Fast Iterative Coding Techniques for Feedback Channels

James M. Ooi and Gregory W. Wornell, *Member, IEEE*

***Abstract*—A class of capacity-achieving, low-complexity, high-reliability, variable-rate coding schemes is developed for communication over discrete memoryless channels with noiseless feedback. Algorithms for encoding and decoding that require computations growing linearly with the number of channel inputs used are developed. The error exponent associated with the scheme is shown to be optimal and implies that capacity is achievable. Simulations are performed and support the analytically predicted high performance and low complexity.**

***Index Terms*—Error-correction coding, feedback channels, iterative coding.**

I. Introduction

The availability of feedback in a communication system—i.e., a channel from receiver to transmitter through which the receiver passes the transmitter its observations—generally enables schemes for communicating over the forward channel to have lower computational complexity, higher reliability, higher capacity, or a combination of these advantages, in comparison to feedback-free communication schemes. The research of Schalkwijk and Kailath [1], Horstein [2], Berlekamp [3], Yamamoto and Itoh [4], Burnashev [5], Kudryashov [6], Gaarder and Wolf [7], Cover and Leung [8], Veugen [9], and many others attests to these advantages.

In this paper, we develop a framework through which high-reliability, low-complexity coding schemes for a broad class of channels with feedback can be designed. We then focus on using this framework to develop for discrete memoryless channels with feedback (DMC$_\mathrm{f}$'s) a coding scheme with the highest possible error exponent (optimum reliability); we study the complexity of the resulting scheme in some detail and show it to be minimal in a natural asymptotic sense.

The central notion underlying the framework, which we term the compressed-error-cancellation framework, is best conveyed via the following example of a coding scheme for communicating over a DMC$_\mathrm{f}$ with channel transition function $q_{Y|X}$ and capacity-achieving input distribution q_X. For notational convenience, let X and Y be random variables such that

$$p_{X,Y}(x, y) = q_X(x) q_{Y|X}(y|x).$$

Then consider a coding scheme in which the transmitter, receiver, and channel act as follows[1] (also see Fig. 1 for a graphical representation):

Tx.1: a) Precodes N message bits into $N_1 = N/H(X)$ channel inputs[2] $X_1^{N_1}$ that look independent and identically distributed (i.i.d.) according to q_X.

b) Sends $X_1^{N_1}$ over channel.

Ch.1: Corrupts $X_1^{N_1}$ according to $q_{Y|X}$.

Rx.1: Feeds corrupted data $Y_1^{N_1}$ back to Tx.

Tx.2: a) Using $Y_1^{N_1}$, compresses $X_1^{N_1}$ into $N_1 H(X|Y)$ new data bits.[3]

b) Precodes new data bits into

$$N_2 = N_1 H(X|Y)/H(X)$$

channel inputs $X_{N_1+1}^{N_1+N_2}$, which look i.i.d. according to q_X.

c) Sends $X_{N_1+1}^{N_1+N_2}$ over channel.

Ch.1: Corrupts $X_{N_1+1}^{N_1+N_2}$ according to $q_{Y|X}$.

Rx.2: Feeds corrupted data $Y_{N_1+1}^{N_1+N_2}$ back to Tx.

Tx.3: a) Using $Y_{N_1+1}^{N_1+N_2}$, compresses $X_{N_1+1}^{N_1+N_2}$ into $N_2 H(X|Y)$ new data bits.

b) Precodes new data bits into

$$N_3 = N_2 H(X|Y)/H(X)$$

channel inputs $X_{N_1+N_2+1}^{N_1+N_2+N_3}$, which look i.i.d. according to q_X.

c) Sends $X_{N_1+N_2+1}^{N_1+N_2+N_3}$ over channel.

$\vdots$

If we assume both precoding and source coding to be invertible, then data transmitted at Tx.$(i+1)$ along with data received at Rx.i are sufficient to determine the data transmitted at Tx.i. Therefore, a receiver can recover the original message if it can recover any one of the messages transmitted on any iteration.

Manuscript received October 20, 1996; revised February 14, 1998. This work was supported in part by the Office of Naval Research under Grant N00014-96-1-0930, the Army Research Laboratory under Cooperataive Agreement DAAL01-96-2-0002, and an AT&T Doctoral Fellowship.

The authors are with the Department of Electrical Engineering and Computer Science, and the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Publisher Item Identifier S 0018-9448(98)06749-2.

[1] Note that this description is a high-level one and omits a number of details that will be given shortly.

[2] As a notational convenience to consolidate lists of variables in this paper, we adopt the shorthand a_m^n for $(a_m, a_{m+1}, \cdots, a_n)$, and, in turn, the shorthand a^n for a_1^n. As related notation, we use $\mathcal{A}^n$ to denote the n-fold Cartesian product of a set $\mathcal{A}$ with itself, where n may be infinite. This notation holds only for sub- and superscripted variables that have not otherwise been specifically defined.

[3] Note that in the system we develop later in this paper, the length of the compressed data is actually a random variable with mean close to $N_1 H(X|Y)$.

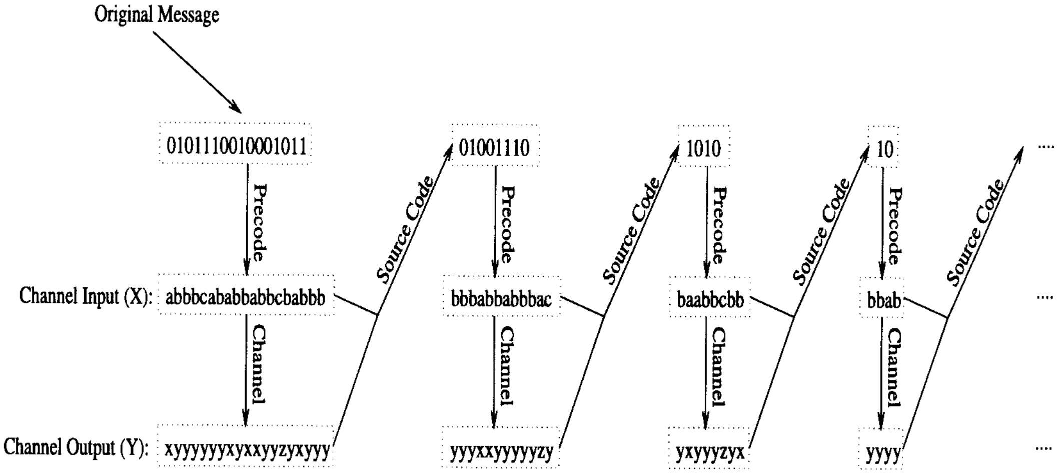

Fig. 1. Graphical representation of the iterative coding scheme for a DMC$_\mathrm{f}$ with input alphabet $\{a, b, c\}$ and output alphabet $\{x, y, z\}$; the capacity-achieving distribution is such that $H(X) < 1$.

When the process iterates indefinitely, the number of channel inputs used is

$$\frac{N}{H(X)} + \frac{N}{H(X)}\left(\frac{H(X|Y)}{H(X)}\right) + \frac{N}{H(X)}\left(\frac{H(X|Y)}{H(X)}\right)^2 + \cdots = \frac{N}{H(X) - H(X|Y)}$$

giving a rate equal to the channel capacity. Again, in practice, we terminate the process after a finite number of iterations by sending a final block of data over the channel using a termination coder.

We should note that an idea of this type was used by Ahlswede in his proof of the coding theorem for DMC$_\mathrm{f}$'s [10], though his investigation was rather limited in scope. In the remainder of this paper, we precisely develop how this idea can be applied to construct low-complexity, high-rate, optimum-reliability coding schemes for DMC$_\mathrm{f}$'s. And although beyond the scope of the present paper, we emphasize at the outset that the same framework can also be used to obtain low-complexity, high-rate, high-reliability strategies for discrete finite-state channels, unknown channels, and multiple-access channels, all with complete, noiseless feedback, as well as channels with noisy and partial feedback [11].

We begin the detailed development of a coding scheme for DMC$_\mathrm{f}$'s with a formulation of variable-length coding in Section II. We follow with a description and analysis of the coding scheme in Sections III–VI and discuss some remaining practical and theoretical issues in Section VII.

II. Variable-Length Codes and Their Properties: Reliability and Complexity

A variable-length *code* is a 4-tuple $(N, \varepsilon, \{\chi_i\}_i, \Delta)$, where N is the number of message bits to be transmitted,

$$\varepsilon : \{0, 1\}^N \times \mathcal{Y}^\infty \to \mathcal{X}^\infty$$

is the encoding function,

$$\{\chi_i : \mathcal{Y}^i \to \{0, 1\}\}_{i=1}^\infty$$

is a sequence of stopping functions, and[4] Δ: $\mathcal{Y}^\dagger \to \{0, 1\}^N$ is the decoding function.

The first argument to the encoding function ε represents the message to be encoded, while the second argument represents the stream of feedback coming back to the transmitter. Because of the causal nature of the feedback, ε is restricted to have the form

$$\varepsilon(w^N, y^\infty) = (\tilde{\varepsilon}_1(w^N), \tilde{\varepsilon}_2(w^N, y^1), \tilde{\varepsilon}_3(w^N, y^2), \cdots) \quad (1)$$

where the range of $\tilde{\varepsilon}_i$ is $\mathcal{X}$ for all i.

In describing the performance characteristics of such a code on a given DMC$_\mathrm{f}$ $(\mathcal{X}, q_{Y|X}, \mathcal{Y})$, we let the random variables $W_1, \cdots, W_N$ represent the N i.i.d. equally likely message bits, i.e., the message W^N is a discrete random variable that is uniformly distributed over the set

$$\mathcal{W} = \{0, 1\}^N. \quad (2)$$

We then let the process $\{Y_k\}$ be the output process resulting from passing W^N through the DMC$_\mathrm{f}$ $q_{Y|X}$ via $\{\{\tilde{\varepsilon}_i(m, \cdot)\}_{i=1}^\infty\}_{m \in \{0,1\}^N}$—i.e.,

$$p_{Y^n|W^N}(y^n|w^n) = \prod_{k=1}^{n} q_{Y|X}(y_k|\tilde{\varepsilon}_k(w^n, y^{k-1})) \quad (3)$$

for all $y^n \in \mathcal{Y}^n$ and all $w^n \in \{0, 1\}^n$.

To define the rate of the code, we must define its transmission length, which is the point at which the receiver stops data acquisition. The stopping functions χ_i are the mechanism by which the receiver determines from its observations when to stop data acquisition. By simulating the receiver using the feedback link, the transmitter can also determine when to correspondingly terminate transmission. In particular, at each time k, the function χ_k maps the data observed up to that time, Y^k, into a decision as to whether to terminate transmission; the value one is the signal to terminate, while the value zero is the signal to continue, i.e., $\chi_i : \mathcal{Y}^i \to \{0, 1\}$. In terms of

[4] In the description of variable-length tuples, it is convenient to use $\mathcal{A}^\dagger$ to denote the set of all tuples whose elements are in the set $\mathcal{A}$, i.e., $\mathcal{A}^\dagger = \cup_{n=1}^\infty \mathcal{A}^n$.

this notation, the transmission length is given by the random variable

$$L^* = \min_k \{k: \chi_k(Y^k) = 1\}. \tag{4}$$

The rate of the code $(N, \varepsilon, \{\chi_i\}_{i=0}^{\infty}, \Delta)$ is then defined to be $N/E[L^*]$.

The decoder makes an estimate $\hat{W}^N$ of the message sent using its decoding function $\Delta: \mathcal{Y}^{\dagger} \rightarrow \mathcal{W}$, i.e.,

$$\hat{W}^N = \Delta(Y^{L^*}) \tag{5}$$

and the associated error probability of the code is $\Pr\{\hat{W}^N \neq W^N\}$.

A *coding scheme* is a function mapping two parameters p_1 and p_2 to a variable-length code. We say a coding scheme c attains an error-exponent function $F: I \rightarrow \mathbb{R}$ if for each rate $r \in I$, where $I \subset \mathbb{R}$, there is a sequence of parameter pairs $\{(p_{1,n}(r), p_{2,n}(r))\}_n$ such that the resulting sequence of codes $\{c(p_{1,n}(r), p_{2,n}(r))\}_n$ is such that the corresponding rate sequence $\{R_n\}_n$ and the probability of error sequence $\{P_n\}_n$ obey

$$\lim_{n \rightarrow \infty} R_n \geq r \tag{6}$$

$$\lim_{n \rightarrow \infty} -\frac{R_n \log P_n}{N_n} \geq F(r) \tag{7}$$

where N_n is the number of message bits for the code $c(p_{1,n}(r), p_{2,n}(r))$.

While reliability is an important property of codes, it says little about the computational resources—i.e., the time and space complexities—required by encoding and decoding algorithms. Making our notions of time and space complexity more precise, we define time complexity and space complexity to be the asymptotic order of growth of the time cost and space cost used to run an algorithm on a random-access machine under the *uniform-cost* criterion. See [12] for more details. It is important to keep in mind, however, that there are many alternative notions of computational complexity, a number of which are outlined in [13]. Since no single notion seems to have become dominant in the channel coding literature, we choose the above measure as for its tractability and meaningfulness.

