

Channels That Die

Lav R. Varshney*[†], Sanjoy K. Mitter* and Vivek K Goyal[†]

*Laboratory for Information and Decision Systems, Massachusetts Institute of Technology

[†]Research Laboratory of Electronics, Massachusetts Institute of Technology

Abstract—Given the possibility of communication systems failing catastrophically, we investigate limits to communicating over channels that fail at random times. These channels are finite-state semi-Markov channels. We show that communication with arbitrarily small probability of error is not possible. Making use of results in finite blocklength channel coding, we determine sequences of blocklengths that optimize transmission volume communicated at fixed maximum message error probabilities. A dynamic programming formulation is used to show that channel state feedback does not improve performance.

“a communication channel... might be inoperative because of an amplifier failure, a broken or cut telephone wire, ...”

— I. M. Jacobs [1]

I. INTRODUCTION

Physical systems have a tendency to fail at random times [2]. This is true whether considering communication systems embedded in networks that may run out of energy [3], synthetic systems embedded in biological cells that may die [4], systems embedded in spacecraft that may enter black holes [5], or systems embedded in oceans with undersea cables that may be cut [6]. In these scenarios and beyond, communication system failure may be modeled as communication channel death.

It is of interest to study information-theoretic limits on communicating over channels that die at random times. This paper studies one such channel model and provides results on what is possible and what is impossible. Communication with arbitrarily small probability of error (*Shannon reliability*) is not possible over channels that die, however a suitably defined notion of η -reliability is possible. Limits on channel coding with finite blocklength [7], which have seen renewed interest [8]–[12], are central to our development. Indeed, channels that die bring the notion of finite blocklength to the fore and provide a concrete reason to step back from infinity.

The notion of outage in wireless communication [13] is similar to channel death, except that outage is not a permanent condition. Likewise for lost letters in postal channels [14]. Therefore blocklength asymptotics are useful to study those channel models but are not useful for channels that die. Recent work that has similar motivations as our study [15] provides the outage capacity of a wireless channel.

The central trade-off in communicating over channels that die is in the lengths of codeword blocks. Longer blocks improve communication performance as classically known,

whereas shorter blocks have a smaller probability of being prematurely terminated due to channel death. We develop a dynamic programming formulation to optimize the ordered integer partition that determines the sequence of blocklengths. Solving the dynamic program demonstrates that channel state feedback does not improve performance.

The optimization of codeword blocklengths is reminiscent of frame size control in wireless networks [16], however such techniques are used in conjunction with automatic repeat request protocols and are motivated by amortizing protocol information.

The remainder of the paper is organized as follows. Section II defines discrete channels that die and shows that these channels have zero Shannon capacity. Section III states the communication system model and also fixes novel performance criteria. Section IV shows that our notion of Shannon reliability is not achievable, strengthening the result of zero Shannon capacity. Section V provides the optimal communication scheme and determines its performance. Section VI optimizes performance for geometric death distributions whereas Section VII optimizes performance for finite-support death distributions. Section VIII provides an example and Section IX suggests several extensions to this work.

II. CHANNEL MODEL

Consider a channel with finite input alphabet \mathcal{X} and finite output alphabet \mathcal{Y} . It has an *alive* state $s = a$ when it acts like a discrete memoryless channel (DMC) and a *dead* state $s = d$ when it almost surely erases the input, producing symbol $?$.¹

Assume throughout the paper that the DMC from the alive state has zero-error capacity equal to zero.

For example, if the channel acts like a binary symmetric channel (BSC) with crossover probability $0 < \varepsilon < 1$ in the alive state, with $\mathcal{X} = \{0, 1\}$, and $\mathcal{Y} = \{0, 1, ?\}$, then the transmission matrix in the alive state is

$$p(y|x, s = a) = p_a(y|x) = \begin{bmatrix} 1 - \varepsilon & \varepsilon & 0 \\ \varepsilon & 1 - \varepsilon & 0 \end{bmatrix}, \quad (1)$$

and the transmission matrix in the dead state is

$$p(y|x, s = d) = p_d(y|x) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

The channel starts in state $s = a$ and then transitions to $s = d$ at some random time T , where it remains for all time thereafter. That is, the channel is in state a for times $n =$

¹Our results can be extended to cases where the channel acts like other channels in the alive state.

1, 2, ..., T and in state d for times $n = T + 1, T + 2, \dots$. The death time distribution is denoted $p_T(t)$.

A. Finite-State Semi-Markov Channel

We can show that a channel that dies is a finite-state channel (FSC) [17, Sec. 4.6].

Proposition 1: A channel that dies $(\mathcal{X}, p_a(y|x), p_d(y|x), p_T(t), \mathcal{Y})$ is a finite-state channel.

Proof: Follows by definition. ■

Although rather trivially so, the channel state sequence is a semi-Markov process [18, Sec. 4.8], [19, Sec. 5.7] with fixed initial state and is not irreducible.

Definition 1: The sequence of states of a semi-Markov process is a Markov chain; this is called the *embedded Markov chain* of the semi-Markov process.

Definition 2: A semi-Markov process is *irreducible* if its embedded Markov chain is irreducible.

Proposition 2: The channel state sequence for a channel that dies $(\mathcal{X}, p_a(y|x), p_d(y|x), p_T(t), \mathcal{Y})$ is a non-irreducible semi-Markov process.

Proof: When in state a , the next state is d with probability 1 and given that the next state is to be d , the time until the transition from a to d has distribution $p_T(t)$. When in state d , the next state is d with probability 1. Thus, the channel state sequence is a semi-Markov process.

The semi-Markov state process is not irreducible because the a state of the embedded Markov chain is transient. ■

Note that when T is a geometric random variable, the channel state process forms a Markov chain, with transient state a and recurrent, absorbing state d .

We can define further special classes of FSCs.

Definition 3: An FSC is a *finite-state semi-Markov channel* (FSSMC) if its state sequence forms a semi-Markov process.

