Coding to reduce delay on a permutation channel

John MacLaren Walsh and Steven Weber

Abstract— **This paper introduces and analyzes the permutation channel: an error free channel that takes a block of M codewords as the input, and returns a random permutation of the codewords as output, where the permutation reflects the sequence of time instants at which the codewords are received. The decoding delay of a dataword is defined to be proportional to the earliest time that the dataword and all preceding datawords are recoverable from the currently held codewords. The problem is motivated by practical networking scenarios where packet reordering at the receiver may limit the performance of an application, e.g., a media stream where playback may be stalled on account of waiting for a particular frame to arrive. The key performance tradeoff is the code rate and decoding delay. Intuitively decoding delay may be reduced by spreading datawords across multiple codewords, but this redundancy lowers the code rate. We introduce a zero delay code, also known as a priority encoded transmission code, based on independent linear combinations of portions of the datawords. We generalize the latter code to non-zero delays, and formulate the rate delay tradeoff problem for large block lengths as a calculus of variations problem, which we then solve. We establish that our code is able to achieve an asymptotic code rate of $(1 - \log \sqrt{\delta})^{-1}$ when subject to a delay bound of δ. Finally, we note that the permutation channel is a particular instance of the degraded broadcast channel, where receivers correspond to information available at the codeword reception instants, establishing the rate optimality of the codes introduced.**

I. Introduction

This paper addresses the problem of how to encode an ordered sequence of datawords into codewords when the sequence of codewords arrive in permuted order at the receiver, with a channel that additionally provides the location of each received dataword within the original sequence. The model presumes that each dataword can only be "decoded" at the receiver once each of the preceding datawords in the sequence has been received. The channel introduces no errors and as such there is no need for channel coding in the conventional sense. Instead of coding for channel errors, we add redundancy to reduce the decoding "delay". In a sense the problem domain shares some similarities with that of network coding [1], where there too channels are assumed to be error free (at least in the original formulation), and the objective is to maximize throughput by appropriate combination of packets. Perhaps more compelling, however, is the fact that the permutation channel is in fact a particular instance of the degraded broadcast channel [2], [3], as we discuss in Section V. The capacity region of this degraded broadcast channel yields timesharing as a capacity achieving strategy, which translates to the use of unequal error protection codes for the data corresponding to different time instants. Indeed, the codes we propose may be considered as instances of priority encoded transmission [4] codes, with different source datawords associated with different priorities. These codes have been shown to achieve points on the capacity region [5] of the afore mentioned broadcast channel.

Contact author: Steven Weber (`sweber@ece.drexel.edu`). Both authors are with the Department of Electrical and Computer Engineering of Drexel University in Philadelphia, PA. Manuscript date: September 28, 2007.

Example. To solidify the discussion, consider the simplest possible case of two datawords, W_1 and W_2; see Figure 1. These datawords are encoded into codewords $(X_1, X_2) = f(W_1, W_2)$. The permutation channel selects at random either $Y = (X_1, X_2)$ or $Y = (X_2, X_1)$, where the sequence indicates the order in which the receiver obtains the codewords. Let time be slotted on the codeword arrival times; thus Y_1 arrives at time 1 and Y_2 arrives at time 2. The presumption is that the receiver can only decode dataword W_2 once datawords W_1 and W_2 have been received. The objective is to design the encoder so that the "delay" in decoding both W_1, W_2 is minimized. Each dataword has a decoding deadline, for simplicity we say W_1 is due at time 1 and W_2 is due at time 2; there is no benefit for early arrivals. The (random) dataword delay is then defined as the (normalized) positive difference between its decoding time and its deadline. Consider three obvious coding strategies. **First**, the identity map yields $X_1 = W_1$ and $X_2 = W_2$, which means the dataword delays are both zero if $Y = (X_1, X_2)$ but the delay for W_1 is 1 if $Y = (X_2, X_1)$ since W_1 can first be decoded at time 2. **Second**, a trivial zero delay encoding is to set $X_1 = X_2 = (W_1, W_2)$, but this incurs a cost in the rate of the code; the rate of this code is $1/2$ whereas the rate of the identity code is 1. A **third** scheme is to set

