, obtaining the encoded sequence x (see Fig. 5) transmitted over the channel. At the receiver we apply standard BP decoding based on the channel output y corresponding to the input x , taking into account the a priori marginal probabilities on the source bitnodes s .

It can be easily shown that this BP algorithm corresponds to the standard BP algorithm for the case where the source symbols have uniform prior probability, but the additional observation (side "systematic channel") $v = s + z$ is available at the receiver, where the BSC noise z has the same marginals of the original source sequence. Hence, the design of Lotus codes for joint source-channel coding reduces to the design of Lotus codes for standard channel coding where the systematic bits (corresponding to the source nodes) are sent via a (possibly time-varying) BSC with independent noise with the same marginal probabilities of the source, and the convolutionally encoded parity bits (output by the RSCC component) are sent through the actual transmission channel.

In order to optimize the Lotus ensemble $\mathcal{C}(\lambda, \rho, \text{RSCC})$ for given source and channel statistics, we make use of the by-now standard approach of EXIT functions and Gaussian approximations [40], [41]. This method aims at optimizing the average mutual information flow over the graph representing the code, under the assumptions that the graph is cycle free and that the messages exchanged by the graph nodes follow a *symmetric* Gaussian distribution (see [40]). The symmetry condition for Gaussian distributions imposes that the variance is equal to twice the mean.

³Due to T. Richardson and R. Urbanke, work in progress.

Let X be a binary random variable, Y some observation about it, and $\mathcal{L} = \log P(X = 0|Y)/P(X = 1|Y)$ denote its posterior log-likelihood ratio. The mutual information $I(X; \mathcal{L})$ assuming that \mathcal{L} is conditionally distributed as $\mathcal{N}(\mu, 2\mu)$ given $X = 0$ is given by the function

$$J(\mu) = 1 - \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-z^2} \log_2 \left(1 + e^{-2\sqrt{\mu}z - \mu} \right) dz, \quad (1)$$

For any desired RSCC component and any memoryless stationary transmission channel (e.g., for the BSC), we can compute by Monte Carlo simulation the *Extrinsic Information Transfer* (EXIT) function of RSCC assuming that the messages from the parity nodes to the trellis are Gaussian symmetric, with mean value μ satisfying $\mu = J^{-1}(v)$. By letting v varying from 0 to 1, we obtain the EXIT function $u = f(v, C)$, where u denotes the mutual information between the parity nodes and the corresponding leftbound messages received from the trellis and C denotes the capacity of the transmission channel.

We let x denote the mutual information between the source symbols and the corresponding rightbound messages (sent from the source nodes to the check nodes) and y denote the mutual information between the source symbols and the corresponding leftbound messages (sent from the check nodes to the source nodes). Moreover, for a given degree sequence $a(x) = \sum_i a_i x^{i-1}$ we define the function

$$F_a(x, y) = \sum_i a_i J \left((i-1)J^{-1}(x) + J^{-1}(y) \right) \quad (2)$$

From standard arguments [40], under the cycle-free and Gaussian approximation assumptions we have the following (approximate) relations

$$\begin{aligned} x &= F_\lambda(y, 1 - H(S)) \\ y &= 1 - F_\rho(1 - x, 1 - u) \\ v &= 1 - F_{f\rho}(1 - x, 0) \\ u &= f(v, C) \end{aligned}$$

where $H(S)$ denotes the entropy of the source (assuming it is independent possibly non-stationary), and where $f\rho$ is a short-hand notation for the degree sequence $\int_0^x \rho(z) dz / \int_0^1 \rho(z) dz$. By eliminating y, u and v , we obtain the fixed-point equation for x

$$x = F_\lambda \left(1 - F_\rho \left(1 - x, 1 - f \left(1 - F_{f\rho}(1 - x, 0), C \right) \right), 1 - H(S) \right)$$

It is easily seen that the right side of (3), for each given value of $x \in [0, 1]$, is a linear function of the coefficients $\{\lambda_i\}$. Hence, we can set-up a joint source-channel optimization procedure such that, for each desired value of C and $H(S)$, each desired guess for the RSCC component, and each desired guess for the right degree sequence $\rho(x)$, we find the optimal $\lambda(x)$ such that the