Definition 4: An FSC is a *finite-state Markov channel* (FSMC) if its state sequence forms a Markov chain.

Proposition 3: A channel that dies $(\mathcal{X}, p_a(y|x), p_d(y|x), p_T(t), \mathcal{Y})$ is an FSSMC and is an FSMC when T is geometric.

Proof: Follows from Props. 1 and 2. ■

FSMCs in general [17], [20], and in particular the child's toy/panic button channel of Gallager [20, p. 26], [17, p. 103] and the Gilbert–Elliott channel and its extensions [21], have been widely studied in the literature. Contrarily, FSSMCs seem to not have been specifically studied in information theory. There are a couple works [22], [23] that give semi-Markov channel models for wireless communications systems but do not provide information-theoretic characterizations.

B. Capacity is Zero

Our first result on transmission over a channel that dies is that the Shannon capacity [17] is zero.

First we show that if the initial state of a channel that dies were not fixed, then it would be an indecomposable FSC [17, Sec. 4.6], where the effect of the initial state dies away.

Proposition 4: If the initial state of a channel that dies $(\mathcal{X}, p_a(y|x), p_d(y|x), p_T(t), \mathcal{Y})$ is not fixed, then it is an indecomposable FSC.

Proof: The embedded Markov chain for a channel that dies has a unique absorbing state d . ■

Indecomposable FSCs have the property that the upper capacity, defined in [17, (4.6.6)], and lower capacity, defined in [17, (4.6.3)], are identical [17, Thm. 4.6.4]. This can be used to show that the capacity of a channel that dies is zero.

Proposition 5: The Shannon capacity, C , of a channel that dies $(\mathcal{X}, p_a(y|x), p_d(y|x), p_T(t), \mathcal{Y})$ is zero.

Proof: Although the initial state $s_1 = a$ in our problem, temporarily suppose that s_1 may be either a or d . Then our channel is indecomposable by Prop. 4.

The lower capacity \underline{C} equals the upper capacity \overline{C} , for indecomposable channels by [17, Thm. 4.6.4]. The information rate of a memoryless $p_d(y|x)$ ‘dead’ channel is clearly zero for any input distribution, so the lower capacity $\underline{C} = 0$. Thus the Shannon capacity for a channel that dies with initial alive state is $C = \overline{C} = 0$. ■

III. COMMUNICATION SYSTEM

In order to information theoretically characterize a channel that dies, we propose a system in which communication is carried out over the channel.

We have an information stream (like i.i.d. equiprobable bits), which can be grouped into a sequence of k messages, (W_1, W_2, \dots, W_k) . Each message W_i is drawn from a message set $\mathcal{W}_i = \{1, 2, \dots, M_i\}$. Each message W_i is encoded into a channel input codeword $X_1^{n_i}(W_i)$ and these codewords $(X_1^{n_1}(W_1), X_1^{n_2}(W_2), \dots, X_1^{n_k}(W_k))$ are transmitted in sequence over the channel. A noisy version of this codeword sequence is received, $Y_1^{n_1+n_2+\dots+n_k}(W_1, W_2, \dots, W_k)$. The receiver then guesses the sequence of messages using an appropriate decoding rule g , to produce $(\hat{W}_1, \hat{W}_2, \dots, \hat{W}_k) = g(Y_1^{n_1+n_2+\dots+n_k})$. The \hat{W}_i s are drawn from alphabets $\mathcal{W}_i^\ominus = \mathcal{W}_i \cup \ominus$, where the \ominus message indicates the decoder declaring an erasure. The receiver makes an error on message i if $\hat{W}_i \neq W_i$ and $\hat{W}_i \neq \ominus$.

Block coding results are typically expressed with the concern of sending one message rather than k messages, as here.

We can write definitions somewhat more formally.

Definition 5: An (M_i, n_i) *individual message code* for a channel that dies $(\mathcal{X}, p_a(y|x), p_d(y|x), p_T(t), \mathcal{Y})$ consists of:

- 1) An individual message index set $\{1, 2, \dots, M_i\}$, and
- 2) An individual message encoding function $f_i : \{1, 2, \dots, M_i\} \mapsto \mathcal{X}^{n_i}$.

The individual message index set $\{1, 2, \dots, M_i\}$ is denoted \mathcal{W}_i , and the set of individual message codewords $\{f_i(1), f_i(2), \dots, f_i(M_i)\}$ is called the *individual message codebook*.

Definition 6: An $(M_i, n_i)_{i=1}^k$ *code* for a channel that dies $(\mathcal{X}, p_a(y|x), p_d(y|x), p_T(t), \mathcal{Y})$ is a sequence of k individual message codes, $(M_i, n_i)_{i=1}^k$, in the sense of comprising:

- 1) A sequence of individual message index sets $\mathcal{W}_1, \dots, \mathcal{W}_k$,
- 2) A sequence of individual message encoding functions $f = (f_1, \dots, f_n)$, and

3) A decoding function $g : \mathcal{Y}^{\sum_{i=1}^k n_i} \mapsto \mathcal{W}_1^\ominus \times \cdots \times \mathcal{W}_k^\ominus$.

There is no essential loss of generality by assuming that the decoding function g is decomposed into a sequence of individual message decoding functions $g = (g_1, g_2, \dots, g_n)$ where $g_i : \mathcal{Y}^{n_i} \mapsto \mathcal{W}_i^\ominus$ when individual messages are chosen independently, due to this independence and the conditional memorylessness of the channel. To define performance measures, we assume that the decoder operates on an individual message basis. That is, when applying the communication system, let $\hat{W}_1 = g_1(Y_1^{n_1})$, $\hat{W}_2 = g_2(Y_{n_1+1}^{n_1+n_2})$, and so on.

For the sequel, we make a further assumption.

Assumption 1: If all n_i channel output symbols used by individual message decoder g_i are not ?, then the range of g_i is \mathcal{W}_i . If any of the n_i channel output symbols used by individual message decoder g_i are ?, then g_i maps to \ominus .