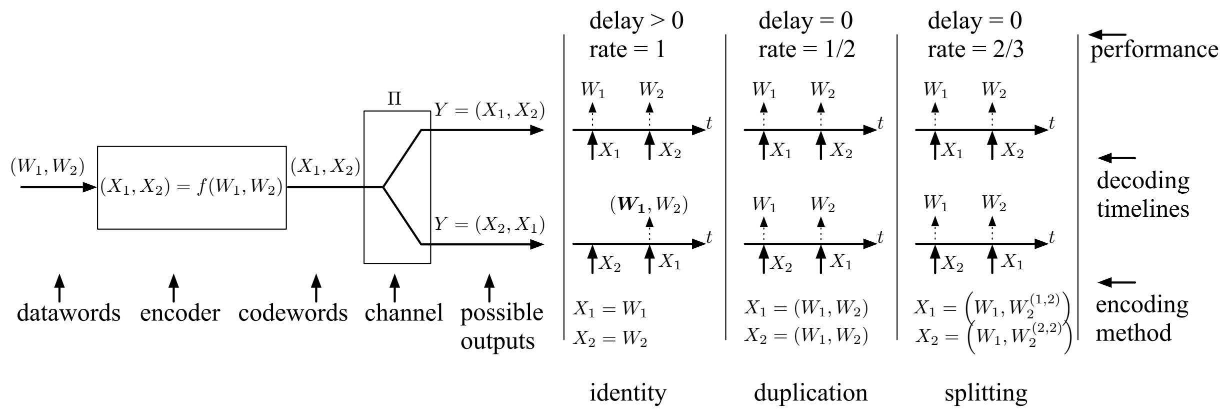

Fig. 1. Illustration of the order two permutation channel, Π. Datawords (W_1, W_2) are encoded as (X_1, X_2) which are received either in order, $Y = (X_1, X_2)$, or reversed, $Y = (X_2, X_1)$. W_2 is only decodable once W_1 is received. The timelines on the right side of the figure illustrate the possible times at which the datawords are decodable under the three encoding methods (identity, duplication, splitting).

$X_1 = (W_1, W_2^{(1,2)})$ and $X_2 = (W_1, W_2^{(2,2)})$, where the notation $W_k^{(i,l)}$ indicates part i of l parts of the symbol-string representation of W_k. Thus this code places all of W_1 in both X_1, X_2, and breaks W_2 into two parts, placing half of the symbols in each packet. This too is a zero delay code, but the rate is $2/3$, which is superior to the previous code. Although instructive, unfortunately the $M = 2$ channel does not illustrate the richness of the problem found for $M > 2$.

Motivation and Related Work. This paper studies the rate delay tradeoff for a block of M datawords that are encoded and sent over the permutation channel. The model and the performance metrics of rate and delay are formally introduced in Section II. One motivation for this model is an application layer view of a wired network employing multiple routes from source to destination, where the application packets (codewords) are, say, encoded from a block of frames (datawords) in a video stream. The various packets become randomly reordered on account of the different routes they travel and the corresponding fluctuations in the queueing and processing delays at the nodes along these routes. Practical transport layer algorithms include the location (identities) of the received packets within the original transmitted sequence, motivating the availability of this information in our model. At the receiver, the media client must play the frames in order, and as such a frame is only "decoded" once it and all preceding frames have arrived. It is this requirement of successive decoding, as opposed to a block based decoding framework which distinguishes our approach from a classic end to end efficient erasure correcting code [6] (e.g. LDPC or RS) approach to the coding for network transmission problem. Rather, the code developed here, which can be considered a special case of priority encoded transmission [4], [5] codes, simply time shares different efficient (maximum distance separable MDS) erasure codes. Apart from a modestly different mathematical formulation and a different practical purpose, the most significant innovation we provide over existing PET work is a calculation of both the finite and assymptotic rate delay tradeoffs and a firmer and simpler connection with degraded broadcast channels. Apart from some qualitative and conceptual similarities, [7] is for a different cascaded binary erasure channel and thus a different problem. Additional applications to networks employing network coding to achieve multicast capacity are described in the conclusion.

Summary of results. Our first result, Lemma 1, identifies the average decoding delay in the absence of coding, i.e., under the identity code. The result is important in establishing the delay under a rate 1 code, which is therefore an upper bound on delay for codes that sacrifice rate to reduce delay. Next, we introduce a superior zero delay code based on taking independent linear combinations of the portions of the datawords (modulo the field size q), which amounts to designing maximum distance separable linear codes, and establish the rate to be $(1+1/2+\cdots+1/M)^{-1}$. We then generalize this code to handle arbitrary delay constraints, and formulate the rate delay tradeoff for this family of codes as a particular integer optimization problem (Lemma 2). For large M we show this optimization problem becomes a calculus of variations problem, and our Theorem 1 solves this problem. We show that the asymptotic rate delay tradeoff to be that a rate of $R = (1 - \log \sqrt{\delta})^{-1}$ is achievable using our proposed code under a (normalized) delay constraint of $\delta \in [0, 1]$. Finally, we formally establish the permutation channel as a particular instance of the degraded broadcast channel.