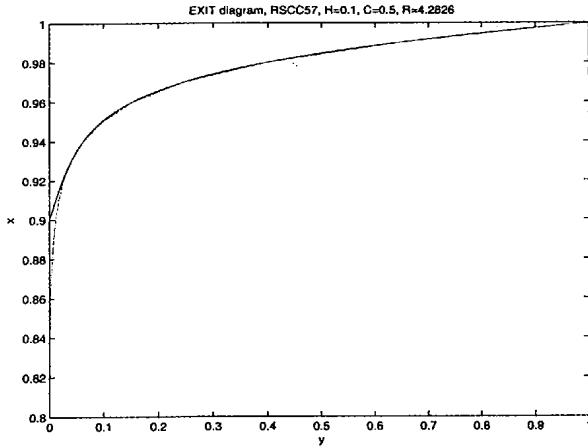


Fig. 6. EXIT diagram for an optimized Lotus ensemble with RSCC57, Bernoulli source with $H(S) = 0.1$ and BSC with $C = 0.5$.

source-channel coding rate

$$R = \frac{n}{m} = R_{\text{rsc}} \frac{\int_0^1 \lambda(z) dz}{\int_0^1 \rho(z) dz} \quad (4)$$

is maximum subject to the constraint that (3) has the unique fixed point $x = 1$. Since both the constraint and the objective function are linear in $\{\lambda_i\}$, we can solve numerically this optimization problem by standard linear programming by sampling the interval $x \in [0, 1]$ on a fine grid. For each value of x we obtain a linear constraint. Moreover, it is easy to show that the only the values of x in the open interval $(1 - H(S), 1)$ generate “biting” constraints.

The approximated mutual information evolution for an optimized Lotus ensemble can be visualized on the EXIT diagram. For a set of optimized coefficients, we plot the functions $x = F_\lambda(y, 1 - H(S))$ and $y = 1 - F_\rho(1 - x, 1 - f(1 - F_f \rho(1 - x, 0), C))$. Intersections between these curves corresponds to fixed point of (3). Fig. 6 shows the EXIT diagram for $C = 0.5$ and $H(S) = 0.1$, a Bernoulli i.i.d. source and a BSC transmission channel, for the convolutional component with generators (5, 7) (octal notation), denoted by RSCC57, with rate 1 and encoder transfer function $G(D) = (1 + D^2)/(1 + D + D^2)$, of rate $R_{\text{rsc}} = 1$. Despite the fact that the room left between the two curves is very small, the achieved source-channel coding rate is only $R = 4.2826$ source symbols per channel use, while the asymptotically optimal Shannon limit is $C/H(S) = 5$.

3 QBP decoding

In the case of low-noise channels where errors or erasures occur with small probability, we propose an effective modification to the pure data compression scheme which we refer to as *Quenched Belief Propagation* (QBP). The rationale of this scheme is that

for those high capacity channels it is only necessary to slightly increase the compression rate in order to obtain good bit error rate with the LDPC-based compression scheme on [29] using the CLID algorithm. Since the encoder no longer has an exact copy of the decoder input because of channel noise it is necessary to modify the scheme in order to obtain good block error rate. We explain the QBP scheme in the particular case of the binary erasure channel:

- 1) Iterate the BP by deleting the parity-check equations corresponding to erased syndrome symbols.
- 2) Quench erased syndrome bits after q iterations, making a hard decision treated henceforth as the true (recovered) syndrome bits.
- 3) Apply d iterations of BP using all the parity-check equations (including the value of the recovered erased syndrome bits) and using the doping bits. If the doping bits contain more than a fixed number g of erasures, then the current hard decisions made by the standard BP algorithm are declared. Otherwise all possible combinations of the erased doped bits are tried and the algorithm stops as soon as all the parity-check equations are verified in exactly d iterations.

Figure 7 shows the result of an experiment with a biased coin source transmitted through a binary erasure channel with erasure rate $e = 10^{-3}$. A library of 8 codes drawn from an irregular LDPC ensemble of rate 1/2 is chosen. In Figure 7, the blocklength is equal to $n = 3000$, and $d = 200$ doping bits are sent. The number of iterations prior to quenching is equal to $q = 200$, and the maximum number of allowed erased doping bits is equal to 4. We see the very different resilience against channel errors of this scheme compared to arithmetic coding as a function of the source bias p , represented in the x -axis as the “rate redundancy”

$$1 - \frac{2h(p)}{1 - e}$$

4 Feedback Schemes

Even in purely channel coding, very low block error rates require either very long codewords or channel feedback. Since (at least for well-behaved stationary sources and channels) the rationale for joint source/channel coding disappears in the regime of asymptotically long blocklengths and since in many of the practical applications where blocklengths are moderate, feedback is available by using a portion of the payload of reverse channels, it is of considerable interest to develop joint source/channel encoding/decoding schemes which can use feedback information effectively. In fact, third-generation wireless systems achieve good block error rate with moderate blocklengths thanks to feedback in the form of ARQ.