A. Performance Measures

We formally write the notion of error as follows.

Definition 7: For all $1 \leq w \leq M_i$, let

$$\lambda_i^w = \Pr[\hat{W}_i \neq w | W_i = w, \hat{W}_i \neq \ominus]$$

be the conditional message probability of error given that the i th individual message is w .

Definition 8: The maximal probability of error for an (M_i, n_i) individual message code is

$$\lambda_i^{\max} = \max_{w \in \mathcal{W}_i} \lambda_i^w.$$

Definition 9: The maximal probability of error for an $(M_i, n_i)_{i=1}^k$ code is

$$\lambda^{\max} = \max_{i \in \{1, \dots, k\}} \lambda_i^{\max}.$$

Since the Shannon capacity of a channel that dies is zero (Prop. 5), we propose weaker performance criteria. In particular, we define formal notions of how much information is transmitted using a code and how long it takes.

Definition 10: The transmission time of an $(M_i, n_i)_{i=1}^k$ code is $N = \sum_{i=1}^k n_i$.

Definition 11: The expected transmission volume of an $(M_i, n_i)_{i=1}^k$ code is

$$V = \mathbb{E}_T \left\{ \sum_{i \in \{1, \dots, k | \hat{W}_i \neq \ominus\}} \log_2 M_i \right\}.$$

The several performance criteria for a code may be combined together.

Definition 12: Given $0 \leq \eta < 1$, a pair of numbers (N_0, V_0) (where N_0 is a positive integer and V_0 is non-negative) is said to be an *achievable transmission time-volume at η -reliability* if there exists, for some k , an $(M_i, n_i)_{i=1}^k$ code for the channel that dies $(\mathcal{X}, p_a(y|x), p_d(y|x), p_T(t), \mathcal{Y})$ such that

$$\lambda^{\max} \leq \eta, N \leq N_0, \text{ and } V \geq V_0. \quad (3)$$

Moreover, (N_0, V_0) is said to be an *achievable transmission time-volume at Shannon reliability* if it is an achievable transmission time-volume at η -reliability for all $0 < \eta < 1$.

The goal of this work is to demarcate what is achievable.

IV. SHANNON RELIABILITY IS NOT ACHIEVABLE

Not only is the Shannon capacity of a channel that dies zero, but there is no $V > 0$ such that (N, V) is an achievable transmission time-volume at Shannon reliability. A coding scheme that always declares erasures would achieve zero error probability (and therefore Shannon reliability) but would not provide positive transmission volume; this is also not allowed under Assumption 1.

Proposition 6: For a channel that dies $(\mathcal{X}, p_a(y|x), p_d(y|x), p_T(t), \mathcal{Y})$, there is no $V > 0$ such that (N, V) is an achievable transmission time-volume at Shannon reliability.

The basic idea is to show that the easiest communication problem, transmitting a single individual message repetition code, cannot be done over a channel that dies with arbitrarily small probability of error. The proof is omitted due to space constraints. Notice that Prop. 6 also directly implies Prop. 5.

V. LIMITS ON COMMUNICATION

In this section, we develop a scheme that performs η -reliable communication over a channel that dies. It is based in part on finite blocklength channel coding, which we first review.

A. Finite Blocklength Channel Coding

Under our definitions, traditional channel coding results [7]–[12] provide information about individual message codes, determining the achievable trios $(n_i, M_i, \lambda_i^{\max})$. In particular, the largest possible $\log M_i$ for a given n_i and λ_i^{\max} is denoted $\log M^*(n_i, \lambda_i^{\max})$.

It is not our purpose to improve upper and lower bounds on finite blocklength channel coding, but rather to use existing results to study channels that die. In fact, for the sequel, we will simply assume that the function $\log M^*(n_i, \lambda_i^{\max})$ is known, as are codes/decoders that achieve this value. In principle, optimal individual message codes may be found through exhaustive search [8], [24]. Although algebraic notions of code quality do not directly imply error probability quality [25], perfect codes such as the Hamming or Golay codes may also be optimal in certain limited cases.

Recent results comparing upper and lower bounds around Strassen's normal approximation to $\log M^*(n_i, \lambda_i^{\max})$ [26] have demonstrated that the approximation is quite good [10].

Remark 1: We assume that optimal $\log M^*(n_i, \eta)$ -achieving individual message codes are known. Exact upper and lower bounds to $\log M^*(n_i, \eta)$ can be substituted to make our results precise. For numerical demonstrations, we will further assume that optimal codes have performance given by Strassen's approximation.

The following expression for $\log M^*(n_i, \eta)$ that first appeared in [26] is also given as [10, Thm. 6].

Lemma 1: Let $M^*(n_i, \eta)$ be the largest size of an individual message code with block length n_i and maximal error probability upper bounded by $\lambda_i^{\max} < \eta$. Then, for any DMC with capacity C and $0 < \eta \leq 1/2$,

$$\log M^*(n_i, \eta) = n_i C - \sqrt{n_i \rho} Q^{-1}(\eta) + O(\log n_i),$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-t^2/2} dt$$

and

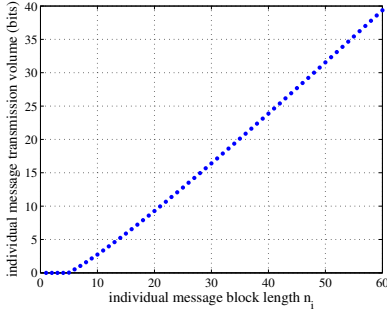
$$\rho = \min_{X:C=I(X;Y)} \text{var} \left[\log \frac{p_{Y|X}(y|x)}{p_Y(y)} \right].$$

For the BSC(ε), the approximation (ignoring the $O(\log n_i)$ term above) is:

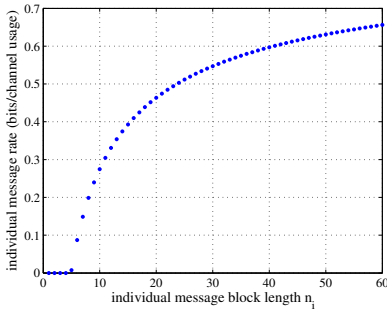
$$\log M^* \approx n_i(1-h_2(\varepsilon)) - \sqrt{n_i\varepsilon(1-\varepsilon)}Q^{-1}(\eta) \log_2 \frac{\varepsilon}{1-\varepsilon}, \quad (4)$$

where $h_2(\cdot)$ is the binary entropy function. This BSC expression first appeared in [27].