Outline. The rest of this paper is as follows. Section II defines the permutation channel and the performance metrics rate and delay. The tradeoff between rate and delay is found in Section III, and an asymptotic tradeoff (in the block length) is presented in Section IV. In Section V we describe the permutation channel as a specific instance of the degraded broadcast channel; Section VI concludes the paper.

II. The permutation channel

Datawords. The block length, M, denotes both the number of distinct datawords and codewords in a block. The ordered sequence of datawords comprising a block, called a datablock, is denoted $\mathbf{W} = (W_1, \ldots, W_M)$. Each dataword is in turn an ordered sequence of K symbols, $W_m = (W_m^{(1)}, \ldots, W_m^{(K)}),\ m = 1, \ldots, M$, where each symbol $W_m^{(k)}$ is an element of $\mathrm{GF}(q)$. We discuss the choice of appropriate q in the sequel. Note that each datablock consists of MK symbols. The datablock is assumed to be the output of a source coding, and therefore incompressible.

Encoder. The ordered sequence of codewords, $\mathbf{X} = (X_1, \ldots, X_M)$, called a codeblock, is obtained by passing the datablock into the encoder $\mathbf{X} = f(\mathbf{W})$. Codewords also consist of an ordered sequence of symbols from $\mathrm{GF}(q)$, each of length L. We say a code is *non–degenerate* provided $\mathbf{W}$ may be recovered from $\mathbf{X}$.

Definition 1: **Encoding rate.** The encoding rate, r, of an encoder f is defined to be the ratio of the symbol length of the datablock over the symbol length of the codeblock, $r = MK/ML = K/L$.

Note that non-degeneracy requires $r \leq 1$. Moreover, the encoding rate is independent of q since the bit–string length of both the datablock and the codeblock is the corresponding symbol length times $\lceil \log_2(q) \rceil$.

Permutation channel. The codeblock, $\mathbf{X}$, is sent over the permutation channel and a permutation $\mathbf{Y} = \pi(\mathbf{X})$ is received. In particular, the received codeblock, $\mathbf{Y} = (Y_1, \ldots, Y_M)$, represents the sequence of codewords in order of their arrival at the receiver. Time is slotted at codeword arrival times so that received codeword Y_m is said arrive at time m. It is important to emphasize that in our model the codewords comprising the codeblock are reordered by the channel, but the sequence of symbols comprising each codeword is not disturbed. In the example of media frames placed in packets on multiple paths given in the introduction, this assumption models the fact that packets are reordered by the channel, while bits in a packet are not. The *general permutation channel* is characterized by a probability mass function $\mathbf{p} = (p(\pi),\ \pi \in \Pi)$, where $p(\pi)$ is the probability of permutation π, and Π is the set of all possible $M!$ permutations of the codeblock. The *uniform permutation channel* assumes all permutations are equally likely, i.e., $p(\pi) = 1/M!,\ \pi \in \Pi$. Aside from the trivial identity code, all the codes in this paper have deterministic delay, and are therefore insensitive to the channel distribution. In analyzing the identity code we assume the uniform permutation channel.

Decoder. Decoding in the conventional sense is trivial due to the fact that there are no errors introduced by the channel; as such the only relevant performance measure is the decoding delay for each dataword. Each dataword has an associated soft deadline; soft meaning that a penalty is incurred for a decoding delay beyond the deadline, but the dataword is still of use to the receiver. For simplicity we assume the deadline for dataword W_m is time m. Define the sequence of random decoding times for $m = 1, \ldots, M$

$$T_m = \inf \left\{ \begin{array}{c} k : (W_1, \ldots, W_m) \text{ is recoverable} \\ \text{from } (Y_1, \ldots, Y_k) \end{array} \right\} \quad (1)$$

reflecting our modeling assumption that dataword W_m is first decodable once $W_1, \ldots, W_m$ are recoverable from the currently held codewords. In the example of media frames this reflects the idea that a frame is only of use to the application when all preceding frames are also available. Note that $T_m \leq M$ for each $m = 1, \ldots, M$ for all non–degenerate codes.

Definition 2: **Decoding delay.** The decoding delay for dataword m is defined to be the random variable $D_m = \frac{2}{M}(T_m - m)^+$ (N.B., $(x)^+ = \max\{x, 0\}$). The expected average delay is defined to be

$$d(M) = \mathbb{E}\left[\frac{1}{M} \sum_{m=1}^{M} D_m \right], \quad (2)$$

where the expectation is taken with respect to the uniform probability distribution on the set of possible permutations of the codeblock.

The $\{D_m\}$ are in general neither independent nor identically distributed. Setting the decoding delay to be proportional to $(T_m - m)^+$ captures the idea that there is no gain for decoding the dataword before its deadline. Again using the example of media frames, this assumption reflects the fact that the playback quality of a media stream is independent of precisely when the frames arrive, provided they arrive in time for playback. The delay is normalized by $2/M$ for reasons that will be evident in the sequel.