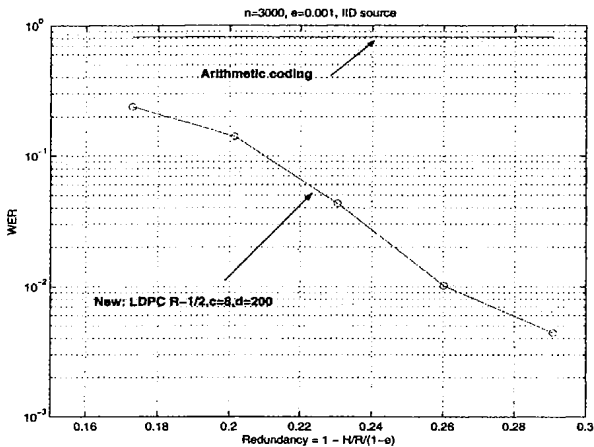


Fig. 7. Arithmetic coding vs QBP decoded LDPC scheme in the presence of erasures.

Constructive schemes for joint source/channel coding with feedback have received some attention in the literature. Pragmatic schemes for lossy coding with feedback have been proposed in [42], [43]. In almost-lossless coding with full noiseless causal feedback, [44] elaborates on a channel coding scheme dating back to [45], to obtain a variable-length scheme that attains the optimum exponential asymptotic decrease in error probability. Although the fundamental Shannon limit does not improve with feedback if the channel is memoryless, the error exponent does improve rather dramatically. However, full noiseless feedback as introduced by Shannon in [46] is too demanding of reverse channel bandwidth to be of relevance to most practical problems.

Here we propose several feedback schemes tailored to a BP decoder. These schemes require far less feedback than in the Shannon paradigm and prove to be quite effective in lowering the block error rate when used in conjunction with our joint source/channel schemes. In general, for a given desired block error rate there is a tradeoff between the number of feedback symbols and the required feedforward blocklength.

A key insight is to request additional information about the source from the encoder as a function of the state of the BP (reliabilities of the source symbols).

Consider the following schemes⁴:

- 1) At certain preselected iterations, the BP identifies the group of source symbols with the lowest reliability. The cardinality of this group can be preagreed or may depend on how many symbols have reliability below a certain threshold. (Note that if symbols are requested that have already appreciable reliability, there is nonnegligible loss of optimality.) Communicating the location of a

⁴In each case we assume that the feedback information and subsequent feedforward transmission are protected against channel noise by sufficient redundancy.

group of L symbols among the n source symbols takes $\lceil \log_2 \binom{n}{L} \rceil$ bits. Upon receipt of that information, the encoder sends the uncompressed L source symbols whose locations have been specified by the decoder. Note that in the extreme case of $L = 1$ this is equivalent to the CLID algorithm for pure data compression [29].

- 2) The n source bits are evenly partitioned into s subsets in a manner preagreed by encoder and decoder. After a certain number of iterations the BP selects the subsets of the lowest reliability symbols increasing in reliability until t subsets have been selected. Note that it is possible that some of the selected subsets will contain more than one low-reliability symbol. Using $\lceil \log_2 \binom{s}{t} \rceil$ feedback bits, it then identifies the group of t subsets containing one or more unreliable bits. Unlike the previous case in which the symbols are directly communicated to the decompressor, here the encoder communicates t symbols each of which is the result of the addition (parity-check) of the symbols in its corresponding subset. The decompressor then uses the new parity check equations in further iterations of the BP. If at later stages the same procedure is repeated, a different grouping of the source symbols is used every time.
- 3) Suppose that we construct a *finite geometry* where "points" take the role of source bits and "lines" take the role of codes [47], so that any bit participates in M codes and any two codes have at most one bit in common. Then, for a small number of residual errors it is likely that we find some codes containing only one bit in error, by requesting the parity bits of these codes, we can correct residual errors. The BP selects the codes in which one and only one bit has low reliability.
- 4) Low block error probability can be bootstrapped from low bit error probability by adding a relatively small amount of redundancy in the form of the parity checks generated by a *expander code* [48]. In the conventional approach without feedback, this can be viewed as a form of precoding which is then decoded jointly with the LDPC by BP iterations on a three-layered Tanner graph. Feedback opens up the possibility that instead of selecting an expander code beforehand, the code is chosen, from among a given preselected library of 2^c expander codebooks, as a function of the source and channel realization. To that end, for any given snapshot of symbol reliabilities, we can compute a score for each codebook and communicate to the encoder using c feedback bits the identity of 'most valuable' codebook. In principle, a desirable codebook is the one that maximizes the mutual information between its parity checks and the input symbols (whose