For intuition, we plot the approximate $\log M^*(n_i, \eta)$ function for a BSC(ε) in Fig. 1(a). Notice that $\log M^*$ is zero for small n_i since no code can achieve the target error probability η . Also notice that $\log M^*$ is a monotonically increasing function of n_i . Moreover, notice in Fig. 1(b) that even when normalized, $(\log M^*)/n_i$, is a monotonically increasing function of n_i . Therefore longer blocks provide more ‘bang for the buck.’ The curve in Fig. 1(b) asymptotically approaches capacity.



(a)



(b)

Fig. 1. (a). The expression (4) for $\varepsilon = 0.01$ and $\eta = 0.001$. (b). Normalized version, $(\log M^*(n_i, \eta))/n_i$, for $\varepsilon = 0.01$ and $\eta = 0.001$. The capacity of a BSC(ε) is $1 - h_2(\varepsilon) = 0.92$.

B. η -reliable Communication

We now describe a coding scheme that achieves positive expected transmission volume at η -reliability. Survival probability of the channel plays a key role in measuring performance.

Definition 13: The survival function of a channel that dies $(\mathcal{X}, p_a(y|x), p_d(y|x), p_T(t), \mathcal{Y})$ is $\Pr[T > t]$, is denoted $R_T(t)$, and satisfies

$$R_T(t) = \Pr[T > t] = 1 - \sum_{\tau=1}^t p_T(\tau) = 1 - F_T(t),$$

where F_T is the cumulative distribution function. $R_T(t)$ is a non-increasing function.

Proposition 7: The transmission time-volume

$$\left(N = \sum_{i=1}^k n_i, V = \sum_{i=1}^k R_T(e_i) \log M^*(n_i, \eta) \right)$$

is achievable at η -reliability for any sequence $(n_i)_{i=1}^k$ of individual message codeword lengths, and $e_0 = 0, e_1 = n_1, e_2 = n_1 + n_2, \dots, e_k = \sum_{i=1}^k n_i$.

Proof:

Code Design: A target error probability η and a sequence $(n_i)_{i=1}^k$ of individual message codeword lengths are fixed. Construct a length- k sequence of (M_i, n_i) individual message codes and individual decoding functions $(\mathcal{W}_i, f_i, g_i)$ that achieve optimal performance. The size of \mathcal{W}_i is $|\mathcal{W}_i| = \log M^*(n_i, \eta)$. Note that individual decoding functions g_i have range \mathcal{W}_i rather than \mathcal{W}_i^\ominus .

Encoding: A codeword $W_1 = w_1$ is selected uniformly at random from the codebook \mathcal{W}_1 . The mapping of this codeword into n_1 channel input letters, $X_{e_0+1}^{e_1} = f_1(w_1)$, is transmitted in channel usage times $n = e_0 + 1, e_0 + 2, \dots, e_1$.

Then a codeword $W_2 = w_2$ is selected uniformly at random from the codebook \mathcal{W}_2 . The mapping of this codeword into n_2 channel input letters, $X_{e_1+1}^{e_2} = f_2(w_2)$, is transmitted in channel usage times $n = e_1 + 1, e_1 + 2, \dots, e_2$.

This procedure continues until the last individual message code in the code is transmitted. That is, a codeword $W_k = w_k$ is selected uniformly at random from the codebook \mathcal{W}_k . The mapping of this codeword into n_k channel input letters, $X_{e_{k-1}+1}^{e_k} = f_k(w_k)$, is transmitted in channel usage times $n = e_{k-1} + 1, e_{k-1} + 2, \dots, e_k$.

We refer to channel usage times $n \in [e_{i-1} + 1, e_i]$ as the i th transmission epoch.

Decoding: For decoding, the channel output symbols for each epoch are processed separately. If any of the channel output symbols in an epoch are erasure symbols $?$, then a decoding erasure \ominus is declared for the message in that epoch, i.e. $\hat{W}_i = \ominus$. Otherwise, the individual message decoding function $g_i : \mathcal{Y}^{n_i} \rightarrow \mathcal{W}_i$ is applied to obtain $\hat{W}_i = g_i(Y_{e_{i-1}+1}^{e_i})$.

Performance Analysis: Having defined the communication scheme, we measure the error probability, transmission time, and expected transmission volume.

The decoder will either produce an erasure \ominus or use an individual message decoder g_i . When g_i is used, the maximal

error probability of individual message code error is bounded as $\lambda_i^{\max} < \eta$ by construction. Since declared erasures \ominus do not lead to error, and since all $\lambda_i^{\max} < \eta$, it follows that

$$\lambda^{\max} < \eta.$$

The transmission time is simply $N = \sum n_i$.

Recall the definition of expected transmission volume:

$$\mathbb{E} \left\{ \sum_{i \in \{1, \dots, k | \hat{W}_i \neq \ominus\}} \log_2 M_i \right\} = \sum_{i \in \{1, \dots, k | \hat{W}_i \neq \ominus\}} \mathbb{E} \{ \log_2 M_i \}$$

and the fact that the channel produces the erasure symbol \ominus for all channel usage times after death, $n > T$, but not before. Combining this with the length of an optimal code, $\log M^*(n_i, \eta)$, leads to the expression

$$\sum_{i=1}^k \Pr[T > e_i] \log M^*(n_i, \eta) = \sum_{i=1}^k R_T(e_i) \log M^*(n_i, \eta),$$

since all individual message codewords that are received in their entirety before the channel dies are decoded using g_i whereas any individual message codewords that are even partially cut off are declared \ominus .