III. A Feedback Coding Scheme for DMC$_\text{f}$'s

Having defined the elements that constitute a variable-length feedback code, we can describe a coding scheme c_{DMC} for a given a DMC$_\text{f}$ $(\mathcal{X}, q_{Y|X}, \mathcal{Y})$ based on the compressed-error-cancellation framework. As stated in the previous section, a coding scheme is a function taking two parameters as input and returning a code. Our scheme c_{DMC} takes as its two parameters a message length N and a termination coder parameter ν, which together determine the rate and probability of error for the code, as we show later. Then $c_{\text{DMC}}(N, \nu)$ is a variable-length feedback code $(N, \varepsilon, \{\chi_i\}_{i=1}^{\infty}, \Delta)$, which we now describe, with primary emphasis on the encoder ε—the corresponding definitions of the decoder and stopping functions are implied by definition of the encoder. Note that for notational cleanliness, we do not make explicit the dependence of the last three elements of the 4-tuple on ν, N, or the DMC$_\text{f}$ parameters. Also for notational purposes, we let q_X be the capacity-achieving input distribution for the channel, and let X and Y be random variables that are jointly distributed according to $p_{X,Y}(x, y) = q_X(x)q_{Y|X}(y|x)$, for all $x \in \mathcal{X}$ and all $y \in \mathcal{Y}$.

A. Encoding

To define ε, we define the sequence of channel inputs $\varepsilon(W^N, Y^{\infty})$ that is transmitted when the message to be sent to the receiver is W^N and when the sequence Y^{∞} is fed back.

To describe this sequence of transmitted channel inputs, we make use of the following subsystems: precoders $\{\pi_n\}_{n=1}^{\infty}$, where π_n precodes n bits; lossless source coders $\{\sigma_n\}_{n=1}^{\infty}$, where σ_n conditionally source codes n channel inputs; and a termination encoder $e_{\kappa_N, \nu}^{\text{term}}$, which encodes a block of κ_N bits, where κ_N is given further below as a function of N, at some rate and probability of error determined by the parameter ν. At this stage, we focus on the basic structure of the encoding process, deferring a precise description of the functions $\{\pi_n\}_{n=1}^{\infty}$, $\{\sigma_n\}_{n=1}^{\infty}$, and $e_{\kappa_N, \nu}^{\text{term}}$ to Sections III-A1)–III-A3).

We begin with the simplifying assumption that we have available a length-t_N sequence $\emptyset^{t_N}[N]$, where[5] $t_N = o_N(N)$, that is perfectly detectable—no false alarms and no missed detections—after passing through the DMC. This assumption is removed in Section V.

The encoder ε is then defined as follows:

Initialization:

$$L_0^{\sigma} = N \quad \Psi_0 = w^N \quad A_0 = 0 \tag{8}$$

$$\varepsilon_0' = \pi_{L_i^{\sigma}}(\Psi_i) \quad L_0 = \ell(\varepsilon_0') \tag{9}$$

for $i = 1, \cdots, B_N$:

$$A_i = A_{i-1} + L_{i-1} \tag{10}$$

$$\Sigma_i = \sigma_{L_{i-1}}(\varepsilon_{i-1}', Y_{A_{i-1}+1}^{A_i}) \tag{11}$$

$$\Psi_i = r((\phi(L_{i-1}), \phi(L_{i-1}^{\sigma}), \Sigma_i), i) \tag{12}$$

$$L_i^{\sigma} = \ell(\Psi_i) \tag{13}$$

$$\varepsilon_i' = \pi_{L_i^{\sigma}}(\Psi_i) \tag{14}$$

$$L_i = \ell(\varepsilon_i'). \tag{15}$$

[5] $o_N(\cdot)$ is the usual order notation with a subscript explicitly denoting the limiting variable, i.e., if $f(N) = o_N(g(N))$, then

$$\lim_{N \rightarrow \infty} \frac{f(N)}{g(N)} = 0.$$

We also use $O_N(\cdot)$ and $\Theta_N(\cdot)$ so that $f(N) = O_N(g(N))$ means that

$$\liminf_{N \rightarrow \infty} \frac{f(N)}{g(N)} \geq 0$$

$$\limsup_{N \rightarrow \infty} \frac{f(N)}{g(N)} < \infty$$

and $f(N) = \Theta_N(g(N))$ means that

$$\liminf_{N \rightarrow \infty} \frac{f(N)}{g(N)} > 0$$

$$\limsup_{N \rightarrow \infty} \frac{f(N)}{g(N)} < \infty.$$

The extra subscript is convenient in expressions where dependencies between variables are implicit.

Final message:

$$G^{\infty} = (\rho(\Sigma_{B_N}),\, \phi(\lceil A_{B_N}/t_N \rceil t_N - A_{B_N}),\, \phi(L_{B_N-1}),\, \phi(L^{\sigma}_{B_N-1}), 0, 1, 0, 1, 0, 1, 0, 1, \cdots) \tag{16}$$

Encoding function:

$$\varepsilon(W^N, Y^{\infty}) = (\varepsilon'_0,\, \varepsilon'_1,\, \cdots,\, \varepsilon'_{B_N-1},\, \Phi^F,\, \emptyset^{t_N}[N],\, e^{\text{term}}_{\kappa_N,\nu}(G_1^{\kappa_N}),\, e^{\text{term}}_{\kappa_N,\nu}(G^{2\kappa_N}_{\kappa_N+1}), e^{\text{term}}_{\kappa_N,\nu}(G^{3\kappa_N}_{2\kappa_N+1}),\, \cdots). \tag{17}$$

In (12), the mapping $\phi : \{0\} \cup \mathbb{N} \rightarrow \{0, 1\}^{\dagger}$ returns a sequence of length $2\lceil \log n \rceil + 2$ corresponding to a binary representation of its integer argument n with each bit repeated once followed by a terminating string $(0, 1)$.[6] The invertible mapping[7] r: $\{0, 1\}^{\dagger} \times \mathbb{N} \rightarrow \{0, 1\}^{\dagger}$ is introduced to ensure that Ψ_i is uniformly distributed over the set $\{0, 1\}^{L_i^{\sigma}}$. To do so, r adds (modulo-2) a pseudorandom Bernoulli-$\frac{1}{2}$ sequence to its first argument, using its second argument as a "seed." As a result, the output of r is indistinguishable from an i.i.d. Bernoulli-$\frac{1}{2}$ source, and r is reversible in the sense that the tuple d can be recovered from $r(d, s)$ and s. In (16), the mapping $\rho : \{0, 1\}^{\dagger} \rightarrow \{0, 1\}^{\dagger}$ repeats each input bit and adds the terminating string $(0, 1)$.[8] In (17), the synchronization sequence $\emptyset^{t_N}[N]$ is used to enable correct parsing of the incoming stream by the receiver, as we discuss in Section III-B. The sequence Φ^F is a fairly arbitrary length-F "filler" sequence to be ignored, where $F = \lceil A_{B_N}/t_N \rceil t_N - A_{B_N} < t_N$; it serves only to ensure that $\emptyset^{t_N}[N]$ is transmitted at an integer multiple of t_N. Note that we have simplified our notation in (17) by suppressing the (potential) dependence of the termination encoder $e^{\text{term}}_{\kappa_N,\nu}$, which may itself be a feedback coder, on the appropriate subsequences of Y^{∞}.

We choose the number of iterations B_N according to

$$B_N = \lceil \log^2 N \rceil \tag{18}$$

to ensure that the expected number of final bits is small enough for sufficiently large N.

To complete a precise specification of the encoder, we precisely define the precoding, source coding, and termination coding subsystems.

1) Precoding Subsystem: To effect a transformation from a sequence of n i.i.d. Bernoulli-$\frac{1}{2}$ random variables (bits) into a sequence that is approximately i.i.d. according to q_X, the precoder $\pi_n : \{0, 1\}^n \rightarrow \mathcal{X}^{\dagger}$ uses two key ideas: 1) that a sequence of variables taking values in a discrete set with cardinality M can be mapped to a real number in $[0, 1)$ via its M-ary expansion and 2) that a real random variable with some desired distribution can be created by applying the inverse cumulative distribution function (cdf) associated with the desired distribution to a uniformly distributed random variable.

The precoding then takes place in three steps: 1) the data bits to be precoded are mapped to a real number $S \in [0, 1)$; 2) an appropriate inverse cdf function $F_{\tilde{X}}^{-1}$ is applied to this real number to form another real number $U \in [0, 1)$; and 3) an appropriate number of M-ary expansion digits of U are taken to be the output of the precoder. These three steps are similar to those taken by a decoder for an arithmetic source coder.

To define π_n precisely, we define the sequence of channel inputs $\pi_n(D^n)$ that correspond to precoding the n data bits D^n. We first transform D^n to a real number $S \in [0, 1)$ according to

$$S = 0_2.D^n + 2^{-n}Z \tag{19}$$

where

$$0_K.a^j = 0_K.a_1a_2 \cdots a_j = \sum_{i=1}^{j} a_i K^{-i} \tag{20}$$

and Z is a random variable that is uniformly distributed over $[0, 1)$. The sole purpose of Z is so S is uniformly distributed over $[0, 1)$ when D^n is uniformly distributed over $\{0, 1\}^n$.

Next, assuming $\mathcal{X} = \{0, 1, \cdots, M-1\}$ (which sacrifices no generality), we define the cdf $F_{\tilde{X}}$ whose inverse is used for the transformation. Let $\{\tilde{X}_k\}_{k=1}^{\infty}$ be an i.i.d. process with marginal pmf q_X. We then map this process onto the unit interval $[0, 1)$ by letting $\tilde{X}$ be a random variable defined by $\tilde{X} = 0_M.\tilde{X}_1\tilde{X}_2 \cdots$, and we let $F_{\tilde{X}}$ be the cdf for $\tilde{X}$.

With the base of all expansions in this section taken to be M, the precoder π_n is defined by[9]

$$T_n(u) = u^{[l_n(u)]} \tag{21}$$

$$\pi_n(D^n) = T_n(F_{\tilde{X}}^{-1}(S)) \tag{22}$$

where the expansion in (21) is M-ary, and $l_n : [0, 1) \rightarrow \mathbb{N}$ is defined as follows to ensure that the output of the precoder stops after enough digits of the M-ary expansion of $F_{\tilde{X}}^{-1}(S)$ have been put out to uniquely determine the first n bits of S. That is, l_n is defined by

$$l_n(u) = \min\{k: F_{\tilde{X}}([0_M.u^{[k]}, 0_M.u^{[k]} + M^{-k})) \subseteq [i2^{-n}, (i+1)2^{-n})\} \quad \text{for } u \in F_{\tilde{X}}^{-1}([i2^{-n}, (i+1)2^{-n})) \quad \text{for } i = 0, \cdots, 2^n - 1. \tag{23}$$

This definition of π_n implies that the precoder is a lossless coder, i.e., if $d^n \in \{0, 1\}^n$, then $\pi_n(d^n)$ is a variable-length tuple whose elements are in $\mathcal{X}$; from n and $\pi_n(d^n)$, we can determine d^n. No knowledge of Z is required by the decoder.

This definition also implies that the distribution of the output of the precoder approximates the desired channel input distribution in the following sense: if the input to the precoder

[6] For example, $\phi(6) = (1, 1, 1, 1, 0, 0, 0, 1)$, since the number 6 has binary representation $(1, 1, 0)$.

[7] We have observed that, in practice, the randomizing function r in (12) is unnecessary; it is, however, convenient for analysis.

[8] So that, for example, $\rho((1, 0, 1)) = (1, 1, 0, 0, 1, 1, 0, 1)$.

[9] As additional notation, for any given number $a \in [0, 1)$, we use $a_{[i]}$ to denote its ith K-ary (base-K) expansion digit, i.e.,

$$a = \sum_{i=1}^{\infty} a_{[i]} K^{-i}, \qquad \text{with } a_{[i]} \in \{0, 1, \cdots, K-1\}.$$

When using this notation, the base K of the expansion is either stated explicitly or clear from context. If a has two K-ary expansions, then $a_{[i]}$ specifically denotes the ith digit of the expansion that ends in an infinite sequence of zeros. Also, $a_{[s]}^{[t]}$, where $s < t$ denotes the tuple $(a_{[s]}, \cdots, a_{[t]})$, and $a_{[1]}^{[t]}$ is abbreviated by $a^{[t]}$.

D^n is uniformly distributed over $\{0, 1\}^n$, then the elements of $\pi_n(D^n)$ form a random process that can be generated by taking an i.i.d. process with marginal pmf q_X and truncating it according to a stopping rule [14].

2) Source Coder: We now define the source coding function $\sigma_n : \mathcal{X}^n \times \mathcal{Y}^n \rightarrow \{0, 1\}^\dagger$. This function compresses its first argument, a sequence $x^n \in \mathcal{X}^n$ representing the channel inputs, using a statistical model for its dependence on its second argument, a sequence $y^n \in \mathcal{Y}^n$ representing the corresponding channel outputs.

Since x^n is generated by the precoder, whose output is approximately an i.i.d. process, and y^n results from passing x^n through the DMC effectively without feedback, we choose the following statistical model for use in the source coder. Let $\{\hat{X}_i\}$ denote an i.i.d. process with marginal pmf q_X, and let $\{\hat{Y}_i\}_{i=1}^{\infty}$ denote the channel output process resulting from passing $\{\hat{X}_i\}$ through the DMC $q_{Y|X}$ without feedback. The source coder then assumes that the probability of x^n given y^n is simply

$$p_{\hat{X}^n|\hat{Y}^n}(x^n|y^n). \tag{24}$$

Source coding is accomplished with this model via a Shannon–Fano strategy [15]. That is, with $\tilde{X} = 0_M . \hat{X}^n$,

$$\sigma_n(x^n, y^n) = u^{[l]} \tag{25}$$

where the expansion in above is binary, and

$$u = F_{\tilde{X}|\hat{Y}^n}(0_M . x^n|y^n) + p_{\hat{X}^n|\hat{Y}^n}(x^n|y^n)/2 \tag{26}$$

$$l = \lceil -\log p_{\hat{X}^n|\hat{Y}^n}(x^n|y^n)\rceil + 1 \tag{27}$$

and $F_{\tilde{X}|\hat{Y}^n}$ is the cdf for $\tilde{X}$ conditioned on $\hat{Y}^n$. The source coder is lossless in the sense that x^n can be recovered from y^n with $\sigma_n(x^n, y^n)$.