distributions are fixed at the values determined by their reliabilities). Since this involves too much computation a pragmatic way to compute the score of a codebook for a given snapshot of symbol reliabilities is to compute the probabilities of the parity-check equations (using the BP equations) and then their corresponding entropies. Approximating the parity-check values to be independent (in keeping with the BP philosophy), the score is then the sum of the individual parity-check entropies. Once the parity-checks of the selected codebook are received the BP incorporates the corresponding new equations into the graph and proceeds with its iterations starting with the last snapshot of reliabilities.

In Figure 8 we show the effect on both source-symbol-error rate (lower set of curves) and block-error rate (upper set of curves) of precoding and feedback for blocklength equal to 10,000 source symbols. The baseline Lotus code has rate 4.28 source symbols per channel use and uses the convolutional code RSCC57 analyzed in Fig. 6. No attempt is made to optimize the code graph. The abscissa is the normalized gap from the separation limit, defined as $1 - R/(C/H(S))$. Channel capacity is $C = 0.51$ bit and the entropy of the source ranges from 0.06 to 0.09 bits per source symbol.

The curves labelled “precoded ensemble average” refer to the insertion of a systematic precoder drawn randomly from a (3, 117) regular LDPC ensemble, with 256 checknodes and rate $R_{\text{prec}} = 1 - 3/117 = 0.9744$. At the BP decoder, the parity checks produced by the code are assumed to be equiprobable.

The feedback scheme tested in 8 is a hybrid of the general schemes presented above. It uses again a (different) (3, 117) regular LDPC graph, with 256 checknodes. Instead of transmitting the checknodes, they are randomly partitioned into 64 subsets of 4 checknodes each in a manner preagreed by encoder and decoder. After a fixed number of iterations the BP algorithm sends a feedback message of 6q bits requesting the checksums corresponding to the subcodes that cover the q least reliable source nodes. After several more iterations the process can be repeated by requesting the checknodes of subcodes that have not been requested before. The curves labelled “feedback” in Figure 6 consider the case $q = 1$ and are parametrized with the number of feedback rounds (from 0 to 40), duly incorporating the feedforward traffic in the computation of the overall encoding rate.

References

[1] C. E. Shannon, “A mathematical theory of communication,” *Bell Sys. Tech. J.*, vol. 27, pp. 379–423, 623–656, Jul.-Oct. 1948.
 [2] S. Vembu, S. Verdú, and Y. Steinberg, “The source-channel separation theorem revisited,” *IEEE Trans. Inform. Theory*, vol. 41, pp. 44–54, Jan. 1995.

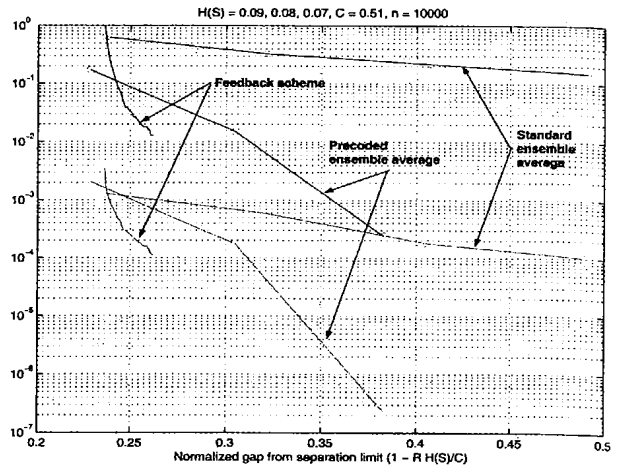


Fig. 8. Effect of precoding and feedback.