Prop. 7 is valid for any choice of $(n_i)_{i=1}^k$. Since $(\log M^*)/n_i$ is monotonically increasing, it is better to use individual message codes that are as long as possible. With longer individual message codes, however, there is a greater chance of many channel usages being wasted if the channel dies in the middle of transmission. The basic trade-off is captured in picking the set of values $\{n_1, n_2, \dots, n_k\}$. For fixed and finite N , this involves picking an ordered integer partition $n_1 + n_2 + \dots + n_k = N$. We optimize this choice in Section VI.

C. Converse Arguments

Since we simply have operational expressions and no informational expressions in our development, as per Remark 1, and since optimal individual message codes and individual message decoders are assumed to be used, it may seem as though converse arguments are not required. This would indeed follow, if the following two things were true, which follow from Assumption 1. First, that there is no benefit in trying to decode the last partially erased message block. Second, that there is no benefit to errors-and-erasures decoding [28] by the g_i for codewords that are received before channel death. Under Assumption 1, Prop. 7 gives the best performance possible.

One might wonder whether Assumption 1 is needed. That there would be no benefit in trying to decode the last partially erased block follows from the conjecture that an optimal individual message code would have no latent redundancy that could be exploited to achieve a $\lambda_{i=\text{last}}^{\max} < \eta$, but this is a property of the actual optimal code.

On the other hand the effect of errors-and-erasures decoding by the individual message decoders is unclear. As given in

a Neyman-Pearson style result by Forney [28], decoding regions determined by likelihood ratio tests should be used to optimally trade off between (average) error probability and erasure probability by varying the threshold. It is unclear how the choice of threshold would affect the expected transmission volume

$$\sum_{i=1}^k (1 - \xi_i) R_T(e_i) \log M^*(n_i, \xi_i, \eta),$$

where ξ_i would be the specified erasure probability for individual message i , and $M^*(n_i, \xi_i, \eta)$ would be the maximum individual message codebook size under erasure probability ξ_i and maximum error probability η . Error bounds for errors-and-erasures decoding [28, Thm. 2] can certainly be converted into bounds on $\log M^*(n_i, \xi_i, \eta)$. It is an open question, however, whether there is a good Strassen-style approximation to this quantity.

VI. OPTIMIZING THE COMMUNICATION SCHEME

In Section V-B, we had not optimized the lengths of the individual message codes; we do so here. For fixed η and N , we maximize the expected transmission volume V over the choice of the ordered integer partition $n_1 + n_2 + \dots + n_k = N$:

$$\max_{(n_i)_{i=1}^k: \sum n_i = N} \sum_{i=1}^k R_T(e_i) \log M^*(n_i, \eta). \quad (5)$$

For finite N , this optimization can be carried out by an exhaustive search over all 2^{N-1} ordered integer partitions. If the death distribution $p_T(t)$ has finite support, there is no loss of generality in considering only finite N . Since exhaustive search has exponential complexity, however, there is value in trying to use a simplified algorithm. A dynamic programming formulation for the finite horizon case is developed in Section VII. The next subsection develops a greedy algorithm which is applicable to both the finite and infinite horizon cases and yields the optimal solution for certain problems.

A. A Greedy Algorithm

To try to solve the optimization problem (5), we propose a greedy algorithm that optimizes blocklengths n_i one by one.

Algorithm 1:

- 1) First maximize $R_T(n_1) \log M^*(n_1, \eta)$ through the choice of n_1 independently of any other n_i .
- 2) Then maximize $R_T(e_2) \log M^*(n_2, \eta)$ after fixing n_1 , but independently of later n_i .
- 3) Then maximize $R_T(e_3) \log M^*(n_3, \eta)$ after fixing e_2 , but independently of later n_i .
- 4) Continue in the same manner for all subsequent n_i .

Sometimes the algorithm produces the correct solution.

Proposition 8: The solution produced by the greedy algorithm, (n_i) , is locally optimal if

$$\begin{aligned} & \frac{\log M^*(n_i, \eta) - \log M^*(n_i - 1, \eta)}{\log M^*(n_{i+1} + 1, \eta) - \log M^*(n_{i+1}, \eta)} \\ & \geq \frac{R_T(e_{i+1})}{R_T(e_i - 1)} = \Pr[T > e_{i+1} | T > e_i - 1]. \end{aligned} \quad (6)$$

for each i .

Proof: The solution of the greedy algorithm partitions time using a set of epoch boundaries (e_i) . The proof proceeds by testing whether local perturbation of an arbitrary epoch boundary can improve performance. There are two possible perturbations: a shift to the left or a shift to the right.

First consider shifting an arbitrary epoch boundary e_i to the right by one. This makes the left epoch longer and the right epoch shorter. Lengthening the left epoch does not improve performance due to the greedy optimization of the algorithm. Shortening the right epoch does not improve performance since $R_T(e_i)$ remains unchanged whereas $\log M^*(n_i, \eta)$ does not increase since $\log M^*$ is a non-decreasing function of n_i .

Now consider shifting an arbitrary epoch boundary e_i to the left by one. This makes the left epoch shorter and the right epoch longer. Reducing the left epoch will not improve performance due to greediness, but enlarging the right epoch might improve performance, so the gain and loss must be balanced.