Note that the statistical model used in the source coder is inaccurate: Modeling the precoder output as n samples of the i.i.d. process $\{\hat{X}_k\}_{k=1}^{\infty}$ is not strictly correct, because n itself is dependent on the values of $\hat{X}^n$. Nevertheless, we use this source coder in our overall scheme and show that these inaccuracies are inconsequential.

3) Termination Coding Subsystem: We now describe the termination encoder $e^{\text{term}}_{\kappa_N, \nu}$ that we use in the code $c_{\text{DMC}}(N, \nu)$. The termination encoder, as we mentioned earlier, protects κ_N bits with a rate and probability of error determined by its parameter ν. Let us first discuss how many bits κ_N the termination coder should be designed to protect.

For the overall scheme to have high reliability, the final data bits should be encoded within a single κ_N-bit block, on average. For this reason, we choose κ_N according to

$$\kappa_N = \lceil N^{1/4}\rceil. \tag{28}$$

To show that κ_N is sufficiently large, we let B_G denote the number of κ_N-bit blocks required to send the first $\tilde{L}$ bits of G^∞, where

$$\tilde{L} = 2\ell(\Sigma_{B_N}) + 2\lceil \log t\rceil + 2\lceil \log L_{B_N-1}\rceil + 2\lceil \log L^{\sigma}_{B_N-1}\rceil + 10 \tag{29}$$

represents the number of final data bits that are sent by the termination coder up to and including the first appearance of the string $(0, 1, 0, 1)$ in G^∞. Then

$$B_G = \left\lceil \frac{\tilde{L}}{\kappa_N} \right\rceil \tag{30}$$

$$\leq \frac{\tilde{L}}{\kappa_N} + 1 \tag{31}$$

where the inequality follows from the simple ceiling function property $\lceil x\rceil \leq x + 1$. Taking expectations of the right-hand side of (31)

$$E[B_G] \leq 1 + \frac{E[\tilde{L}] + 2}{\kappa_N}. \tag{32}$$

We show in Appendix E that with the precoder and source coder designs given in the previous two sections

$$E[B_G] = 1 + o_N(1) \tag{33}$$

which shows that our choice of κ_N gives the desired behavior.

While any of a wide range of codes can be used as the termination coding scheme that protects the final bits, we choose what we call a *modified Schalkwijk–Barron* (mSB) coding scheme, so named because it can be viewed as a special case of a coder developed by Yamamoto and Itoh [4] in a paper bearing this name in the title. The mSB coder is itself a highly reliable feedback coder, which, as we show in Section IV, gives the overall scheme high reliability.

The mSB coder that we use sends a block of κ_N bits as follows: let x and x' be two elements of $\mathcal{X}$ defined by

$$(x, x') = \arg\max_{(x, x')} D(q_{Y|X}(\cdot|x) \,\|\, q_{Y|X}(\cdot|x')) \tag{34}$$

where $D(\cdot\|\cdot)$ denotes the Kullback–Leibler distance between two pmf's.[10] Assuming the channel has positive capacity, it can be shown easily that the probability of error associated with maximum-likelihood decoding of the two-codeword, length-κ_N codebook consisting of the words $a^{\kappa_N}[0] \triangleq (x, \cdots, x)$ and $a^{\kappa_N}[1] \triangleq (x', \cdots, x')$ is below $2^{-\alpha\kappa_N}$ for some $\alpha > 0$.

Each of the κ_N bits is sent via this two-codeword codebook, i.e., the sequence $a^{\kappa_N}[0]$ is sent for a 0 and $a^{\kappa_N}[1]$ is sent for a 1. After the κ_N bits are sent via this procedure, the transmitter determines whether the receiver has decoded any of the bits incorrectly. Via the union bound, the probability P_{in} that any of these bits are decoded incorrectly can be shown to be $o_{\kappa_N}(1)$. If any errors occur, then the transmitter sends the length-ν sequence $w^\nu \triangleq (x', \cdots, x')$. Otherwise, the length-$\nu$ sequence $c^\nu \triangleq (x, \cdots, x)$ is sent; if the receiver successfully determines that w^ν (and not c^ν) was sent, the process is repeated from the beginning. Note that the κ_N bits are decoded correctly unless w^ν is mistaken for c^ν. Since the probability of this event clearly decreases as ν increases, the parameter ν controls

[10] The Kullback–Leibler distance, also called the information divergence, between two pmf's p and q is given by [15]

$$D(p\|q) = \sum_x p(x) \log \frac{p(x)}{q(x)}.$$

the probability of error for this termination coding scheme. Because mistaking w^ν for c^ν causes a decoding error while the opposite error merely causes a retransmission, the two errors are not equally important. The two sequences are therefore not distinguished with maximum-likelihood decoding—rather, the method described in [4] is used that trades a smaller probability of mistaking w^ν for c^ν for a larger probability of the opposite error.

B. Decoding

Let us now outline the processing that takes place in the associated decoder for the encoder (17). The following high-level description of this processing implicitly defines the stopping functions $\{\chi_i\}_i$ and decoder Δ.

The receiver has two operating modes, "normal mode," in which it starts, and "termination-decoding mode." In normal mode, the receiver saves the data from the incoming bitstream directly to memory without processing it while watching for the occurrence of the sequence $\emptyset^{t_N}[N]$. When this sequence is detected, it enters termination-decoding mode and begins decoding incoming data using the decoder corresponding to the termination code. It concatenates the decoded messages into a long decoded string, appending each newly decoded message to the end of the string. After decoding each block of κ_N bits and appending it to the string, the receiver searches for the termination sequence $(0, 1, 0, 1)$ anywhere in the full string and stops data acquisition when it detects this sequence.

The receiver then proceeds to decode enough of G^∞ to recover Σ_{B_N}, F, L_{B_N-1}, and $L^\sigma_{B_N-1}$. Starting with this information, it decodes the data set acquired in normal mode according to the following recursive procedure:

1) Subtract F from the location at which $\emptyset^{t_N}[N]$ was detected to determine A_{B_N}. Let $i = B_N$.
2) Let $A_{i-1} = A_i - L_{i-1}$. Use the received data $Y_{A_{i-1}+1}^{A_i}$ and L_{i-1} to construct the (*a posteriori*) source-coding pmf that was used to generate Σ_i, and invert the source-coded block Σ_i according to this pmf to obtain ε'_{i-1}.
3) Use L^σ_{i-1} to invert the precoded block ε'_{i-1} to obtain Ψ_{i-1}.
4) Decrement i. If $i \geq 1$, extract Σ_i, L_{i-1}, and L^σ_{i-1} from Ψ_i using the seed i to invert the effect of r, and go to Step 2); otherwise, stop because $\Psi_0 = W^N$, and the message has been determined.

IV. Reliability

The reliability of a coding scheme, also known as its error exponent, gives the asymptotic relationship among the coding scheme's rate, probability of error, and blocklength. In this section, we prove at a high level, leaving most of the details to the Appendix, the following theorem regarding the reliability for the scheme:

Theorem 4.1: Let c_{DMC} be a coding scheme mapping N and ν to the corresponding code consisting of the encoder described in Section III-A with its implied associated stopping functions and decoder. Then, in the sense of Section II, c_{DMC} attains the error exponent function E_{CEC} defined on the interval $(0, I(X;Y))$ by

$$E_{\text{CEC}}(r) = (1 - r/I(X;Y))E_{cw} \tag{35}$$

where

$$E_{cw} = \max_{x\in\mathcal{X},\, x'\in\mathcal{X}} D(q_{Y|X}(\cdot|x)\|q_{Y|X}(\cdot|x')). \tag{36}$$

Remarks: Burnashev [5] has shown that E_{CEC} is an *upper* bound to the error exponent of any variable-length feedback transmission scheme for a DMC_f. This scheme therefore attains the largest possible error exponent at all rates. Finally, note that E_{cw} may be infinite (e.g., if an output symbol has positive probability under one channel input but zero probability under another as in a "Z-channel") in which case the error exponent becomes degenerate and somewhat meaningless.

Proof: This theorem follows directly from three key properties of the coding scheme, which we highlight to begin the proof. We call these properties "Subsystem Properties," because each focuses on a key aspect of one of the subsystems. The reader should be aware, though, that some of the properties do depend on multiple subsystems and the way they interconnect. The main point is that if we choose a different design for the precoder, source coder, termination coder, or synchronization subsystem, then as long as these properties hold, then Theorem 4.1 also holds.

Subsystem Property 1: If D^n is uniformly distributed over $\{0, 1\}^n$, then there exists a constant $0 < C_\pi < \infty$ that is independent of n such that

$$E[\ell(\pi_n(D^n))] \leq (n + C_\pi)/H(X). \tag{37}$$

Subsystem Property 2: The precoders $\{\pi_n\}_{n=1}^\infty$ and the source coders $\{\sigma_n : \mathcal{X}^n \times \mathcal{Y}^n \to \{0, 1\}^\dagger\}_{n=1}^\infty$ are such that there exists a function λ such that

$$E[L^\sigma_{i+1}] \leq E[L_i]\, H(X|Y) + \lambda(E[L_i]), \qquad \text{for } i = 0, \cdots, B_N - 1 \tag{38}$$

where λ also has the properties that $\lambda(x) = o_x(x)$ and λ is nonnegative, monotonically increasing, and concave $(\cap)$ over $[1, \infty)$.

Subsystem Property 3: The termination coding scheme c^{term} takes two parameters κ and ν and returns a code. For any $\alpha > 0$, there exists a sequence of parameters $\{(\kappa_n, \nu_n(\alpha))\}_{n=1}^\infty$ such that $c^{\text{term}}(\kappa_n, \nu_n(\alpha))$, whose corresponding encoder is denoted $e^{\text{term}}_{\kappa_n, \nu_n(\alpha)}$, encodes $\kappa_n = \lceil n^{1/4} \rceil$ message bits into an average number of channel inputs $\eta_n = \alpha n + o_n(n)$ and has error probability $P_{e,\,\text{term},\,n}$ bounded according to

$$P_{e,\,\text{term},\,n} < \exp_2\{-\eta_n(E_{cw} - o_n(1))\} \tag{39}$$

where $\exp_2(x) \triangleq 2^x$.

Subsystem Property 1 is proven in Appendix A, Subsystem Property 2 is proven in Appendix B, and Subsystem Property 3 is proven in Appendix C.

Using these key properties, we can now prove the theorem as follows: let $r < I(X; Y)$ be given, and let us construct a sequence of codes with corresponding sequence of rates and error probabilities satisfying (6) and (7).

Intuition from the third illustration in Section I suggests that the expected transmission length $E[L^*]$ of a code sending N message bits and using a termination coder that puts out a sequence of average length η, satisfies

$$E[L^*] \approx \frac{N}{I(X; Y)} + \eta. \tag{40}$$

This equation in turn suggests that a sequence of codes with rate converging to r is $\{c_{\text{DMC}}(n, \nu_n)\}_{n=1}^{\infty}$, where ν_n is a termination code parameter giving the termination code a corresponding expected length of $\eta_n(r) + o_n(n)$, where

$$\eta_n(r) = \frac{n}{r}\left(1 - \frac{r}{I(X; Y)}\right). \tag{41}$$

That an appropriate sequence of parameters $\{\nu_n\}$ exists is guaranteed by Subsystem Property 3. Let us examine this sequence of codes $\{c_{\text{DMC}}(n, \nu_n)\}_{n=1}^{\infty}$ more closely to verify that it behaves as desired.

To prove that the sequence of rates corresponding to this sequence of codes satisfies (6), we first develop a bound on the expected transmission length $E[L^*]$ of the code $c_{\text{DMC}}(N, \nu_N)$ as follows. Using the notation of Section III, first consider the termination-coded transmission of the sequence G^{∞} defined in (16). If the receiver fails to detect the sequence $(0, 1, 0, 1)$ when it first appears in the transmission G^{∞}, then it is detected in a subsequent block because this coded sequence is repeated indefinitely thereafter (cf. (16)). Moreover, for each of these blocks the probability of a missed detection is also less than $P_{e,\,\text{term},\,N}$, the probability of error associated with the termination coder used by $c_{\text{DMC}}(N, \nu_N)$. Thus the expected length of the transmission starting with the length-t_N transmission of $\emptyset^{t_N}[N]$ until termination is less than

$$\mu_{\text{I}} \triangleq t_N + \left(E[B_G] + \frac{P_{e,\,\text{term},\,N}}{(1 - P_{e,\,\text{term},\,N})}\right)(\eta_N(r) + o_N(N)). \tag{42}$$

Furthermore, the expected length of the transmission before $\emptyset^{t_N}[N]$ in (17) is

$$\mu_{\text{II}} \triangleq \sum_{i=0}^{B_N - 1} E[L_i]. \tag{43}$$

Hence, the total expected length of the transmission is bounded according to

$$E[L^*] < \mu_{\text{I}} + \mu_{\text{II}}. \tag{44}$$

The following lemma, which is proven in Appendix D, uses Subsystem Properties 1 and 2 to upper-bound μ_{II}.

Lemma 4.1:

$$\mu_{\text{II}} \le \left(\frac{N}{H(X) - H(X|Y)}\right) + o_N(N). \tag{45}$$

And the next lemma, which is proven in Appendix E, uses Subsystem Property 3 to upper-bound μ_{I}.

Lemma 4.2:

$$\mu_{\text{I}} < t_N + (1 + o_N(1))(\eta_N(r) + o_N(N)). \tag{46}$$

Since $t_N/N = o_N(1)$, by substituting (45) and (46) into (44), we get that

$$E[L^*]/N < \frac{1}{I(X; Y)} + (1 + o_N(1))\,\frac{\eta_N(r)}{N} + o_N(1). \tag{47}$$

This inequality with (41) implies that the rate R_N of $c_{\text{DMC}}(N, \nu_N)$ satisfies $R_N > r + o_N(1)$.