The subject of this paper is the rate delay tradeoff for a class of codes defined in the next section.

III. Rate delay tradeoff

A. Identity code

One extreme of the rate delay tradeoff is to simply employ the identity code, $\mathbf{X} = \mathbf{W}$. This code trivially has rate $r = 1$. Lemma 1 identifies the expected average delay under this code.

Lemma 1: The expected average delay of the identity code on the uniform permutation channel is

$$d(M) = \frac{(M+1)(M+2-2H(M+1))}{M^2}, \qquad (3)$$

where $H(k)$ is the Harmonic number: $H(k) = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$.
The proof is found in the Appendix. Note that $d(M)$ is monotone concave increasing in M and that $\lim_{M\to\infty} d(M) = 1$. As we will discuss shortly, we can reduce the delay at the expense of reducing the rate. The motivation for normalizing the delay by $2/M$ is now evident: under this normalization the "worst case" delay is 1. That is, the set of all possible rate delay points is the unit box $[0,1]^2$, and the identity code asymptotically achieves the point $(r,d) = (1,1)$.

B. Zero delay independent linear combination code

Consider the following code, which we call a zero delay independent linear combination code:

$$\left[\begin{array}{c} X_1 \\ X_2 \\ X_3 \end{array}\right] = \left[\begin{array}{c|c|c} W_1 & W_2^{(1,2)} & W_3^{(1,3)} \\ W_1 & W_2^{(2,2)} & W_3^{(2,3)} \\ W_1 & W_2^{(1,2)} \oplus W_2^{(2,2)} & W_3^{(3,3)} \end{array}\right], \qquad (4)$$

where $\oplus$ denotes symbol–wise addition of the symbol strings using $GF(q)$ arithmetic. This code is zero delay since W_m is decodable from an arbitrary subset of m codewords, for $m = 1,2,3$. Moreover, the rate of this code is

$$r = \frac{3K}{3K(1+1/2+1/3)} = 6/11. \qquad (5)$$

which is superior to the best corresponding repetition/splitting code (for $M = 3$), which has rate $1/2$.

Code construction. For general M the zero delay independent linear combination code is constructed as follows. Fix M. Split the string of K symbols comprising each dataword W_m into m parts, denoted $W_m = [W_m^{(1,m)}|\cdots|W_m^{(m,m)}]$, for $m = 1,\ldots,M$. Each of the m sub-strings $W_m^{(k,m)}$ is of length K/m symbols (from $\mathrm{GF}(q)$). Next, create a sequence of M matrices, $\mathbf{C}^{(M,1)},\ldots,\mathbf{C}^{(M,M)}$, where matrix $\mathbf{C}^{(M,m)}$ is of dimensions $M \times m$. Select the entries of matrix $\mathbf{C}^{(M,m)}$ from $\mathrm{GF}(q)$ such that any subset of m rows of the matrix are linearly independent. Such matrices are generator matrices of maximum distance separable (MDS) linear codes. Equivalently, for each m, each square sub-matrix of $\mathbf{C}^{(M,m)}$ (of dimension $m \times m$), has full rank, m. Each codeword X_k has the form $X_k = \left[Z_k^1|\cdots|Z_k^M\right]$ where Z_k^m is the portion of codeword k generated from dataword m, for $k = 1,\ldots,M$. Set

$$Z_k^m = \sum_{j=1}^{m} C_{k,j}^{(M,m)} W_m^{(j,m)}, \ \forall k,m \in \{1,\ldots,M\} \qquad (6)$$

as the linear combination (in $GF(q)$ arithmetic) of the sub-strings $W_m^{(1,m)},\ldots,W_m^{(m,m)}$ of dataword m, where the sub-strings are weighted by the coefficients in row k of matrix $\mathbf{C}^{(M,m)}$. Note that $C_{k,j}^{(M,m)} \in \mathrm{GF}(q)$ is a scalar, while $W_m^{(j,m)} \in \mathrm{GF}(q)^{K/m}$ is a substring of K/m symbols. The multiplication of the scalar by the string is therefore assumed to be performed component-wise, and the resulting $Z_k^m \in \mathrm{GF}(q)^{K/m}$ is itself a sub-string of K/m symbols.