The loss in performance (a positive quantity) for the left epoch is

$$\Delta_l = R_T(e_i - 1) [\log M^*(n_i, \eta) - \log M^*(n_i - 1, \eta)]$$

whereas the gain in performance (a positive quantity) for the right epoch is

$$\Delta_r = R_T(e_{i+1}) [\log M^*(n_{i+1} + 1, \eta) - \log M^*(n_{i+1}, \eta)].$$

If $\Delta_l \geq \Delta_r$, then perturbation will not improve performance. The condition may be rearranged as

$$\frac{\log M^*(n_i, \eta) - \log M^*(n_i - 1, \eta)}{\log M^*(n_{i+1} + 1, \eta) - \log M^*(n_{i+1}, \eta)} \geq \frac{R_T(e_{i+1})}{R_T(e_i - 1)}$$

This is the condition (6), so the left-perturbation does not improve performance. Hence, the solution produced by the greedy algorithm is locally optimal. ■

Proposition 9: The solution produced by the greedy algorithm, (n_i) , is globally optimal if

$$\frac{\log M^*(n_i, \eta) - \log M^*(n_i - K_i, \eta)}{\log M^*(n_{i+1} + K_i, \eta) - \log M^*(n_{i+1}, \eta)} \geq \frac{R_T(e_{i+1})}{R_T(e_i - 1)}. \quad (7)$$

for each i , and any non-negative integers $K_i \leq n_i$.

Proof: The result follows by repeating the argument for local optimality in Prop. 8 for shifts of any possible size K_i . ■

There is an easily checked special case of global optimality condition (7) under the Strassen approximation.

Lemma 2: The function $\log M_S^*(z, \eta) - \log M_S^*(z - K, \eta)$ is a non-decreasing function of z for any K , where

$$\log M_S^*(z, \eta) = zC - \sqrt{z\rho}Q^{-1}(\eta) \quad (8)$$

is Strassen's approximation.

Proof: Essentially follows from the fact that \sqrt{z} is a concave \cap function in z . ■

Proposition 10: If the solution produced by the greedy algorithm using Strassen's approximation (8) satisfies $n_1 \geq$

$n_2 \geq \dots \geq n_k$, then condition (7) for global optimality is satisfied.

Proof: Since $R_T(\cdot)$ is a non-increasing survival function,

$$R_T(e_i - 1) \geq R_T(e_{i+1}). \quad (9)$$

Since the function $[\log M_S^*(z, \eta) - \log M_S^*(z - K, \eta)]$ is a non-decreasing function of z by Lem. 2, and since the n_i are in non-increasing order,

$$\begin{aligned} & \log M_S^*(n_i, \eta) - \log M_S^*(n_i - K, \eta) \\ & \geq \log M_S^*(n_{i+1} + K, \eta) - \log M_S^*(n_{i+1}, \eta). \end{aligned} \quad (10)$$

Taking products of both sides of (9) and (10), and rearranging yields the global optimality condition (7). ■

B. Geometric Death Distribution

A common failure mode for systems that do not age is a geometric death time T [2].

Proposition 11: When T is geometric, then the solution to (5) under Strassen's approximation yields equal epoch sizes. This optimal size is given by

$$\arg \max_{\nu} R_T(\nu) \log M^*(\nu, \eta).$$

Proof: We begin by showing that Algorithm 1 will produce a solution with equal epoch sizes. Recall that the survival function of a geometric random variable with parameter $0 < 1 - \alpha \leq 1$ is $R_T(t) = \alpha^t$. Therefore the first step of the algorithm will choose n_1 as

$$n_1 = \arg \max_{\nu} \alpha^{\nu} \log M^*(\nu, \eta).$$

The second step of the algorithm will choose

$$\begin{aligned} n_2 &= \arg \max_{\nu} \alpha^{n_1} \alpha^{\nu} \log M^*(\nu, \eta) \\ &= \arg \max_{\nu} \alpha^{\nu} \log M^*(\nu, \eta), \end{aligned}$$

which is the same as n_1 . In general,

$$\begin{aligned} n_i &= \arg \max_{\nu} \alpha^{e_i - 1} \alpha^{\nu} \log M^*(\nu, \eta) \\ &= \arg \max_{\nu} \alpha^{\nu} \log M^*(\nu, \eta), \end{aligned}$$

so $n_1 = n_2 = \dots$.

Such a solution satisfies $n_1 \geq n_2 \geq \dots$ and so it is optimal by Prop. 10. ■

Notice that the geometric death time distribution forms a boundary case for Prop. 10.

VII. DYNAMIC PROGRAMMING

The greedy algorithm of the previous section solves (5) under certain conditions. For finite N , a dynamic program (DP) may be used to solve (5) under any conditions. To develop the DP formulation [29], we assume that channel state feedback (whether the channel output is ? or whether it is either 0 or 1) is available to the transmitter, however solving the DP will show that channel state feedback is not required.

System Dynamics:

$$\begin{bmatrix} \zeta_n \\ \omega_n \end{bmatrix} = \begin{bmatrix} (\zeta_{n-1} + 1)\hat{s}_{n-1} \\ \omega_{n-1}w_{n-1} \end{bmatrix}, \quad (11)$$

for $n = 1, 2, \dots, N + 1$. The following state variables, disturbances, and controls are used:

- $\zeta_n \in \mathbb{Z}^*$ is a state variable that counts the location in the current transmission epoch
- $\omega_n \in \{0, 1\}$ is a state variable that indicates whether the channel is alive (1) or dead (0)
- $w_n \in \{0, 1\} \sim \text{Bern}(R_T(n))$ is a disturbance that kills (0) or preserves (1) the channel in the next time step
- $\hat{s}_n \in \{0, 1\}$ is a control input that starts (0) or continues (1) a transmission epoch in the next time step

Initial State: Since the channel starts alive (note that $R_T(1) = 1$) and since the first transmission epoch starts at the beginning of time,

$$\begin{bmatrix} \zeta_1 \\ \omega_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (12)$$

Additive Cost: Transmission volume $\log M^*(\zeta_n + 1, \eta)$ is credited if the channel is alive (i.e. $\omega_n = 1$) and the transmission epoch is to be restarted in the next time step (i.e. $1 - \hat{s}_n = 1$). This implies a cost function

$$c_n(\zeta_n, \omega_n, \hat{s}_n) = -(1 - \hat{s}_n)\omega_n \log M^*(\zeta_n + 1, \eta). \quad (13)$$

This is negative so that smaller is better.