The last step in showing that E_{CEC} is attainable is to find the probability of error P_N corresponding to $c_{\text{DMC}}(N, \nu_N)$. With the decoder described above, the invertibility of the source coder and precoder—together with the perfect detectability of $\emptyset^{t_N}[N]$—mean that decoding errors in the overall scheme occur only if one of the B_G blocks that is termination-coded is decoded incorrectly. Since P_N equals the probability of such an event, it can be union bounded above according to

$$P_N \le E[B_G]\,P_{e,\,\text{term},\,N}. \tag{48}$$

Inequality (39) with (33) gives an upper bound on the right-hand side of (48). Substituting (41) into this upper bound, taking the log, and multiplying by $-R_N/N$ yields

$$-\frac{R_N \log P_N}{N} \ge \frac{R_N}{r}\left(1 - \frac{r}{I(X; Y) + o_N(1)}\right)(E_{cw} - o_N(1)) \tag{49}$$

$$\ge E_{\text{CEC}}(r) - o_N(1) \tag{50}$$

where (50) follows from the fact that $R_N > r + o_N(1)$. Since these results hold for $c_{\text{DMC}}(N, \nu_N)$ for arbitrary N, the theorem is proved. □

V. Removing the Perfect-Detection Assumption

In the previous section, we assume that a perfectly detectable sequence $\emptyset^{t_N}[N]$ exists. Since, in general, no such sequence exists when the forward channel is a DMC, we must modify the coding scheme before it can work with a forward channel that is a DMC. In this section, we outline modifications to our basic iterative coding scheme that allow us to remove the assumption that $\emptyset^{t_N}[N]$ is perfectly detectable.

To develop the modified scheme, we first construct $\emptyset^{t_N}[N]$ out of legitimate channel inputs from $\mathcal{X}$. Let a be an element of $\mathcal{X}$ defined by

$$a = \arg\max_{x \in \mathcal{X}} D(q_{Y|X}(\cdot|x) \| p_Y) \tag{51}$$

and then let $\emptyset^{t_N}[N] = (a, \cdots, a)$. The encoder uses the sequence in the same way, communicating the time index A_{B_N} defined in (10) to the decoder by sending $\emptyset^{t_N}[N]$ at time $A_{B_N} + F + 1$ and later sending bits describing F. The decoder tests for the presence of this sequence in each new incoming block of t_N samples, using a minimum-probability-of-error detector δ_{t_N} of the form

$$\delta_{t_N}(y^{t_N}) = \begin{cases} 1, & \text{if } \prod_{i=1}^{t_N} q_{Y|X}(y_i|a) \ge \prod_{i=1}^{t_N} p_Y(y_i) \\ 0, & \text{otherwise.} \end{cases} \tag{52}$$

There is now the possibility that δ_{t_N} returns a 0 when $\emptyset^{t_N}[N]$ is sent over the channel (missed detection) or that δ_{t_N} returns a 1 when $\emptyset^{t_N}[N]$ is not sent over the channel (false alarm). As t_N increases, the probability of either sort of error can be shown to be less than $2^{-\beta t_N}$ for some $\beta > 0$.

We now encounter a dilemma in choosing t_N: if we accept that $W^N \neq \hat{W}^N$ whenever a false alarm or missed detection occurs, then we need to choose t_N proportional to N to maintain probability of error decaying exponentially in N. But choosing t_N proportional to N causes the rate to decrease by an asymptotically nonnegligible amount. On the other hand, if we choose $t_N = o_N(N)$, then the probability of error does not decrease exponentially in N.

The solution is to choose $t_N = o_N(N)$ but to use feedback to detect missed detections and false alarms, allowing the encoder to take corrective action. We therefore choose t_N according to

$$t_N = \lceil \sqrt{N} \rceil \tag{53}$$

which ensures that the probability of a false alarm or missed detection occurring at any time during the transmission decays to 0 as $N \to \infty$. The scheme is then modified as follows.

As in the idealized case, the receiver has two modes of operation: normal mode and termination-decoding mode. In normal mode, at every time k that is an integer multiple of t_N, the receiver tests the most recent t_N channel outputs to see whether $\delta_{t_N}(Y_{k-t_N+1}^k) = 1$. The receiver enters termination-decoding mode if and only if this condition holds true. Once in termination-decoding mode, the receiver decodes each incoming block to find the message coded in that block. Concatenating these messages, the receiver stops receiving when it is in termination-decoding mode and finds the sequence $(0, 1, 0, 1)$ somewhere in the concatenated message.

The transmitter knows exactly what the receiver is doing via the feedback. Hence, it can exploit the fact that the receiver always enters termination-decoding mode on detection of $\emptyset^{t_N}[N]$ by sending the receiver a message regarding its detection of $\emptyset^{t_N}[N]$. In particular, if a false alarm occurs, then the sequence

$$(1, 0, W_1, W_1, W_2, W_2, \cdots, W_N, W_N, 0, 1, 0, 1, 0, 1, \cdots)$$

is transmitted in blocks of κ_N bits using the termination coder. The first two elements of the sequence, $(1, 0)$, inform the receiver that a false alarm has occurred and that the remainder of the sequence is to be regarded as the original message. Note that even if some of these κ_N-bits blocks are decoded incorrectly, the receiver eventually sees the sequence $(0, 1, 0, 1)$ and stops data acquisition.

In the case of a missed detection—that is, when $\emptyset^{t_N}[N]$ is transmitted but not detected by the receiver—the transmitter resends $\emptyset^{t_N}[N]$ until it is detected by the receiver. After detection, the transmitter sends $(\phi(C_{\text{MD}}), G^\infty)$ coded in κ_N-bit blocks using the termination coder. The sequence $\phi(C_{\text{MD}})$ encodes the number C_{MD} of missed detections that occurred. From this information, the receiver can correctly ascertain the value of $A_{B_N} + 1$.

In Fig. 2, a flowchart giving an outline of how the scheme makes use of the synchronization subsystem is shown.

1) Reliability of the Modified Scheme: It is proven in Appendix G that Theorem 4.1 continues to hold for the modified scheme when a synchronization subsystem can be designed with the following property:

Subsystem Property 4: With $\{t_N\}_{N=1}^\infty$, a sequence satisfying $t_N = o_N(1)$, the sequence of detector-sequence pairs $\{(\delta_{t_N}, \emptyset^{t_N}[N])\}_{N=1}^\infty$ is such that the false-alarm and missed-detection probabilities $P_{\text{FA}, \text{t}_\text{N}}$ and $P_{\text{MD}, \text{t}_\text{N}}$, respectively, associated with each use of δ_{t_N} by the receiver, satisfies

$$P_{\text{FA}, t_N} < o_N(N^{-3}) \tag{54}$$

$$P_{\text{MD}, t_N} < M_{\text{MD}} + o_N(1) \tag{55}$$

for some constant $M_{\text{MD}} < 1$.

In Appendix F, Subsystem Property 4 is shown to hold for the synchronization subsystem design given by (51)–(53).

VI. Complexity

In what follows, we show that the above coding scheme can be carried out with time and space complexity that is linear in the length of the number of channel inputs used. This linear time complexity is clearly the lowest time complexity (in terms of asymptotic order of growth) achievable by any coding scheme.

A. Time and Space Complexity for the Transmitter and Receiver

To show that the encoder and decoder can be implemented with linear time and space complexities, we show that the four constituent subsystems—precoding, source coding, synchronization, and termination coding—can individually be implemented with linear time and space complexities.

Precoding and postcoding (precoding inversion) operate nearly identically to arithmetic source-coding decoders and encoders, respectively, for i.i.d. sources. The dominant computation required for arithmetic source coding and decoding is the computation of the relevant cdf. Because the cdf can be computed using a recursive algorithm [15], it can easily be seen that arithmetic coding can be performed with time cost that is proportional to the sum of the number of inputs and outputs under the uniform-cost criterion. Space cost, excluding buffering of input and output, can be easily seen to be a constant, so the total space cost including buffering requirements is proportional to the sum of the number of inputs and outputs. Thus the total time cost and total space cost associated with precoding in the transmitter and postcoding in the receiver are linear in L^*.

The source encoding and decoding subsystems are based on Shannon–Fano source coding, which can also be carried out using the arithmetic source-coding algorithm. Use of arithmetic coding again results in the time costs of the source encoder and decoder both being linear in the length of their inputs plus outputs. Space cost is again constant. Therefore, the time complexities and space complexities associated with source encoding and decoding are also $O_{L^*}(L^*)$.

The synchronization subsystem requires that the transmitter send the synchronization sequence and that the decoder test

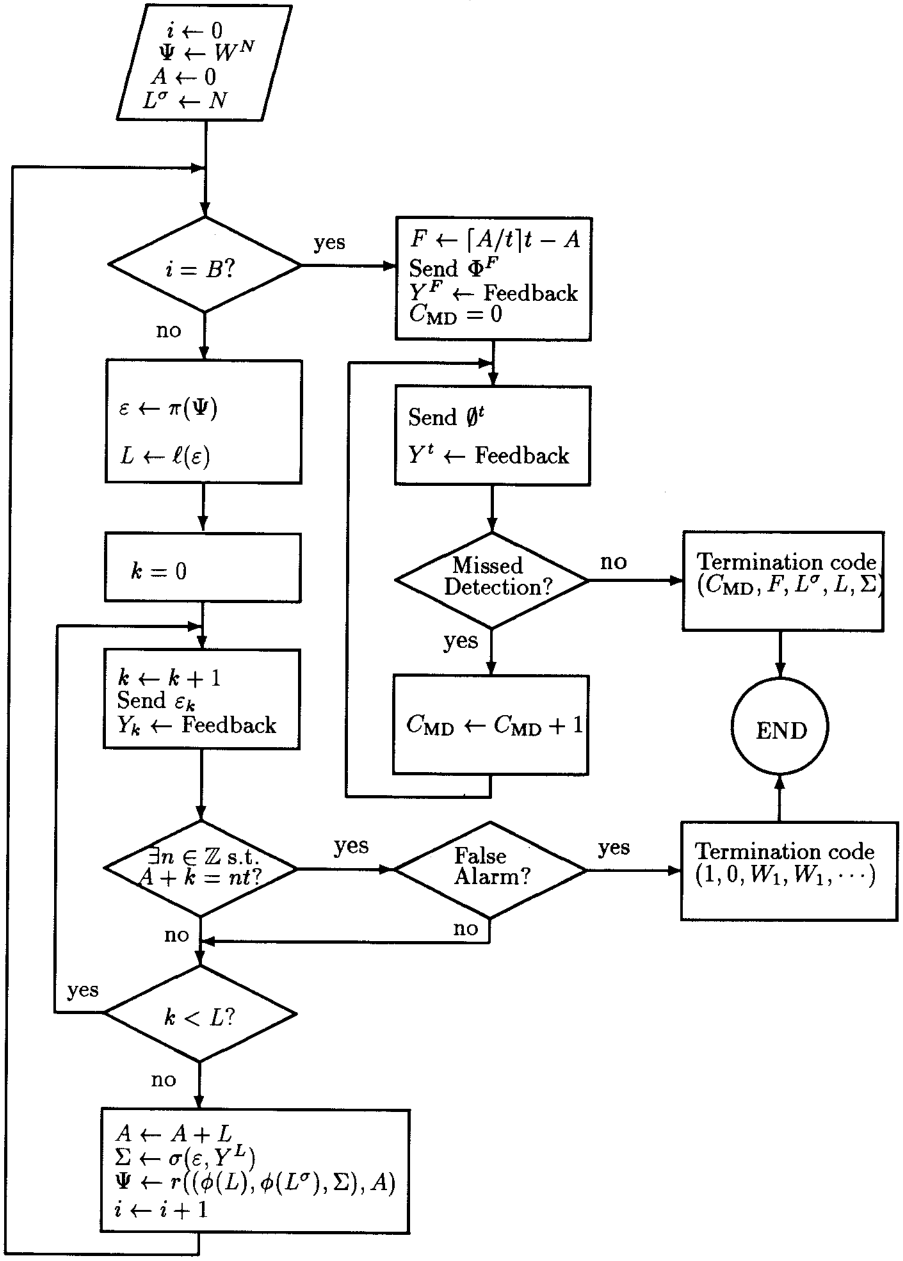
$i \leftarrow 0$
$\Psi \leftarrow W^N$
$A \leftarrow 0$
$L^\sigma \leftarrow N$
$i = B$?
yes
no
$F \leftarrow \lceil A/t \rceil t - A$
Send Φ^F
$Y^F \leftarrow$ Feedback
$C_{\mathrm{MD}} = 0$
$\varepsilon \leftarrow \pi(\Psi)$
$L \leftarrow \ell(\varepsilon)$
Send $\emptyset^t$
$Y^t \leftarrow$ Feedback
$k = 0$
Missed Detection?
no
yes
Termination code $(C_{\mathrm{MD}}, F, L^\sigma, L, \Sigma)$
$k \leftarrow k + 1$
Send ε_k
$Y_k \leftarrow$ Feedback
$C_{\mathrm{MD}} \leftarrow C_{\mathrm{MD}} + 1$
END
$\exists n \in \mathbb{Z}$ s.t. $A + k = nt$?
yes
False Alarm?
yes
Termination code $(1, 0, W_1, W_1, \cdots)$
no
no
yes
$k < L$?
no
$A \leftarrow A + L$
$\Sigma \leftarrow \sigma(\varepsilon, Y^L)$
$\Psi \leftarrow r((\phi(L), \phi(L^\sigma), \Sigma), A)$
$i \leftarrow i + 1$

Fig. 2. Encoding modified for imperfect detection of $\emptyset^{t_N}[N]$. The notation "$Y_k \leftarrow$ Feedback" indicates that one sample of feedback is retrieved and stored in Y_k. Similarly, "$Y^a \leftarrow$ Feedback" indicates that a samples are retrieved and stored in Y^a.

for the sequence every t_N samples using δ_{t_N}. Sending the sequence clearly takes a number of operations linear in t_N and requires t_N buffer registers to store the sequence. Each use of δ_{t_N} clearly costs time linear in t_N under the uniform-cost criterion. Space cost is clearly $O_N(t_N)$. Since δ_{t_N} is used fewer than L^*/t times, the total time complexity attributable to the synchronization subsystem in the receiver is linear in L^*. Each time δ_{t_N} is used, it can reuse its registers, so the total space cost is only $O_N(t_N)$. The transmitter must also perform each of these hypothesis tests to determine the state of the receiver, so it shares the same complexity.