Zero delay and dataword recovery. This code will achieve zero delay provided the requirement is met that each subset of m rows of $\mathbf{C}^{(M,m)}$ are linearly independent. To see this, suppose the receiver at time m is in possession of an arbitrary subset of m codewords with indices $k_1,\ldots,k_m$. Then dataword W_m may be recovered from these m codewords since the corresponding coefficient sub-matrix is invertible.

$$\left[\begin{array}{c} Z_{k_1}^m \\ \vdots \\ Z_{k_m}^m \end{array}\right] = \left[\begin{array}{c} C_{k_1}^{(M,m)} \\ \vdots \\ C_{k_m}^{(M,m)} \end{array}\right] \left[\begin{array}{c} W_m^{(1,m)} \\ \vdots \\ W_m^{(m,m)} \end{array}\right] \Rightarrow$$

$$W_m = \left[\begin{array}{c} W_m^{(1,m)} \\ \vdots \\ W_m^{(m,m)} \end{array}\right] = \left[\begin{array}{c} C_{k_1}^{(M,m)} \\ \vdots \\ C_{k_m}^{(M,m)} \end{array}\right]^{-1} \left[\begin{array}{c} Z_{k_1}^m \\ \vdots \\ Z_{k_m}^m \end{array}\right].$$

The received packets include their encoding vectors, but the associated rate loss is neglected since it can be made arbitrarily small by choosing a sufficiently large q.

Rate. The rate of the code is

$$r = \frac{MK}{MK\sum_{m=1}^{M}\frac{1}{m}} = \left(\sum_{m=1}^{M}\frac{1}{m}\right)^{-1} = H(M)^{-1}. \qquad (7)$$

Note that the asymptotic rate of the code is zero.

Selection of q. The field size, q, must be sufficiently large to permit the coefficient matrices to each have the above linear independence property, that is q must be large enough to guarantee existence of MDS separable linear codes of the given rate. Define $\mathcal{C}\left(\mathbf{C}^{(M,m)}\right)$ as the set of all $m \times m$ matrices obtained by selecting each of the possible $\binom{M}{m}$ combinations of rows from the $M \times m$ matrix $\mathbf{C}^{(M,m)}$. Define

$$q(M) = \inf \left\{ q : \begin{array}{c} \exists\ \mathbf{C}^{(M,m)} \in \mathrm{GF}(q)^{M\times m} \text{ s.t.} \\ \mathrm{rank}(\mathbf{A}) = m,\ \ \mathbf{A} \in \mathcal{C}\left(\mathbf{C}^{(M,m)}\right), \end{array} \right\}$$

as the smallest field size such that there exists a matrix $\mathbf{C}^{(M,m)}$ with the property that each square sub-matrix has full rank, for each $m = 1, \ldots, M$. The reader familiar with network coding will recall a similar rank requirement in that context, where the rank requirement is satisfied with probability $1-O(q^{-1})$ when the rows of the matrix are chosen in an appropriate random manner [8].

C. General delay independent linear combination code

The independent linear combination code of the previous sub-section achieves zero delay by ensuring that dataword W_m is divided into m parts, $W_m = [W_m^{(1,m)}|\cdots|W_m^{(m,m)}]$, and thereby achieves a rate of $H(M)^{-1}$. We can increase the rate of the code by relaxing the zero delay requirement; this is achieved by dividing W_m into $m' > m$ parts. Let the vector $\mathbf{m} = (m_1, \ldots, m_M)$ have elements m_k indicating the number of components into which we divide dataword W_k. Then the rate and delay of the corresponding independent linear combination code are

$$r(m_1, \ldots, m_M) = \left(\sum_{k=1}^{M} \frac{1}{m_k} \right)^{-1}, \quad (8)$$

$$d(m_1, \ldots, m_M) = \frac{1}{M} \sum_{k=1}^{M} \frac{2}{M} \left(\max_{j\le k} m_j - k \right)^{+}. \quad (9)$$

Note that the delay expression employs the fact that the time at which dataword W_k may be decoded, T_k, is in fact deterministic, and is given by $T_k = \max_{j\le k} m_j$. The feasible set are all tuples $\mathbf{m} \in \{1, \ldots, M\}^M$. We are interested in determining the maximum achievable rate r_δ^* with a delay of at most δ. The maximization and non-negativity of the delay function make clear that the feasible set may thus be reduced to

$$\mathcal{R} = \left\{ \begin{pmatrix} m_1 \\ \vdots \\ m_M \end{pmatrix} : \begin{array}{c} 1 \le m_1 \le \cdots \le m_M \le M, \\ m_k \in \{k, \ldots, M\}, \\ k = 1, \ldots, M \end{array} \right\}$$

Combining these observations yields the following Lemma.