- [3] J. A. Storer and J. H. Reif, “Error resilient optimal data compression,” *SIAM J. Computing*, vol. 26, no. 4, pp. 934–949, Aug. 1997.
 [4] B. D. Pettijohn, M. W. Hoffman, and K. Sayood, “Joint source/channel coding using arithmetic codes,” *IEEE Trans. Communications*, pp. 826–836, May 2001.
 [5] S. Lonardi and W. Szpankowski, “Joint source-channel lz77 coding,” *2003 Data Compression Conference*, pp. 273–282, Snowbird, 2003.
 [6] I. Csiszar and J. Korner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*, Academic, New York, 1981.
 [7] J. L. Massey, “Joint source and channel coding,” in *Communications Systems and Random Process Theory*, vol. 11, pp. 279–293. Sijthoff and Nordhoff, 1978.
 [8] G.D. Forney, “The Viterbi algorithm,” *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, March 1973.
 [9] J. Hagenauer, “Source-controlled channel decoding,” *IEEE Trans. Communications*, vol. 43, pp. 2449–2457, Sep. 1995.
 [10] J. Garcia-Frias and Y. Zhao, “Compression of binary memoryless sources using punctured Turbo codes,” *IEEE Communication Letters*, vol. 6, pp. 394–396, Sep. 2002.
 [11] G-C Zhu and F. Alajaji, “Turbo codes for nonuniform memoryless sources over noisy channels,” *IEEE Communications Letters*, vol. 6, pp. 64–66, Feb. 2002.
 [12] A. Aaron and B. Girod, “Compression with side information using Turbo codes,” *Proc. 2002 Data Compression Conference*, 2002.
 [13] G-C. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, “Non-systematic turbo codes for non-uniform I.I.D sources over AWGN channels,” *Proc. 2002 Conf. Information Sciences and Systems*, Mar. 2002.
 [14] J. Garcia-Frias and J. D. Villasenor, “Combining hidden Markov source models and parallel concatenated codes,” *IEEE Communication Letters*, vol. 1, pp. 111–113, July 1997.
 [15] A. Ruscitto and E. Biglieri, “Joint source and channel coding using Turbo codes over rings,” *IEEE Trans. Communications*, vol. 46, pp. 981–984, Aug. 1998.
 [16] A. Guyader, E. Fabre, C. Guillemot, and M. Robert, “Joint source-channel turbo decoding of entropy coded sources,” *IEEE J. Selected Areas in Communications*, vol. 19, pp. 1680–1696, Sep. 2001.
 [17] J. Garcia-Frias and J. D. Villasenor, “Turbo decoding of hidden Markov sources with unknown parameters,” in *Data Compression Conference*, 1998, pp. 159–168.
 [18] N. Goertz, “On the iterative approximation of optimal joint source-channel decoding,” *IEEE Journal on Sel. Areas in Communication*, vol. 14, no. 9, pp. 1662–1670, Sept. 2001.
 [19] J. Garcia-Frias and W. Zhong, “LDPC codes for compression of multi-terminal sources with hidden markov correlation,” *IEEE Communications Letters*, vol. 7, No. 3, pp. 115–117, Mar. 2003.
 [20] J. Garcia-Frias, W. Zhong, and Y. Zhao, “Turbo-like codes for