The number of computations required for mSB decoding depends on the specific inner code used, but is at most

$$O_N(\kappa_N{}^2) + O_\nu(\nu) \tag{56}$$

under the uniform-cost criterion for each time the inner codeword and corresponding length-ν verification message are sent. The first term accounts for decoding of the inner code at the receiver (which must also be replicated at the transmitter). The second term accounts for the computation required for the transmitter to send c^ν or w^ν and for the receiver to distinguish the two sequences. For the important case in which $\nu \propto N$, we can write $\kappa_N = O_\nu(\nu^{1/4})$, and the two terms in (56) can be combined into $O_\nu(\nu)$, i.e., the time and space cost of a single use of the mSB coder is nearly proportional to the number of channel inputs used by the coder. Since the total number of channel uses due to mSB coding is less than L^*, the total computation due to mSB decoding must also be $O_{L^*}(L^*)$. The time complexity of mSB encoding is less than for decoding, so it can also be shown to be $O_{L^*}(L^*)$. Space complexity for the mSB encoder and decoder is $O_\nu(\nu)$ under both cost criteria, since c^ν and w^ν and the inner codebook, all of which are composed of discrete-valued sequences, must be stored.

Summarizing, each subsystem can be implemented with time and space complexity that is $O_{L^*}(L^*)$ under the uniform-cost criterion, and therefore so can the overall scheme. It follows that, at a fixed rate, the expected number of computations and amount of storage required by the transmitter and receiver is $O_N(N)$. More detailed characterizations of the behavior of these algorithms on finite-length register digital computers are given in [11].

B. Uniform Versus Nonuniform Complexity

In comparing this scheme to other low-complexity coding schemes, we find that concatenated codes [16] with linear complexity can also be designed. Specifically, by using codes developed by Spielman in [17] as outer codes and using a randomly selected inner code, one can obtain concatenated codes requiring computations proportional to blocklength with exponentially decaying probability of error at any rate below capacity. It may be tempting then to conclude that feedback offers no advantages in terms of complexity.

But there is an important distinction between the linear complexity of the feedback scheme we have just introduced and the linear complexity of such a concatenated code. A concatenated coding scheme whose inner code is decoded via exhaustive-search maximum-likelihood decoding requires more computation per message bit as its operating rate increases. That is, at a particular value of the rate R, the computation per message bit is independent of the number of message bits. But the computation per message bit depends heavily on the rate and increases rapidly and without bound as the rate approaches capacity. While the problem can be mitigated by using special inner codes that can be decoded with more computationally efficient decoders, no such capacity-achieving codes and corresponding decoders appear to be known.

On the other hand, it is straightforward to verify that the feedback scheme we have introduced does not behave in this way. The rate of the feedback scheme can be made to increase to capacity by letting $\nu/N \to 0$ and $N \to \infty$. Computations per input sample need not grow without bound as these two limits are approached. There must therefore exist a positive number U *independent* of the rate R and the number of message bits N such that the average number of computations per channel input required for encoding and decoding is less than U for any R below the channel capacity.

We say that our feedback scheme has *uniform* linear complexity, while the above concatenated scheme is an example of a scheme with *nonuniform* linear complexity. The difference has important consequences in terms of what rates are actually achievable in practice.

VII. Implementation and Practical Issues

Thus far, we have concerned ourselves with certain fundamental theoretical aspects of the feedback coding scheme. Many issues related to implementation and practical use of the scheme remain and are discussed in this section.

A. Length Variations

Because our scheme produces variable-length transmissions, buffer overflows are possible. However, for large blocklengths, preliminary experiments suggest that variations in transmission length due to the varying lengths of the precoder and source-coder outputs are modest; for example, in an experiment in which the sample mean of $A_{B_N} + F$ was about 204 500, the maximum value of $A_{B_N} + F$ in 100 trials was 206 336, and the sample standard deviation of $A_{B_N} + F$ was about 794. This behavior is not surprising, because for long blocks, most of the relevant sequences are "typical" and are compressed to "typical" lengths, which are close to the corresponding expected length.

Note that it is possible to design a scheme that uses lossy (but effectively lossless) precoders and source coders with fixed-length inputs and outputs. This variation of the scheme is particularly important for analysis in the next section and is described in detail in [11]. The scheme has the advantages that it needs no synchronization subsystem and has output length that is more easily characterized analytically.

B. Structuring Computation

We have not accounted in the foregoing analyses for the fact that computation may take time. Suppose that we have a computer that does a fixed number of computations per unit of time. In the framework we have described in this paper, if we assume that the precoders and source coders require the full block of input before they can compute their outputs, a computational delay arises between iterations that is proportional to the length of the transmission on the previous iteration. If we send information-free filler during the delay, then the rate is reduced, because we waste a number of channel inputs proportional to N.

Fortunately, we can structure the computation so that computation is performed while useful, information-bearing data rather than information-free filler, is being sent. The technique, which we call *interleaving*, is described as follows: using the scheme variation mentioned in Section VII-A that uses a fixed-length precoder and source coder, a $2N$-bit message is sent as follows.

- Send the first length-$N/H(X)$ block of precoded data.
- Send the second length-$N/H(X)$ block of precoded data. While sending, source code and precode the first $N/H(X)$ transmissions into $NH(X|Y)/H^2(X)$ new inputs.
- Send these $NH(X|Y)/H^2(X)$ inputs. While sending, source code and precode the second $N/H(X)$ transmissions into $NH(X|Y)/H^2(X)$ new inputs.
- Send these $NH(X|Y)/H^2(X)$ inputs. While sending, etc.

After approximately $2N/I(X;Y)$ samples have been sent, we send the final coded block of data and the verification message. Note that to support this technique, the computer must be fast enough to process n channel inputs in the time required to send $nH(X|Y)/H(X)$ channel inputs.

C. Feedback Delay and Noisy Feedback

When communicating over large distances, a significant delay in feedback may be present. That is, the transmitter at time k may only know Y^{k-d}, where d is some fixed delay. The primary effect of this delay is that at the beginning of an iteration, the transmitter may not yet have enough feedback to start the iteration. In this case, the transmitter may send information-free filler data as it waits for feedback. This filler data wastes at most $B_N d$ total channel inputs, which is negligible as $N \to \infty$. If feedback delays are large compared to the desired blocklength, however, one could use a multiplexing strategy whereby one sends another message during the periods that would otherwise be idle.

Noisy feedback can be accommodated by applying error-correcting coding to the feedback link; a method for doing so while maintaining low complexity and overall error probabilities that decay exponentially with blocklength is discussed in [11].

D. Simulation

To verify that our scheme behaves as predicted by our analysis and that the asymptotic performance is approachable in practice, we implemented the scheme and simulated its performance on a digital computer with finite-length registers.

To simultaneously demonstrate that the coding scheme is viable on continuous-valued channels, we applied it to the Gaussian channel. To use the scheme on a continuous-valued channel, the channel inputs and outputs must be quantized and the corresponding channel transition probabilities determined. In a preliminary experiment, we used the fifteen input symbols $\{-7, -6, -5, \cdots, 5, 6, 7\}$ and chose an approximately Gaussian input distribution with variance 4.0 as an input to a discrete-time channel with zero-mean additive white Gaussian noise of variance 4.0. We then quantized the output to the twenty-one symbols $\{-10, -9, -8, \cdots, 8, 9, 10\}$. We simulated the Gaussian channel to empirically calculate the channel transition probabilities for this quantized channel, and used this channel model in our coder. With $N = 10^6$, and $\nu = 1000$, our coder achieved a rate[10] of 0.483. The probability of error can be determined to be less than $1 - F(\sqrt{\nu})$, where F is the cdf for a unit-variance, zero-mean Gaussian random variable, which is upper-bounded [18] by $\exp\{-\nu/2\}/\sqrt{2\pi\nu}$. Comparing this performance with the capacity of the discrete-time Gaussian channel with a 0-dB signal-to-noise ratio, which is 0.50-bit/channel input, our scheme appears very promising for use with continuous-valued channels and has certain advantages over the scheme of Schalkwijk *et al.* [1]. Namely, our scheme can be easily adapted to achieve rates near capacity on non-Gaussian channels such as fading channels and also allows quantized feedback.

VIII. Concluding Remarks

In this paper, we developed the compressed-error-cancellation framework and used it to develop a coding scheme for DMC$_{\mathrm{f}}$'s with optimal reliability and minimal complexity. As an illustration of the broader applicability of this framework, in [11] and [19] rich generalizations of the scheme are developed for use on channels with memory, unknown channels, multiple-access channels, and channels with partial feedback. All retain the underlying block-oriented structure of the scheme developed in this paper. As such, one of several interesting directions for future research would be exploring sequential counterparts to this family of schemes within the compressed-error cancellation framework, and interpreting the results in relation to Horstein's sequential schemes [2].

Appendix A
Proof of Subsystem Property 1

To upper-bound $E[\ell(\pi_n(D^n))]$ as in (37), it is useful to upper-bound l_n by a different function $\tilde{l}_n$, which is easier to analyze.

Lemma A.1: Let $b_i = F_{\tilde{X}}^{-1}(i2^{-n})$ for $i = 0, \cdots, 2^n$, and let $\gamma(x, y)$ denote the index of the first M-ary expansion digit at which x and y differ, i.e., let $\gamma : [0, 1)^2 \to \mathbb{N}$ be defined by $\gamma(x, y) = \min\{k \in \mathbb{N}: x_{[k]} \neq y_{[k]}\}$. Then define $\tilde{l}_n$ by

$$\tilde{l}_n(u) = \begin{cases} \gamma(u, b_{i+1}), & \text{if } \bar{b}_{i,\,i+1} \le u < b_{i+1} \\ \gamma(u, b_i), & \text{if } b_i \le u < \bar{b}_{i,\,i+1} \end{cases} \qquad \forall i = 0, 1, \cdots, 2^n - 1 \tag{57}$$

where

$$\bar{b}_{i,\,i+1} = 0_M.\, b_{i+1}^{[\gamma(b_i,\, b_{i+1})]}$$

(for example, if $\mathcal{X} = \{0, 1\}$, $b_3 = 0_2.010011\cdots$, and $b_4 = 0_2.01100101\cdots$ then $\bar{b}_{3,4} = 0_2.011$).

Then

$$l_n(u) \le \tilde{l}_n(u), \qquad \text{for all } u \in [0, 1). \tag{58}$$

Proof: To prove the lemma, we need only show that

$$[0_M.\tilde{T}_n(u), 0_M.\tilde{T}_n(u) + M^{-k}) \subseteq [b_i,\, b_{i+1})$$

where $\tilde{T}_n(u) = u^{[\tilde{l}_n(u)]}$.

To see that this fact holds, first suppose that $u \in [\bar{b}_{i,\,i+1},\, b_{i+1})$. Next, notice that

$$\bar{b}_{i,\,i+1} = 0_M.\, b_{i+1}^{[\gamma(b_i,\, b_{i+1})]}0000\cdots.$$

Since b_{i+1} starts with the same first $\gamma(b_i,\, b_{i+1})$ digits, any element in $[\bar{b}_{i,\,i+1},\, b_{i+1})$ must also start with the same first $\gamma(b_i,\, b_{i+1})$ digits. Since $\tilde{T}_n$ truncates u only after its M-ary expansion *differs* from that of b_{i+1}, $0_M.\tilde{T}_n(u)$ must also begin with these digits. Hence, $0_M.\tilde{T}_n(u) \ge \bar{b}_{i,\,i+1}$. Next, since $u < b_{i+1}$, the M-ary expansion digit at which u first differs

[10] The theoretical capacity of this DMC approximation to the Gaussian channel is 0.493 bit/input. In our simulation, this rate was not approached more closely in part because our implementation of the source coder and precoder used 32- rather than 64-bit integers, which did not allow for sufficiently accurate characterization of the channel transition pmf. This shortcoming can easily be remedied with a more sophisticated implementation.

from b_{i+1} must be smaller than the corresponding digit of b_{i+1}. Therefore, $0_M.\tilde{T}_n(u)+M^{-\ell(\tilde{T}_n(u))}$, which corresponds to $\tilde{T}_n(u)$ with its last element incremented by one, may at most equal b_{i+1}. Hence,

$$[0_M.\tilde{T}_n(u), 0_M.\tilde{T}_n(u)+M^{-\ell(\tilde{T}_n(u))}) \subseteq [\overline{b}_{i,\,i+1}, b_{i+1}).$$

Next, suppose that $u \in [b_i,\, \overline{b}_{i,\,i+1})$. First, it is clear that $0_M.T_n(u) > b_i$, since $u > b_i$, and $u_{[\tilde{l}_n(u)]} > b_{i,\,[\tilde{l}_n(u)]}$. Then note that

$$\begin{aligned}\overline{b}_{i,\,i+1} = & 0_M.b_i^{[\gamma(b_i,\,b_{i+1})-1]}((b_{i+1})_{[\gamma(b_i,\,b_{i+1})]}-1)\\ & \cdot(M-1)(M-1)(M-1)\cdots. \end{aligned} \tag{59}$$

Therefore,

$$\begin{aligned} & 0_M.T_n(u)+M^{-\tilde{l}_n(u)}\\ & \qquad = 0_M.T_n(u)(M-1)(M-1)(M-1)\cdots \le \overline{b}_{i,\,i+1}\end{aligned}$$

since every M-ary expansion digit of u is less than or equal to the corresponding expansion digit of $\overline{b}_{i,\,i+1}$. Hence, the lemma follows. $\bigtriangledown$

We now prove that $E[\tilde{l}_n(F_{\tilde{X}}^{-1}(S))] < (n+C_\pi)/H(X)$ for some $C_\pi < \infty$, where S is defined in (19). Subsystem Property 1 then follows from this bound with the lemma above.