Lemma 2: The maximum rate independent linear combination code with a decoding delay of at most δ is the solution of the following integer optimization problem:

$$\max_{\mathbf{m}\in\mathcal{R}\bigcap\mathcal{W}_\delta} \left(\sum_{k=1}^{M} \frac{1}{m_k} \right)^{-1} \quad (10)$$

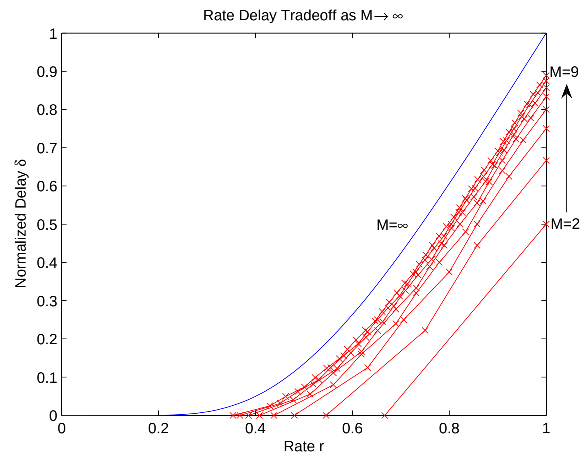

Fig. 2. Rate delay tradeoff for $M = 2, 3, 4, 5, 6, 7, 8, 9$ and $M = \infty$. The asymptotic delay is $D_r^* = \exp\left\{2\left(1 - \frac{1}{r}\right)\right\}$.

where

$$\mathcal{W}_\delta := \left\{ \begin{pmatrix} m_1 \\ \vdots \\ m_M \end{pmatrix} \;\middle|\; \frac{1}{M} \sum_{k=1}^{M} \frac{2}{M} \left(\max_{j\le k} m_j - k \right)^{+} \le \delta \right\} \quad (11)$$

In the following section we identify the asymptotic solution of the above optimization problem as $M \to \infty$.

IV. Asymptotic rate delay tradeoff

For M large we are able to solve the integer optimization problem in (10) by solving a related calculus of variations problem. It is convenient to introduce the change of variables from $\mathbf{m}$ to $\mathbf{x}$ with $x_k = \frac{m_K}{M}$. As $M \to \infty$ the vector $\mathbf{m}/M$ approximates the function $x = \{x(t),\ 0 \le t \le 1\}$ with domain and range of $[0, 1]$. Moreover, the summations in the definition of the rate and delay in (8) become the following integrals:

$$\begin{aligned} r(\mathbf{x}) &= \left(\sum_{k=1}^{M} \frac{1}{M} \frac{1}{x_k} \right)^{-1} \to \\ R(x) &= \left(\int_0^1 \frac{1}{x(t)} \mathrm{d}t \right)^{-1}, \\ d(\mathbf{x}) &= 2 \sum_{k=1}^{M} \frac{1}{M} \left(\max_{j\le k} x_j - \frac{k}{M} \right)^{+} \to \\ D(x) &= 2 \int_0^1 \left(\sup_{0\le s\le t} x(s) - t \right)^{+} \mathrm{d}t, \end{aligned}$$

where the sums are written in a form suggesting the Riemann integral approximation for large M. Let $\mathcal{C}$ be the space of continuously differentiable functions $x(t)$ defined on $[0, 1]$ taking values in $[0, 1]$. The integer

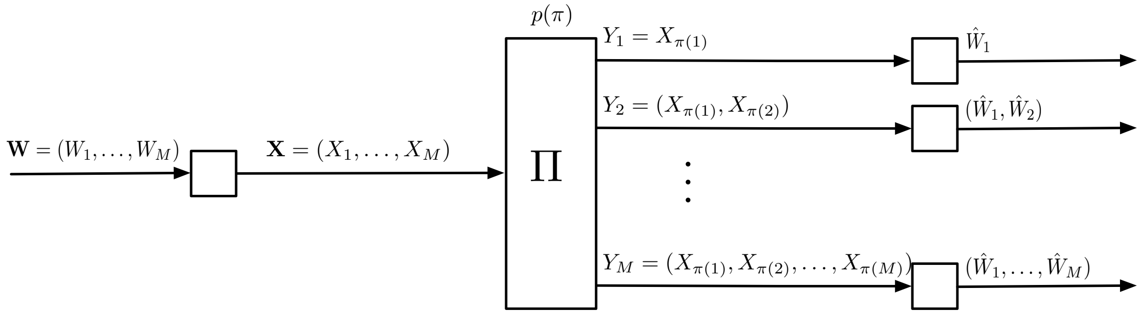

Fig. 3. The permutation channel is a special case of an M–receiver degraded broadcast channel. The "receivers" are the cumulative set of available codewords at each of the codeword reception times.

program in (10) asymptotically becomes the calculus of variations problem:

$$\max_{x \in \mathcal{C}} \left\{ R(x) \;\middle|\; D(x) \le \delta \right\}. \tag{12}$$