- source and joint source-channel coding," *Third International Symposium On Turbo Codes and Related Topics*, pp. 43–50, Brest, France, September 1–5, 2003.
- [21] G. Caire, S. Shamai, and S. Verdú, "Universal data compression with ldpc codes," *Third International Symposium On Turbo Codes and Related Topics*, pp. 55–58, Brest, France, September 1–5, 2003.
- [22] I. Kanter, H. Kfir, and S. Keren, "An efficient joint source-channel coding for a d -dimensional array," preprint, 2003.
- [23] K. Sayood, H. H. Otu, and N. Demir, "Joint source/channel coding for variable length codes," *IEEE Trans. Communications*, pp. 787–794, May 2000.
- [24] R. Bauer and J. Hagenauer, "Iterative source/channel-decoding using reversible variable length codes," in *2000 Data Compression Conference*, 2000, pp. 93–102.
- [25] V. Balakirsky, "Joint source-channel coding with variable length codes," *Problems of Information Transmission*, vol. 36, no. 4, pp. 1–15, 2000.
- [26] J. Hagenauer and R. Bauer, "The turbo principle in joint source channel decoding of variable length codes," *2001 IEEE Information Theory Workshop*, Cairns, Australia, Sep. 2–7, 2001.
- [27] R. Bauer and J. Hagenauer, "On variable length codes for iterative source/channel-decoding," *Proc. 2001 Data Compression Conference*, 2001.
- [28] S. Shamai (Shitz) and S. Verdú, "The empirical distribution of good codes," *IEEE Trans. Inform. Theory*, vol. 43, pp. 836–846, May 1997.
- [29] G. Caire, S. Shamai, and S. Verdú, "A new data compression algorithm for sources with memory based on error correcting codes," *2003 IEEE Workshop on Information Theory*, pp. 291–295, Mar. 30–Apr. 4, 2003.
- [30] G. Caire, S. Shamai, and S. Verdú, "Lossless data compression with error correction codes," *2003 IEEE Int. Symp. on Information Theory*, p. 22, June 29–July 4, 2003.
- [31] M. Burrows and D. J. Wheeler, "A block-sorting lossless data compression algorithm," Tech. Rep. SRC 124, May 1994.
- [32] M. Effros, K. Visweswariah, S. Kulkarni, and S. Verdú, "Data compression based on the Burrows-Wheeler transform: Analysis and optimality," *IEEE Trans. on Information Theory*, vol. 48, pp. 1061–1081, May 2002.
- [33] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Information Theory*, vol. 47, pp. 638–656, Feb. 2001.
- [34] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: Turbo-codes," *Proc. International Conference on Communications*, 1993.
- [35] R. G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, 1963.
- [36] M.G. Luby, M. Mitzenmacher, A. Shokrollahi, and D.A. Spielman, "Improved low-density parity-check codes using irregular graphs," *IEEE Trans. Inform. Theory*, vol. 47, pp. 585–598, Feb. 2001.
- [37] D. Divsalar, H. Jin, and R. McEliece, "Coding theorems for 'turbo-like' codes," Sept 1998.
- [38] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat-accumulate codes," *Proc. 2nd International Symposium on Turbo codes and Related Topics*, pp. 1–8, Sept. 4, 2000.
- [39] H. E. Sawaya and J. J. Boutros, "Irregular turbo codes with symbol-based iterative decoding," *Int. Symp on Turbo Codes*, 2001.
- [40] A. Roumy, S. Guemghar, G. Caire, and S. Verdú, "Design methods for irregular repeat accumulate codes," *IEEE Trans. Information Theory*, to appear, 2004.
- [41] S. ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Trans. Signal Proc.*, vol. 51, no. 11, Nov. 2003.
- [42] Z. S. Peng, Y. F. Huang, and D. J. Costello, "Turbo codes for image transmission - a joint channel and source decoding approach," *IEEE J. Selected Areas Communications*, vol. 18 (6), pp. 868–879, June 2000.
- [43] B. Girod and N. Farber, "Feedback-based error control for mobile video transmission," *Proc. IEEE*, vol. 87 (10), pp. 1707–1723, Oct. 1999.
- [44] J.M. Ooi and G. W. Wornell, "Fast iterative coding techniques for feedback channels," *IEEE Trans. Information Theory*, vol. 44 no. 7, pp. 2960–2976, Nov. 1998.
- [45] R. Ahlswede, "A constructive proof of the coding theorem for discrete memoryless channels with feedback," *Proc. 6th Prague Conf. Information Theory, Statistical Decision Functions, and Random Processes*, p. 3950, 1971.
- [46] C. E. Shannon, "The zero error capacity of a noisy channel," *IRE Trans. Information Theory*, pp. 112–124, Sept. 1956.
- [47] E. F. Assmus, "The coding theory of finite geometries and designs," in *Lecture Notes in Computer Science, Vol. 357*, T. Mora, Ed., pp. 1–6. Springer-Verlag, 1989.
- [48] M. Sipser and D.A. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1710–1722, Nov. 1996.
- [49] D. M. Mandelbaum, "An adaptive-feedback coding scheme using incremental redundancy," *IEEE Trans. Inform. Theory*, vol. 20, pp. 388–389, May 1974.
- [50] C. F. Leanderson, G. Caire, and O. Edfors, "On the performance of incremental redundancy schemes with turbo codes," *Proc. Radiotekenskap och Kommunikation 2002*, pp. 57–61, Jun 2002.