Consider a device that takes the random process $F_{\tilde{X}}^{-1}(S)_{[1]}$, $F_{\tilde{X}}^{-1}(S)_{[2]}, \cdots$, and traverses an M-ary tree until a leaf is reached. (An M-ary tree is a tree in which every node has M children.) We let the stopping rule $\tilde{l}_n$, which is nonanticipatory (as a stopping rule must be) and deterministic, define the leaves and hence the tree. An example of how the tree is traversed is the following: If $M=2$, starting from the root of the tree, we branch to the left if the first process value is zero, and branch to the right if it is one; now being at a new node, we branch to the left if the second process value is zero, and branch to the right if it is one; we continue until a leaf is reached. Each value of S leads to a different leaf of the tree. We can think of $V = \tilde{T}_n(F_{\tilde{X}}^{-1}(S))$ as the random leaf at which we end, and of $\ell(V)$ as the depth of this leaf. It is shown in [20] that

$$E[\ell(V)] = H(V)/H(X). \tag{60}$$

We now show that $H(V) < n + C_\pi$, where C_π is independent of n, which gives us the desired upper bound $E[\ell(V)] \le (n+C_\pi)/H(X)$. Let

$$\mathcal{V} = \{\tilde{T}_n(u)\}_{u\in[0,\,1)} \subset \{0,\,1,\,\cdots,\,M-1\}^\dagger$$

be the set of all leaves of the tree. Since the stopping rule that defines the leaves is deterministic, the probability of a leaf $v \in \mathcal{V}$ can be written

$$p_V(v) = \Pr\{F_{\tilde{X}}^{-1}(S) \in [0_M.v,\, 0_M.v+M^{-\ell(v)})\}$$

which implies that

$$p_V(v) = \prod_{i=1}^{\ell(v)} q_X((0_M.v)_{[i]}).$$

Now, let us evaluate the entropy of V, which is, by definition,

$$H(V) = \sum_{v\in\mathcal{V}} p_V(v) \log \frac{1}{p_V(v)}.$$

To do so, we divide the sum into manageable portions as follows. Consider for now only the leaves $v \in \mathcal{V}$ for which $0_M.v \in [b_i,\, b_{i+1})$, for some i. Define the sets

$$\begin{aligned}\mathcal{V}_i &= \{v \in \mathcal{V} | 0_M.v \in [b_i,\, b_{i+1})\}\\ \mathcal{V}_i^+ &= \{v \in \mathcal{V} | 0_M.v \in [\overline{b}_{i,\,i+1},\, b_{i+1})\}\end{aligned}$$

and

$$\mathcal{V}_i^- = \{v \in \mathcal{V} | 0_M.v \in [b_i,\, \overline{b}_{i,\,i+1})\}$$

all of which have a countable number of elements. Further restrict consideration to the leaves $v \in \mathcal{V}_i^+$. Let v_i^+ be the element of $\mathcal{V}_i^+$ such that $0_M.v_i^+$ is the smallest member of $\{0_M.v\}_{v\in\mathcal{V}_i^+}$. Note that $\ell(v_i^+)$ is also the smallest element of $\ell(\mathcal{V}_i^+)$, although other elements of $\mathcal{V}_i^+$ may have equal length. Label the elements of $\mathcal{V}_i^+$ according to their length l and their lth M-ary-expansion digit. That is, let $v_i^{l,\,m}$ be the element of $\mathcal{V}_i^+$ that satisfies $\ell(v_i^{l,\,m}) = l$ and $(0_M.v_i^{l,\,m})_{[l]} = m$. Note that $(b_{i+1})_{[l]}$ is the number of elements in $\mathcal{V}_i^+$ with length l. Now, the probability of a leaf $v_i^{l,\,m} \in \mathcal{V}_i^+$ can be written as

$$p_V(v_i^{l,\,m}) = \frac{p_V(v_i^+)}{q_X((0_M.v_i^+)_{[\ell(v_i^+])})} \prod_{k=\ell(v_i^+)}^{l} q_X((0_M.v_i^{l,\,m})_{[k]}) \tag{61}$$

because all leaves in $\mathcal{V}_i^+$ have their first $\ell(v_i^+)-1$ elements in common. For convenience, let

$$p_i^+ = p_V(v_i^+)/q_X((0_M.v_i^+)_{[\ell(v_i^+])})$$

and note that

$$\Pr\{V \in \mathcal{V}_i^+\} \le p_i^+ \le \Pr\{V\in\mathcal{V}_i^+\}/p_{\min}. \tag{62}$$

This inequality holds because $0_M.v_i^+$ comes from expanding $\overline{b}_{i,\,i+1}$ until and including the first digit it differs from b_{i+1}. Therefore, if we set the last digit of $0_M.v_i^+$, $(0_M.v_i^+)_{[\ell(v_i^+])}$, to zero to form a number q, then the interval $[q,\, q+M^{-\ell(v_i^+)+1})$ includes both $\overline{b}_{i,\,i+1}$ and b_{i+1}. Since

$$p_i^+ = \Pr\{0_M.V \in [q,\, q+M^{-\ell(v_i^+)+1})\}$$

we arrive at the left half of (62). The right half of (62) holds because $p_V(v_i^+) \le \Pr\{V \in \mathcal{V}_i^+\}$. The part of the entropy contributed by the leaves in $\mathcal{V}_i^+$ can be upper-bounded as

$$\begin{aligned} & \sum_{v\in\mathcal{V}_i^+} p_V(v) \log \frac{1}{p_V(v)}\\ & \quad = \sum_{l=1}^{\infty} \sum_{m=0}^{(b_{i+1})_{[l]}-1} p_V(v_i^{l,\,m}) \log \frac{1}{p_V(v_i^{l,\,m})} && (63)\\ & \quad = \sum_{l=\ell(v_i^+)}^{\infty} \sum_{m=0}^{(b_{i+1})_{[l]}-1} p_V(v_i^{l,\,m})\\ & \qquad \cdot \left(\log\frac{1}{p_i^+} + \log\frac{p_i^+}{p_V(v_i^{l,\,m})}\right) && (64)\end{aligned}$$

$$\begin{aligned}
&= \sum_{l=\ell(v_i^+)}^{\infty} \sum_{m=0}^{(b_{i+1})_{[l]}-1} p_V(v_i^{l,m}) \log \frac{1}{p_i^+} \\
&\quad + p_i^+ \sum_{l=\ell(v_i^+)}^{\infty} \sum_{m=0}^{(b_{i+1})_{[l]}-1} \prod_{k=\ell(v_i^+)}^{l} \\
&\quad \cdot q_X((0_M.v_i^{l,m})_{[k]}) \sum_{k'=\ell(v_i^+)}^{l} \log \frac{1}{q_X((0_M.v_i^{l,m})_{[k']})} && (65)\\
&\le \Pr\{V \in \mathcal{V}_i^+\} \log \frac{1}{p_i^+} \\
&\quad + p_i^+ M \sum_{l=1}^{\infty} p_{\max}^l \left(l \log \frac{1}{p_{\min}} \right) && (66)\\
&\le \Pr\{V \in \mathcal{V}_i^+\} \left(\log \frac{1}{\Pr\{V \in \mathcal{V}_i^+\}} \right) + p_i^+ K_1 && (67)\\
&\le \Pr\{V \in \mathcal{V}_i^+\} \left(\log \frac{1}{\Pr\{V \in \mathcal{V}_i^+\}} \right) + 2^{-n} K_{p_1} && (68)
\end{aligned}$$

where K_1 and K_{p_1} are chosen appropriately. Equation (65) follows from (61), and inequalities (67) and (68) follow from (62). The part of the entropy contributed by the leaves in $\mathcal{V}_i^-$ can be similarly upper-bounded as

$$\sum_{v \in \mathcal{V}_i^-} p_V(v) \log \frac{1}{p_V(v)} \le \Pr\{V \in \mathcal{V}_i^-\} \left(\log \frac{1}{\Pr\{V \in \mathcal{V}_i^-\}} \right) + 2^{-n} K_{p_2}.$$

Summing the bounds on the contributions to the entropy from $\mathcal{V}_i^-$ and $\mathcal{V}_i^+$, we can upper-bound the entropy contributed by the leaves in $\mathcal{V}_i$ by the sum of the two

$$\begin{aligned}
\sum_{v \in \mathcal{V}_i} p_V(v) \log \frac{1}{p_V(v)} &\le \Pr\{V \in \mathcal{V}_i^-\} \log \frac{1}{\Pr\{V \in \mathcal{V}_i^-\}} \\
&\quad + \Pr\{V \in \mathcal{V}_i^+\} \log \frac{1}{\Pr\{V \in \mathcal{V}_i^+\}} \\
&\quad + 2^{-n}(K_{p_1} + K_{p_2}) && (69)\\
&= 2^{-n}(n+1) + 2^{-n}(K_{p_1} + K_{p_2}) && (70)
\end{aligned}$$

where (70) follows from the fact that

$$\Pr\{V \in \mathcal{V}_i^-\} = \frac{\Pr\{V \in \mathcal{V}_i^-\}}{\Pr\{V \in \mathcal{V}_i^-\} + \Pr\{V \in \mathcal{V}_i^+\}} \cdot 2^{-n} \quad (71)$$

$$\Pr\{V \in \mathcal{V}_i^+\} = \frac{\Pr\{V \in \mathcal{V}_i^+\}}{\Pr\{V \in \mathcal{V}_i^-\} + \Pr\{V \in \mathcal{V}_i^+\}} \cdot 2^{-n} \quad (72)$$

and that the entropy of a binary random variable is less than 1. Summing over all 2^n intervals, we have that $H(V) < n + C_\pi$ for an appropriate constant C_π that is independent of n. □

Appendix B
Proof of Subsystem Property 2

Subsystem Property 2 gives an upper bound on the expected length $E[L_{i+1}^\sigma]$ in terms of $E[L_i]$, which is obtained as follows.

Our approach is to first find an upper bound on $E[\ell(\Sigma_{i+1}) | L_i^\sigma = n]$ in terms of $E[L_i | L_i^\sigma = n]$. To find such a bound, we must know the distribution of the precoder's output ε_i' (the source-coder's input) conditioned $L_i^\sigma = n$. To find this distribution, we require the distribution of the precoder's input Ψ_i conditioned on $L_i^\sigma = n$. Fortunately, because of r in (12), we can assume that Ψ_i is, conditioned on $L_i^\sigma = n$, uniformly distributed over the set $\{0, 1\}^n$. Given such an input, we asserted in Section III-A1) that the precoder π_n produces output that is a stopped sequence of random variables that are i.i.d. according to q_X. Using this characterization of the precoder π_n's output, we model the source coding of this precoder's output as follows. Let $\{\hat{X}_k\}$ and $\{\hat{Y}_k\}$ be as defined in Section III-A2). Let $\hat{L}_n = l_n(0_M.\hat{X}^\infty)$, where l_n is defined as in (23). The precoder π_n's output is then represented by the random variable-length tuple $\hat{X}^{\hat{L}_n}$. The transmitter sends $\hat{X}^{\hat{L}_n}$ over the channel. The receiver feeds back $\hat{Y}^{\hat{L}_n}$, and the transmitter source codes $\hat{X}^{\hat{L}_n}$ according to

$$\prod_{j=1}^{\hat{L}_n} p_{X|Y}(\cdot | \hat{Y}_j)$$

which results in a stream of bits of length $\hat{L}_{\mathrm{out},n}$. Because

$$\left\lceil -\log \prod_{j=1}^{l} p_{X|Y}(\hat{X}_j | \hat{Y}_j) \right\rceil + 1$$

bits are used to represent $\hat{X}^{\hat{L}_n}$, we can write the expected value of $\hat{L}_{\mathrm{out},n}$ as

$$\begin{aligned}
E[\hat{L}_{\mathrm{out},n}] \le 2 + \sum_{l=1}^{\infty} p_{\hat{L}_n}(l) \sum_{\hat{y}^l \in \mathcal{Y}^l} \sum_{\hat{x}^l \in \mathcal{X}^l} p_{\hat{Y}^l | \hat{X}^l}(\hat{y}^l | \hat{x}^l) \\
\cdot p_{\hat{X}^l | \hat{L}_n}(\hat{x}^l | l) \log \frac{p_{\hat{Y}^l}(\hat{y}^l)}{p_{\hat{Y}^l | \hat{X}^l}(\hat{y}^l | \hat{x}^l) p_{\hat{X}^l}(\hat{x}^l)}. \quad (73)
\end{aligned}$$

We now prove the following lemma that bounds the length of $E[\hat{L}_{\mathrm{out},n}]$.