Theorem 1: The solution of (12) is

$$x^*(t) = \begin{cases} \sqrt{\delta}, & 0 \le t \le \sqrt{\delta} \\ t, & \sqrt{\delta} < t \le 1 \end{cases}. \tag{13}$$

The associated maximum rate is

$$R^*_\delta = R(x^*) = \left(1 - \log \sqrt{\delta}\right)^{-1}. \tag{14}$$

The proof employs the Euler–Lagrange equation to identify the optimal x^*. The function R^*_δ gives the optimal rate as a function of the maximum permissible delay, $D \le \delta$. Equivalently, we can solve for D^*_ρ, the minimum delay obtainable for a rate $R \ge \rho$ as

$$D^*_\rho = \exp\left\{ 2\left(1 - \frac{1}{\rho}\right) \right\}. \tag{15}$$

Note that the optimal function x^* implies that the asymptotic optimal independent linear combination code incurs the steepest delay costs for early datawords, and that zero delay is incurred for datawords of index exceeding $\sqrt{\delta}M$.

In Figure 2, left, we show the rate delay tradeoff for $M = 2, 3, 4, 5, 6, 7, 8, 9$ and $M = \infty$. The plot illustrates that the asymptotic tradeoff from Theorem 1 is quite accurate even for moderate values of M.

V. Connection with the Degraded Broadcast Channel

Define Y_m as the collection of codewords received by time m; thus $Y_m = ((X_{\pi(1)}, \pi(1)), \ldots, (X_{\pi(m)}, \pi(m)))$. The sequence $(Y_M, Y_{M-1}, \ldots, Y_2, Y_1)$ is a Markov chain with transition probabilities

$$\mathbb{P}\left(Y_{m-1} = (y_1, \ldots, y_m) \setminus y_k \;\middle|\; Y_m = (y_1, \ldots, y_m)\right) = \frac{1}{m}, \quad k = 1, \ldots, m.$$

That is, when viewing the cumulative codeword reception process in reverse time at each time step one codeword is lost, selected uniformly at random. Viewing each time step $1, \ldots, M$ as a receiver, the permutation channel is seen as a special case of the degraded broadcast channel [2], [3]; see Figure 3. Note that the branches leaving the right side of the permutation channel are the information available to each of the M receivers (i.e., at each of the M time slots), whereas the branches leaving the right side of the permutation channel in Figure 1 are the two possible realizations of the channel. The definition of the broadcast channel distinguishes information meant for individual receivers and common information meant for a subset of the receivers. In the permutation channel we see that W_1 is common to all receivers, W_2 is common to receivers $2, \ldots, M$, and in general W_m is common information to receivers $m, \ldots, M$. The capacity region results for the degraded broadcast channel can thus be applied to the particular form of the permutation channel. The next example gives the case for $M = 2$, demonstrating that the codes we have designed are rate optimal.

Example. To illustrate the fertile ground provided by this broadcast channel interpretation, we provide an example showing that our $M = 2$ zero delay codes achieve a point on the boundary of the corresponding degraded broadcast channel capacity region. The capacity region of rates R_1 to Y_1 and R_2 to Y_2 in units of bits per channel use on the corresponding degraded broadcast channel is the closure of the convex hull of the region $R_1 \le I(U_1; Y_1)$, $R_2 \le I(U_2; Y_2 | U_1)$ over all U_1, U_2 obeying $U_1 \to U_2 \to Y_2 \to Y_1$. We will choose the specific case when Y_2 is two bits $Y_2 = (Y_{2,1}, Y_{2,2})$. U_1 and U_2 may then be selected to be two bit random variables. The capacity region of this degraded broadcast channel is shown in Figure 4, Right, along with the rates of two zero delay codes. Surprisingly, the independent linear combination code is rate optimal, even though it only requires a block length of one permutation degraded

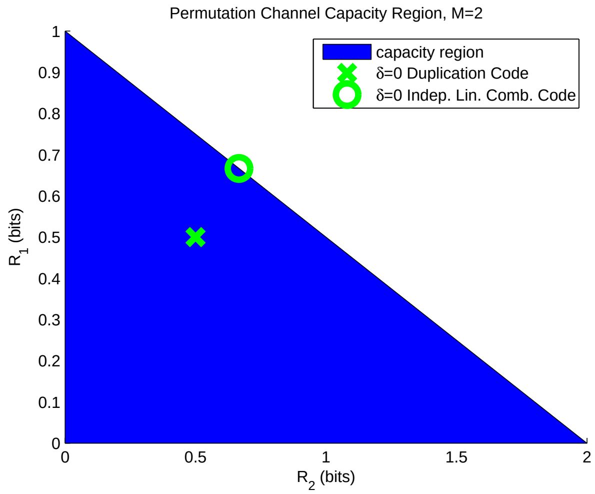

Fig. 4. The rate region of the permutation degraded broadcast channel for $M = 2$, and the rate points achieved by the duplication and independent linear combination zero delay codes. Surprisingly, the zero delay independent linear combination codes are rate optimal, despite the fact that they only require a block length of one complete permutation broadcast channel use.