Terminal Cost: There is no terminal cost, $c_{N+1} = 0$.

Cost-to-go: From time n to time $N + 1$ is:

$$\mathbb{E}_{\bar{w}} \left\{ \sum_{i=n}^N c_i(\zeta_i, \omega_i, \hat{s}_i) \right\} = -\mathbb{E}_{\bar{w}} \left\{ \sum_{i=n}^N (1 - \hat{s}_i)\omega_i \log M^*(\zeta_i + 1, \eta) \right\}.$$

Notice that the state variable ζ_n which counts epoch time is known to the transmitter and is determinable by the receiver through transmitter simulation. The state variable ω_n indicates the channel state and is known to the receiver by observing the channel output. It may be communicated to the transmitter through channel state feedback. It follows directly that:

Proposition 12: A communication scheme that follows the dynamics (11) and additive cost (13) achieves the transmission time-volume

$$\left(N, V = -\mathbb{E} \left[\sum_{n=1}^N c_n \right] \right)$$

at η -reliability.

DP may be used to find the optimal control policy (\hat{s}_n).

Proposition 13: The optimal $-V$ for the initial state (12), dynamics (11), additive cost (13), and no terminal cost is equal to the cost of the solution produced by the dynamic programming algorithm.

Proof: The system described by initial state (12), dynamics (11), and additive cost (13) is in the form of the *basic problem* [29, Sec. 1.2]. Thus the result follows from [29, Prop. 1.3.1] ■

A. Computation

In this subsection, we carry out the DP, using standard J notation for cost [29]. The base case at time $N + 1$ is

$$J_{N+1}(\zeta_{N+1}, \omega_{N+1}) = c_{N+1} = 0.$$

In proceeding backwards from time N to time 1:

$$\begin{aligned} J_n(\zeta_n, \omega_n) &= \min_{\hat{s}_n \in \{0, 1\}} \mathbb{E}_{w_n} \{ c_n(\zeta_n, \omega_n, \hat{s}_n) + J_{n+1}(f_n(\zeta_n, \omega_n, \hat{s}_n, w_n)) \}, \end{aligned}$$

for $n = 1, 2, \dots, N$, where $f_n(\zeta_n, \omega_n, \hat{s}_n, w_n) = [\zeta_{n+1} \ \omega_{n+1}]^T = [(\zeta_n + 1)\hat{s}_n \ \omega_n w_n]^T$. Substituting our additive cost function yields:

$$\begin{aligned} J_n(\zeta_n, \omega_n) &= \min_{\hat{s}_n \in \{0, 1\}} -\mathbb{E}_{w_n} \{ (1 - \hat{s}_n)\omega_n \log M^*(\zeta_n + 1, \eta) \} + \mathbb{E}_{w_n} \{ J_{n+1} \} \\ &= \min_{\hat{s}_n \in \{0, 1\}} -(1 - \hat{s}_n)R_T(n) \log M^*(\zeta_n + 1, \eta) + \mathbb{E}_{w_n} \{ J_{n+1} \}. \end{aligned} \quad (14)$$

Notice that the state variable ω_n dropped out of the first term when we took the expectation with respect to the disturbance w_n . This is true for each stage in the DP.

Proposition 14: For a channel that dies ($\mathcal{X}, p_a(y|x), p_d(y|x), p_T(t), \mathcal{Y}$), channel state feedback does not improve performance.

Proof: By repeating the expectation calculation in (14) for each stage n in the stage-by-stage DP algorithm, it is verified that state variable ω does not enter into the stage optimization problem. Hence the transmitter does not require channel state feedback to determine the optimal signaling strategy. ■

VIII. AN EXAMPLE

To provide some intuition on the choice of epoch lengths, we present a short example. Consider the channel that dies with BSC($\varepsilon = 0.01$) alive state and $p_T(t)$ that is uniform over a finite horizon of length 40 (disallowing death in the first time step):

$$p_T(t) = \begin{cases} 1/39, & t = 2, \dots, 40, \\ 0 & \text{otherwise.} \end{cases}$$

Our goal is to communicate with η -reliability, $\eta = 0.001$.

Since the death distribution has finite support, there is no benefit to transmitting after death is guaranteed. If we run the DP algorithm to optimize the ordered integer partition (under the Strassen approximation), we get a result ($n_1 = 20, n_2 = 12, n_3 = 6, n_4 = 2$). Notice that since the solution is in order, the greedy algorithm would also have succeeded. The expected transmission volume for this strategy is

$$\begin{aligned} V &= R_T(20) \log M^*(20, 0.001) + R_T(32) \log M^*(12, 0.001) \\ &\quad + R_T(38) \log M^*(6, 0.001) \\ &= (20/39) \cdot 9.2683 + (8/39) \cdot 3.9694 + (2/39) \cdot 0.5223 \\ &= 5.594 \text{ bits at reliability } 0.001. \end{aligned}$$

IX. CONCLUSION AND FUTURE WORK

We have formulated the problem of communication over channels that die and have shown how to maximize expected transmission volume at a given level of error probability.

There are several extensions to the basic formulation studied in this work that one might consider; we list a few:

- We have disallowed the decoder to declare erasures before the channel dies. As noted in Section V-C, one might consider the benefits of errors-and-erasures decoding [28] while the channel is alive.
- Inspired by synthetic biology [4], rather than thinking of death time as independent of the signaling scheme X_1^n , one might consider channels that die because they lose fitness as a consequence of operation: T would be dependent on X_1^n .² This would be similar to Gallager's panic button/child's toy channel, and would have intersymbol interference [17], [20]. There would also be strong connections to channels that heat up [31].
- Since channel death is indicated by erasure symbols, the receiver unequivocally knows death time. Other channel models might not have a distinct output letter for death and would need to detect death, perhaps using the theory of estimating stopping times [32].
- One might consider a channel that is born at a random time in addition to dying at a random time. Channel state feedback would provide benefit.
- This work has simply considered the channel coding problem, however there are several formulations of end-to-end information transmission problems over channels that die, which are of interest in many application areas. There is no reason to suspect a separation principle.
- Networks of birth-death channels are also of interest.