Lemma B.1:

$$E[\hat{L}_{\mathrm{out},n}] \le E[\hat{L}_n] H(X|Y) + H(\hat{L}_n) + 2. \quad (74)$$

Proof: We begin by expanding the logarithm in (73) as

$$\begin{aligned}
\log \frac{p_{\hat{Y}^l}(\hat{y}^l)}{p_{\hat{Y}^l | \hat{X}^l}(\hat{y}^l | \hat{x}^l) p_{\hat{X}^l}(\hat{x}^l)} &= \log \frac{p_{\hat{Y}^l | \hat{L}_n}(\hat{y}^l | l)}{p_{\hat{Y}^l | \hat{X}^l}(\hat{y}^l | \hat{x}^l) p_{\hat{X}^l | \hat{L}_n}(\hat{x}^l | l)} \\
&\quad + \log \frac{p_{\hat{X}^l | \hat{L}_n}(\hat{x}^l | l)}{p_{\hat{X}^l}(\hat{x}^l)} \\
&\quad - \log \frac{p_{\hat{Y}^l | \hat{L}_n}(\hat{y}^l | l)}{p_{\hat{Y}^l}(\hat{y}^l)}
\end{aligned}$$

we then obtain

$$\begin{aligned}
E[\hat{L}_{\mathrm{out},n}] &= 2 + \sum_{l=1}^{\infty} p_{\hat{L}_n}(l) [H(\hat{X}^l | \hat{Y}^l, \hat{L}_n = l) \\
&\quad + D(p_{\hat{X}^l | \hat{L}_n = l} \| p_{\hat{X}^l}) - D(p_{\hat{Y}^l | \hat{L}_n = l} \| p_{\hat{Y}^l}(\hat{y}^l))] && (75)\\
&\le 2 + \sum_{l=1}^{\infty} p_{\hat{L}_n}(l) [H(\hat{X}^l | \hat{Y}^l, \hat{L}_n = l) \\
&\quad + D(p_{\hat{X}^l | \hat{L}_n = l} \| p_{\hat{X}^l})]. && (76)
\end{aligned}$$

Since $\hat{L}_n$ is a deterministic stopping rule for $\{\hat{X}_j\}$, and because $\{\hat{X}_j\}$ is an i.i.d. process and the channel is memoryless, we have that for all integers $m > 0$

$$p_{\hat{X}_{l+1}^{l+m}|\hat{L}_n=l,\,\hat{X}^l} = p_{\hat{X}_{l+1}^{l+m}} \tag{77a}$$

$$p_{\hat{Y}_{l+1}^{l+m}|\hat{L}_n=l,\,\hat{Y}^l} = p_{\hat{Y}_{l+1}^{l+m}} \tag{77b}$$

$$p_{\hat{X}_{l+1}^{l+m},\,\hat{Y}_{l+1}^{l+m}|\hat{L}_n=l,\,\hat{X}^l,\hat{Y}^l} = p_{\hat{X}_{l+1}^{l+m},\,\hat{Y}_{l+1}^{l+m}}. \tag{77c}$$

From (77), it follows that for all integers $m > 0$

$$\sum_{l=1}^{m} p_{\hat{L}_n}(l) D(p_{\hat{X}^l|\hat{L}_n=l} \| p_{\hat{X}^l})$$

$$= \sum_{l=1}^{m} p_{\hat{L}_n}(l) D(p_{\hat{X}^m|\hat{L}_n=l} \| p_{\hat{X}^m}) \tag{78}$$

$$\le I(\hat{X}^m; \hat{L}_n) \tag{79}$$

$$\le H(\hat{L}_n), \tag{80}$$

and

$$\sum_{l=1}^{m} p_{\hat{L}_n}(l) H(\hat{X}^l|\hat{Y}^l, \hat{L}_n = l)$$

$$= \sum_{l=1}^{m} \sum_{\hat{x}^l} \sum_{\hat{y}^l} p_{\hat{X}^l, \hat{Y}^l, \hat{L}_n}(\hat{x}^l, \hat{y}^l, l) \cdot \log \frac{p_{\hat{Y}^l|\hat{L}_n}(\hat{y}^l|l)}{p_{\hat{X}^l, \hat{Y}^l|\hat{L}_n}(\hat{x}^l, \hat{y}^l|l)} \tag{81}$$

$$= \sum_{l=1}^{m} \sum_{\hat{x}^m} \sum_{\hat{y}^m} p_{\hat{X}^m, \hat{Y}^m, \hat{L}_n}(\hat{x}^m, \hat{y}^m, l) \cdot \log \frac{p_{\hat{Y}^m|\hat{L}_n}(\hat{y}^m|l)}{p_{\hat{X}^m, \hat{Y}^m|\hat{L}_n}(\hat{x}^m, \hat{y}^m|l)} - \sum_{l=1}^{m} p_{\hat{L}_n}(l) \sum_{\hat{x}_{l+1}^m} \sum_{\hat{y}_{l+1}^m} p_{\hat{X}_{l+1}^m, \hat{Y}_{l+1}^m}(\hat{x}_{l+1}^m, \hat{y}_{l+1}^m) \cdot \log \frac{p_{\hat{Y}_{l+1}^m}(\hat{y}_{l+1}^m)}{p_{\hat{X}_{l+1}^m, \hat{Y}_{l+1}^m}(\hat{x}_{l+1}^m, \hat{y}_{l+1}^m)} \tag{82}$$

$$\le H(\hat{X}^m|\hat{Y}^m, \hat{L}_n) - \left(\sum_{l=1}^{m} p_{\hat{L}_n}(l)(m-l)\right) H(\hat{X}_1|\hat{Y}_1) \tag{83}$$

$$\le mH(\hat{X}_1|\hat{Y}_1) - \left(\sum_{l=1}^{m} p_{\hat{L}_n}(l)(m-l)\right) H(\hat{X}_1|\hat{Y}_1) \tag{84}$$

$$\le H(\hat{X}_1|\hat{Y}_1)\left(\sum_{l=1}^{m} p_{\hat{L}_n}(l) l + m \Pr\{\hat{L}_n > m\}\right) \tag{85}$$

$$\le H(\hat{X}_1|\hat{Y}_1) E[\hat{L}_n]. \tag{86}$$

Inequality (84) follows from the fact that conditioning reduces entropy and that entropy is positive; (86) follows from writing out the sum corresponding to $E[\hat{L}_n]$ and using basic algebra. With (80) and (86) holding for all integers $m > 0$, substituting the appropriate quantities into (76) and taking the limit as $m \to \infty$ allows us to upper-bound $E[\hat{L}_{\mathrm{out},n}]$ according to

$$E[\hat{L}_{\mathrm{out},n}] \le 2 + H(\hat{L}_n) + E[\hat{L}_n] H(\hat{X}_1|\hat{Y}_1) \tag{87}$$

which completes the proof of the lemma. ▽

Lemma B.1 immediately implies that

$$E[\ell(\Sigma_{i+1})|L_i^\sigma] \le 2 + H(L_i|L_i^\sigma) + E[L_i|L_i^\sigma] H(X|Y) \tag{88}$$

which implies, after averaging both sides over L_i^σ, that

$$E[\ell(\Sigma_{i+1})] \le 2 + H(L_i) + E[L_i] H(X|Y) \tag{89}$$

$$\le 4 + \log E[L_i] + E[L_i] H(X|Y) \tag{90}$$

where (90) follows because $H(L) < 2 + \log E[L]$ for any positive, integer-valued random variable L (see, for example, [21, Corollary 3.12] for a proof).

We may conclude from (90) that for $i = 0, 1, \cdots, B_N - 1$

$$E[L_{i+1}^\sigma] \le E[L_i] H(X|Y) + \log E[L_i] + 4 + 4 + 2E[\lceil \log L_i \rceil] + 2E[\lceil \log L_i^\sigma \rceil] \tag{91}$$

where the last three terms are due to the encodings of L_i and L_i^σ in Ψ_i. To express the right-hand side completely in terms of L_i, we first prove the following lemma.

Lemma B.2: For all $d^n \in \{0, 1\}^n$

$$\ell(\pi_n(d^n)) > n/M_\pi \tag{92}$$

where $M_\pi = -\log p_{\min}$ and

$$p_{\min} = \min\{q_X(x) : q_X(x) \ne 0, x \in \mathcal{X}\}.$$

Proof: The probability of the sequence $\pi_n(d^n)$ must be less than 2^{-n}. The shortest such sequence would consist entirely of the element of $\mathcal{X}$ with lowest nonzero probability, giving the sequence probability $p_{\min}^{\ell(\pi_n(d^n))}$. The lemma then follows. ▽

This lemma implies that $L_i^\sigma < L_i M_\pi$, which implies that we can write

$$E[L_{i+1}^\sigma] \le E[L_i] H(X|Y) + 5 \log E[L_i] + 12 + \log M_\pi. \tag{93}$$

That the logarithm is concave and monotonically increasing completes the proof that Subsystem Property 2 holds. □

Appendix C
Proof of Subsystem Property 3

To bound the probability of decoding error for the mSB coding subsystem, we observe that an mSB decoding error occurs only if the transmitter sends w^ν, and the decoder mistakes it for c^ν. Let P_{cw} be the probability of this event, and let P_{wc} be the probability of the reverse error (mistaking c^ν for w^ν). If c^ν and w^ν are distinguished with the detector described in [4], which makes P_{cw} very small at the expense of making P_{wc} large (but sufficiently small), then it is shown in [4] that the choice of c^ν and w^ν given in Section III-A3) yields the bound [4] that

$$P_{cw} < \exp_2\{-\nu(E_{cw} - o_\nu(1))\} \tag{94}$$

where E_{cw} is defined as in (36).

The average length of the mSB encoder's output η can be upper- and lower-bounded, respectively, according to

$$\kappa^2 + \nu < \eta < (\kappa^2 + \nu)/(1 - P_{\mathrm{in}} - P_{wc}) \tag{95}$$

It can easily be shown that $P_{wc} = o_\nu(1)$ and $P_{\mathrm{in}} = o_\kappa(1)$.

If we set $\nu = \lceil \alpha N \rceil$ and $\kappa = \lceil N^{1/4} \rceil$, then (95) implies that $\eta = \alpha N + o_N(N)$, which can be used with (94) to arrive at the inequality

$$P_{cw} < \exp_2\{-\eta(E_{cw} - o_N(1))\}. \tag{96}$$

The probability of error $P_{e,\,\text{term},\,N}$ for this mSB scheme can be written

$$P_{e,\,\text{term},\,N} = \frac{P_{\text{in}}P_{cw}}{P_{\text{in}}P_{cw} + (1 - P_{\text{in}})(1 - P_{wc})} \tag{97}$$

which is less than P_{cw} for sufficiently large values of N, proving that the property holds.

As a final remark, note that we need not have chosen the "inner" code to be a bit-by-bit repetition code, but could have used a more complex code. The repetition code happens to be have very low complexity and is simple to construct. □

Appendix D
Proof of Lemma 4.1

To bound μ_{II}, we first use Subsystem Property 1 to conclude that

$$E[L_i|L_i^\sigma] \le (L_i^\sigma + C_\pi)/H(X). \tag{98}$$

Taking expectations of both sides of this bound and combining with (38) in Subsystem Property 2, we find that

$$\begin{aligned} E[L_{i+1}] &\le FE[L_i] + \lambda(E[L_i])/H(X) + C_\pi/H(X) && (99)\\ &= FE[L_i] + \tilde{\lambda}(E[L_i]) && (100)\end{aligned}$$

where $F = H(X|Y)/H(X)$, and $\tilde{\lambda}(x) = (\lambda(x) + C_\pi)/H(X)$. Note that $\tilde{\lambda}(x)/x \to 0$ as $x \to \infty$ and that $\tilde{\lambda}$ is a nonnegative, monotonically increasing, concave function over $[1, \infty)$.

Using the recursion (100), we obtain

$$E[L_i] \le F^i E[L_0] + \sum_{k=0}^{i-1} F^k \tilde{\lambda}(E[L_{i-1-k}]). \tag{101}$$

Using that $F > 0$, that $\tilde{\lambda}(x) > 0$ for all $x \ge 1$, that $E[L_k] \ge 1$ for $k = 0, \cdots, B_N$, and that

$$E[L_0] < (N + C_\pi)/H(X), \tag{102}$$

we can see that

$$\begin{aligned}\mu_{\text{II}} &= \sum_{i=0}^{B_N-1} E[L_i]\\ &\le \frac{1}{1-F}\left(\frac{N + C_\pi}{H(X)} + \sum_{i=0}^{B_N-1} \tilde{\lambda}(E[L_i])\right). && (103)\end{aligned}$$

Using that $\tilde{\lambda}$ is concave over $[1, \infty)$ with the fact that $E[L_k] \ge 1$ for $k = 0, \cdots, B_N$, we can bound the second term in (103) according to

$$\begin{aligned}\sum_{i=0}^{B_N-1} \tilde{\lambda}(E[L_i]) &\le B_N \tilde{\lambda}\left(\sum_{i=0}^{B_N-1} E[L_i]/B_N\right) && (104)\\ &= B_N\tilde{\lambda}(\mu_{\text{II}}/B_N). && (105)\end{aligned}$$

Using (103) and (105), we can write

$$\frac{\mu_{\text{II}}}{N} \le \frac{1}{N}\,\frac{1}{1-F}\left(\frac{N + C_\pi}{H(X)} + B_N\tilde{\lambda}(\mu_{\text{II}}/B_N)\right) \tag{106}$$

which, using elementary algebra, implies that

$$\begin{aligned}\frac{\mu_{\text{II}}}{N} \le &\left(\frac{1}{H(X) - H(X|Y)}\right)\left(\frac{N + C_\pi}{N}\right)\\ &\cdot\left(\frac{1-F}{1 - F - B_N\tilde{\lambda}(\mu_{\text{II}}/B_N)/\mu_{\text{II}}}\right). && (107)\end{aligned}$$

Since $\lim_{x\to\infty} \tilde{\lambda}(x)/x = 0$, and $\mu_{\text{II}}/B_N \to \infty$ as $N \to \infty$, the lemma follows. □

Appendix E
Proof of Lemma 4.2

To bound μ_{I}, we first use (92) with (29) to obtain

$$\tilde{L} \le 2\ell(\Sigma_{B_N}) + 2\log t_N + 4\log L_{B_N-1} + 2\log M_\pi + 16. \tag{108}$$

We next bound L_{B_N-1} as follows: since $\tilde{\lambda}$ from (100) is $o_N(N)$ and also represents a quantity that is finite, it follows that there is some constant C such that

$$\tilde{\lambda}(x) \le \frac{1-F}{2}x + C \tag{109}$$

so that the bound

$$E[L_{i+1}] \le \frac{1+F}{2}E[L_i] + C \tag{110}$$

holds. Using (102) with this recursion, we see that

$$E[L_i] \le \left(\frac{1+F}{2}\right)^i \frac{N}{H(X)} + C' \tag{111}$$

where C' is a constant. With (18), it follows immediately that

$$E[L_{B_N-1}] \le C' + o_N(1). \tag{112}$$

This inequality in turn implies via (110) that $E[\ell(\Sigma_{B_N})] = O_N(1)$. Assuming $t_N \sim o_N(N)$, which is ensured by Subsystem Property 4, we may conclude that $E[\tilde{L}] = O_N(\log N)$.