Randomly stepping back from infinity leads to some new understanding.

ACKNOWLEDGMENT

We thank Barry Canton (Ginkgo BioWorks) and Drew Endy (Stanford University) for discussions on synthetic biology that initially inspired this work. Discussions with Baris Nakiboglu are also appreciated.

REFERENCES

- [1] I. M. Jacobs, "Connectivity in probabilistic graphs: An abstract study of reliable communications in systems containing unreliable components," Ph.D. dissertation, Massachusetts Institute of Technology, Aug. 1959.
- [2] D. J. Davis, "An analysis of some failure data," *J. Am. Stat. Assoc.*, vol. 47, no. 258, pp. 113–150, June 1952.
- [3] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 5, no. 1, p. 5, Feb. 2009.
- [4] B. Canton, A. Labno, and D. Endy, "Refinement and standardization of synthetic biological parts and devices," *Nat. Biotechnol.*, vol. 26, no. 7, pp. 787–793, July 2008.
- [5] J. D. Bekenstein, "The limits of information," *Stud. Hist. Philos. Mod. Phys.*, vol. 32, no. 4, pp. 511–524, Dec. 2001.
- [6] D. R. Headrick, *The Invisible Weapon: Telecommunications and International Politics, 1851–1945*. New York: Oxford University Press, 1991.
- [7] D. Slepian, "Bounds on communication," *Bell Syst. Tech. J.*, vol. 42, pp. 681–707, May 1963.
- [8] S. J. MacMullan and O. M. Collins, "A comparison of known codes, random codes, and the best codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 7, pp. 3009–3022, Nov. 1998.
- [9] J. N. Laneman, "On the distribution of mutual information," in *Proc. Inf. Theory Appl. Inaugural Workshop*, Feb. 2006.
- [10] Y. Polyanskiy, H. V. Poor, and S. Verdú, "New channel coding achievability bounds," in *Proc. 2008 IEEE Int. Symp. Inf. Theory*, July 2008, pp. 1763–1767.
- [11] D. Buckingham and M. C. Valenti, "The information-outage probability of finite-length codes over AWGN channels," in *Proc. 42nd Annu. Conf. Inf. Sci. Syst. (CISS 2008)*, Mar. 2008, pp. 390–395.
- [12] G. Wiechman and I. Sason, "An improved sphere-packing bound for finite-length codes over symmetric memoryless channels," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 1962–1990, May 2008.
- [13] A. Goldsmith, *Wireless Communications*. New York: Cambridge University Press, 2005.
- [14] J. K. Wolf, A. D. Wyner, and J. Ziv, "The channel capacity of the postal channel," *Inf. Control*, vol. 16, no. 2, pp. 167–172, Apr. 1970.
- [15] M. Zeng, R. Zhang, and S. Cui, "On the outage capacity of a dying channel," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM 2008)*, Dec. 2008.
- [16] E. Modiano, "An adaptive algorithm for optimizing the packet size used in wireless ARQ protocols," *Wireless Netw.*, vol. 5, no. 4, pp. 279–286, July 1999.
- [17] R. G. Gallager, *Information Theory and Reliable Communication*. New York: John Wiley & Sons, 1968.
- [18] S. M. Ross, *Stochastic Processes*. John Wiley & Sons, 1996.
- [19] R. G. Gallager, *Discrete Stochastic Processes*. Boston: Kluwer Academic Publishers, 1996.
- [20] R. Gallager, *Information Theory and Reliable Communication*, ser. International Centre for Mechanical Sciences, Courses and Lectures. Vienna: Springer-Verlag, 1972, no. 30.
- [21] A. J. Goldsmith and P. P. Varaiya, "Capacity, mutual information, and coding for finite-state Markov channels," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 868–886, May 1996.
- [22] L. E. Braten and T. Tjelta, "Semi-Markov multistate modeling of the land mobile propagation channel for geostationary satellites," *IEEE Trans. Antennas Propag.*, vol. 50, no. 12, pp. 1795–1802, Dec. 2002.
- [23] J. Wang, J. Cai, and A. S. Alfa, "New channel model for wireless communications: Finite-state phase-type semi-Markov channel model," in *Proc. IEEE Int. Conf. Commun. (ICC 2008)*, May 2008, pp. 4461–4465.
- [24] P. Kaski and P. R. J. Östergård, *Classification Algorithms for Codes and Designs*. Berlin: Springer, 2006.
- [25] A. Barg and A. McGregor, "Distance distribution of binary codes and the error probability of decoding," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4237–4246, Dec. 2005.
- [26] V. Strassen, "Asymptotische abschätzungen in Shannons informations-theorie," in *Transactions of the 3rd Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*. Prague: Pub. House of the Czechoslovak Academy of Sciences, 1962, pp. 689–723.
- [27] L. Weiss, "On the strong converse of the coding theorem for symmetric channels without memory," *Q. Appl. Math.*, vol. 18, no. 3, pp. 209–214, Oct. 1960.
- [28] G. D. Forney, Jr., "Exponential error bounds for erasure, list, and decision feedback schemes," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 2, pp. 206–220, Mar. 1968.
- [29] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 3rd ed. Belmont, MA: Athena Scientific, 2005, vol. 1.
- [30] C. Darwin, *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. London: John Murray, 1859.
- [31] T. Koch, A. Lapiditho, and P. P. Sotiriadis, "Channels that heat up," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3594–3612, Aug. 2009.
- [32] U. Niesen and A. Tchamkerten, "Tracking stopping times through noisy observations," *IEEE Trans. Inf. Theory*, vol. 55, no. 1, pp. 422–432, Jan. 2009.

²In this Darwin sesquicentennial/bicentennial year [30], *survival of the fittest* might be apropos.