Since $\kappa = \Theta_N(N^{1/4})$, it follows from (32) that

$$E[B_G] = 1 + o_N(1). \tag{113}$$

Coupled with the fact that $P_{e,\,\text{term},\,N} = o_N(1)$, the lemma follows. □

Appendix F
Proof of Subsystem Property 4

Here, we prove that both false-alarm and missed-detection probability decay at least exponentially with t_N.

To determine the probability of false alarm, suppose that $Y_1, Y_2, \cdots, Y_{t_N}$ are i.i.d. with marginal pmf p_Y. With $\emptyset^{t_N} = (a, \cdots, a)$, a being defined by (51), we can upper-bound the probability of false alarm according to

$$\begin{aligned}&\Pr\{\delta_{t_N}(Y^{t_N}) = 1\}\\ &\quad\le \sum_{y^{t_N}\in\mathcal{Y}^{t_N}} p_{Y^{t_N}}(y^{t_N})\left(\frac{p_{Y^{t_N}|X^{t_N}}(y^{t_N}|\emptyset^{t_N}[N])}{p_{Y^{t_N}}(y^{t_N})}\right)^{1/2} && (114)\\ &\quad\le \sum_{y^{t_N}\in\mathcal{Y}^{t_N}} \prod_{i=1}^{t_N} p_Y(y_i)^{1/2} p_{Y|X}(y_i|a)^{1/2} && (115)\end{aligned}$$

$$= \prod_{i=1}^{t_N} \left(\sum_{k=1}^{|\mathcal{Y}|} p_Y(k)^{1/2} p_{Y|X}(k|x_i)^{1/2} \right) \tag{116}$$

$$\triangleq (E_\emptyset)^{t_N}. \tag{117}$$

By the Schwartz inequality, $E_\emptyset \leq 1$, with equality if and only if X and Y are independent. The probability of missed detection can also be bounded above by $(E_\emptyset)^{t_N}$. With t_N defined by (53), Subsystem Property 4 follows. □

Appendix G
Modified Scheme Reliability

To find the reliability of the modified scheme, we must find the new expected length and probability of error associated with the new coding scheme. Let $\tilde{c}(N, \nu_N)$ denote the modified counterpart to $c_{\text{DMC}}(N, \nu_N)$ defined in Section IV.

We have the following lemma that characterizes the length for the $\tilde{c}(N, \nu_N)$ in terms of that for the length L^* under the perfect-detection assumption:

Lemma G.1: With L_+^* denoting the length associated with the $\tilde{c}(N, \nu_N)$

$$E[L_+^*] = E[L^*] + o_N(N). \tag{118}$$

Proof: The increase in length $L_+^* - L^*$ has two sources: false alarms and missed detections of the synchronization sequence.

The number of additional transmissions Δ_1^* due to false alarms satisfies

$$E[\Delta_1^*] \leq \left(\frac{E[A_{B_N}]}{t_N} + 1 \right) P_{\text{FA}, t_N} \left(\frac{N}{\kappa} \right) \eta_N(r)(1 + o_N(1)) \tag{119}$$

$$= o_N(1) \tag{120}$$

where P_{FA, t_N} is the probability of false alarm associated with detecting $\emptyset^{t_N}[N]$. (Also recall that $A_{B_N} + 1$ is the time at which transmission of the sequence corresponding to the B_Nth iteration of the coding algorithm, and thus indicates the time at which the symbol $\emptyset^{t_N}[N]$ is transmitted.) The reasoning behind (119) is as follows: there are about A_{B_N}/t_N opportunities for false alarms in a particular transmission. If a false alarm occurs, the N source bits to be sent via the termination coder, where each block of κ_N bits uses about $\eta_N(r)$ channel inputs. Equation (120) follows by using Subsystem Property 4 with the fact that $E[A_{B_N}]$ grows linearly with N (Lemma 4.1).

Next, the number of additional transmissions Δ_2^* due to missed detections satisfies

$$E[\Delta_2^*] \leq \frac{t_N P_{\text{MD}, t_N}}{1 - P_{\text{MD}, t_N}} \tag{121}$$

$$< \frac{M_{\text{MD}} t_N}{1 - M_{\text{MD}}} \tag{122}$$

which is derived as follows. Suppose we send the stream $\emptyset^{t_N}[N], \emptyset^{t_N}[N], \cdots$, and the detector does its hypothesis test every t_N samples. Then the number of times that $\emptyset^{t_N}[N]$ is transmitted before the first detection is a random variable with mean less than $1/(1 - P_{\text{MD}, t_N})$. Because we counted the first transmission of $\emptyset^{t_N}[N]$ in μ_{II} of (44), the number of *additional* transmissions is bounded above by (121). Equation (122) follows from exploiting (55). Note that the additional transmissions due to transmission of $\phi(C_{\text{MD}})$ is on average a constant and therefore negligible, as κ is large enough that $E[B_G]$ remains as in (113).

Combining the sources of additional channel uses, we obtain (118) as desired. □

The probability of error associated with $\tilde{c}(N, \nu_N)$ is characterized in terms of the probability of error associated with the scheme under the perfect-detection assumption by the following lemma:

Lemma G.2: With $P_{e, N}$ and $P_{e, +, N}$ denoting the probabilities of error associated with $c_{\text{DMC}}(N, \nu_N)$ and $\tilde{c}(N, \nu_N)$, respectively,

$$P_{e, +, N} < P_{e, N} + P_{B, N} \tag{123}$$

where

$$P_{B, N} \leq \left(\frac{E[A_{B_N}]}{t_N} + 1 \right) P_{\text{FA}, t_N} \left(\frac{N}{\kappa} \right) P_{e, \text{term}, N} \tag{124}$$

$$= o_N(1) \, P_{e, \text{term}, N}. \tag{125}$$

Proof: The modified scheme introduces only one additional error event: the event that a false alarm occurs and then one of the subsequent N/κ termination-coded blocks is received in error. Since the existence of a false alarm does not affect the probability of error associated with the termination-coded blocks, we arrive at (124) via the union bound. □

Since $P_{e, N} = (1 + o_N(1)) P_{e, \text{term}, N}$, Lemma G.2 says that the probability of error is effectively unchanged. Since Lemma G.1 says that the expected length is effectively unchanged as well, Lemmas G.1 and G.2 together imply that the modified scheme attains the error exponent function E_{CEC} defined in Theorem 4.1.

Appendix H
Precoding with Linear Complexity

Without loss of generality, assume that $\mathcal{X} = \{0, 1, \cdots, M-1\}$. Given a sequence $d^n \in \{0, 1\}^n$ to precode with π_n, let $s = 0_2.\, d^n + 2^{-n} Z$, where Z is a random variable uniformly distributed over $[0, 1)$. Then let $s_b = 0_2.\, s^{[n]}$, and let $s_t = s_b + 2^{-n}$. The precoder then finds the longest interval I of the form $I = [0_M.\, u_1 \cdots u_j, 0_M.\, u_1 \cdots u_j + M^{-j})$ such that $F_{\tilde{X}}(I) \subseteq (s_b, s_t)$ and then puts out u^j. The following algorithm gives a method for doing these steps efficiently.

1) $S_b := 0, S_t := 1, R := 1, l := 1$.
2) If $S_b > s_b$ and $S_t \leq s_t$ then go to 4). Otherwise, go to 3).
3) Compute

   $$b_{l, m} := b_{l,m-1} + R q_X(m-1), \text{ for } m = 1, 2, \cdots, M$$

   starting with $b_{l, 0} = S_b$. For $k = 0, 1, \cdots, M-1$, if $s \in [b_{l, k}, b_{l, k+1})$, then set $U_l := k$, $R := R q_X(k)$, $S_b := b_{l, k}$, $S_t := b_{l, k+1}$, $l := l + 1$, break out of the loop over k, and go to 2).
4) Halt, and return $(U_1, U_2, \cdots, U_{l-1})$.

Assuming real arithmetic requires a single computation, this algorithm has complexity that is linear in the final value of $l-1$, which equals $\ell(\pi_n(s))$.

To recover d^n, we need only recover the first n binary-expansion digits of s from $\pi_n(d^n)$. To do so, we compute $x = F_{\tilde{X}}(0_M.\pi_n(d^n))$. Then s and x share their first n binary-expansion digits. A convenient formula for $F_{\tilde{X}}$ is

$$F_{\tilde{X}}(y) = \sum_{k=1}^{\infty} \sum_{m=0}^{y_{[k]}-1} q_X(m) \prod_{i=1}^{k-1} q_X(y_{[i]}). \tag{126}$$

Because the M-ary-expansion digits of $0_M.\pi_n(d^n)$ are, by definition, all zero after the first $\ell(\pi_n(d^n))$ digits, and because the product term in (126) can be computed recursively, $F_{\tilde{X}}(0_M.\pi_n(d^n))$ can be computed with complexity that is linear in $\ell(\pi_n(d^n))$.

Like arithmetic source encoders and decoders, these methods for precoding and inversion of precoding suffer from numerical precision problems on finite-precision computers. To avoid these problems, we must use systems for arithmetic with arbitrary precision (see, for example, [22]), or we must use special rescaling methods similar to those that are used to carry out arithmetic source encoding and decoding (see, for example, [23]).

Acknowledgment

Helpful discussions with Prof. Amos Lapidoth led to the attainment of the optimal error exponent. The thoughtful suggestions of an anonymous reviewer also improved the paper.

References

[1] J. P. M. Schalkwijk and T. Kailath, "A coding scheme for additive noise channels with feedback—Part I: No bandwidth constraint," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 172–182, Apr. 1966.

[2] M. Horstein, "Sequential transmission using noiseless feedback," *IEEE Trans. Inform. Theory*, vol. IT-9, pp. 136–143, July 1963.

[3] E. R. Berlekamp, "Block coding with noiseless feedback," Ph.D. dissertation, Mass. Inst. Techno., Cambridge, MA, 1964.

[4] H. Yamamoto and K. Itoh, "Asymptotic performance of a modified Schalkwijk–Barron scheme for channels with noiseless feedback," *IEEE Trans. Inform. Theory*, vol. IT-25, pp. 729–733, Nov. 1979.

[5] M. V. Burnashev, "Data transmission over a discrete channel with feedback. Random transmission time," *Probl. Inform. Transm.*, vol. 12, no. 4, pp. 250–265, 1976.

[6] B. D. Kudryashov, "Message transmission over a discrete channel with noiseless feedback," *Probl. Inform. Transm.*, vol. 15, no. 1, pp. 1–9, 1979.

[7] N. T. Gaarder and J. K. Wolf, "The capacity region of a multiple-access discrete memoryless channel can increase with feedback," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 100–102, Jan. 1975.

[8] T. M. Cover and C. S. K. Leung, "An achievable rate region for the multiple-access channel with feedback," *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 292–298, May 1981.

[9] T. Veugen, "Multiple-repetition coding for channels with feedback," Ph.D. dissertation, Eindhoven Univ. Technol., Eindhoven, The Netherlands, 1997.

[10] R. Ahlswede, "A constructive proof of the coding theorem for discrete memoryless channels with feedback," in *Proc. 6th Prague Conf. Information Theory, Statistical Decision Functions, and Random Processes*, 1971, pp. 39–50.

[11] J. M. Ooi, "A framework for low-complexity communication over channels with feedback," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, MA, 1997.

[12] A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms.* Reading, MA: Addison-Wesley, 1974.

[13] M. Sipser and D. Spielman, "Expander codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1710–1722, Nov. 1996.

[14] R. G. Gallager, *Discrete Stochastic Processes.* Boston, MA: Kluwer, 1996.

[15] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: Wiley, 1991.

[16] G. D. Forney, *Concatenated Codes.* Cambridge, MA: MIT Press, 1966.

[17] D. Spielman, "Linear-time encodable and decodable error-correcting codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1723–1731, Nov. 1996.

[18] A. Papoulis, *Probability, Random Variables, and Stochastic Processes.* New York: McGraw-Hill, 1991.

[19] J. M. Ooi and G. W. Wornell, "Fast iterative coding techniques for feedback channels: Finite-state channels and universal communication," *IEEE Trans. Inform. Theory*, submitted for publication.

[20] L. Ekroot and T. M. Cover, "The entropy of a randomly stopped sequence," *IEEE Trans. Inform. Theory*, vol. 37, pp. 1641–1644, Nov. 1991.

[21] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems.* New York: Academic, 1981.

[22] W. H. Press, B. P. Flannery, and S. A. Teukolsky, *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge, U.K.: Cambridge Univ. Press, 1992.

[23] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. Assoc. Comput. Mach.*, vol. 30, pp. 520–540, June 1987.