broadcast channel use. The codes we have specified are rate optimal for the uniform permutation channel because they time share efficient (MDS) erasure codes for the different time instants. These codes have the same properties for $M > 2$, and the rate delay tradeoff selects different points (other than the all rates equal) on the boundary of the corresponding degraded broadcast channel capacity region. Thus, the rate delay tradeoff provided may be interpreted as a programming problem on the capacity region of this broadcast channel.

VI. Conclusion and Extensions

We have introduced the permutation channel and discussed the performance of the independent linear combination (time shared MDS) code, both for finite and infinite block lengths. There are several open questions and extensions of these results that merit investigation. One extension [9], [10] of these results pursues the connection between network coding and the permutation channel: if a block of network coded packets are only decodable at the receiver when the entire block is received, then delay sensitive applications may be required to trade throughput to ensure the decoding delay (measured in number of received encoding packets) is sufficiently small for different portions of the original block. The permutation channel formulation may guide the design of hybrid delay–throughput codes that seek to obtain higher throughput (from network coding) but with minimal effect on delay (from permutation coding). A significant open question is to answer these questions for *general* permutation channels, i.e., with arbitrary distribution $\mathbf{p} = (p(\pi), \pi \in \Pi)$, instead of the assumed uniform distribution. Our proposed codes are designed to achieve a specified deterministic delay, and are therefore insensitive to the channel distribution, but a more efficient rate may be achieved if the delay requirement is relaxed to an expectation. This scenario is of particular practical interest as packet reordering in most networks is not uniformly likely over all orderings.

Appendix

Proof of Lemma 1

The decoding delay for each dataword when the identity code is used on the uniform permutation channel may be written as

$$T_m = \inf\{k : (X_1, \ldots, X_m) \subseteq (Y_1, \ldots, Y_k)\} \tag{16}$$

for $m = 1, \ldots, M$. Thus T_m is the first time that the first m datawords of the datablock are available at the receiver. Note that $\mathbb{P}(T_1 = k) = \frac{1}{K}$ for $k = 1, \ldots, M$, and thus $\mathbb{P}(T_1 \leq k) = \frac{k}{M}$, again for $k = 1, \ldots, M$. More generally,

$$\begin{aligned} \mathbb{P}(T_m = k) &= \frac{P(m,m)P(M-m,k-m)}{P(M,k)}\binom{k-1}{m-1} \\ &= \frac{m(k-1)!(M-m)!}{(k-m)!M!}, \end{aligned} \tag{17}$$

for $k = m, \ldots, M$. Here $P(k,m)$ is the number of permutations of k objects into m slots ($m \leq k$). Consider the event $\{T_m = k\}$. Label each time slot $l = 1, \ldots, k$ as good if $Y_l \in (X_1, \ldots, X_m)$, and bad else. There are m good times slots over the first k time slots and necessarily Y_k is good. The first term $P(m,m)$ accounts for all equally possible ordering of the good slots. The last slot is necessarily a good slot, but the other $m-1$ good slots are selected at random from the available $k-1$ slots; thus the $\binom{k-1}{m-1}$ accounts for the selection of the remaining $m-1$ good slots from among the remaining $k-1$ slots. The term $P(M-m, k-m)$ accounts for all possible orderings of the $M-m$ bad values among the bad slots $k-m$. Finally, the denominator $P(M,k)$ gives the number of possible orderings of the M possible values over the k slots. Note $T_m \geq m$ under the identity code, and thus the expected delay is

$$\begin{aligned} d(M) &= \mathbb{E}\left[\frac{1}{M}\sum_{m=1}^{M} D_m\right] && (18) \\ &= \frac{1}{M}\sum_{m=1}^{M}\mathbb{E}\left[\frac{2}{M}(T_m - m)^+\right] && (19) \\ &= \frac{2}{M^2}\sum_{m=1}^{M}\sum_{k=m}^{M}(k-m)\mathbb{P}(T_m = k). && (20) \end{aligned}$$

Substituting (17) into (18) yields

$$\begin{aligned} d(M) &= \frac{2}{M^2}\sum_{m=1}^{M}\sum_{k=m}^{M}(k-m)\frac{m(k-1)!(M-m)!}{(k-m)!M!} \\ &= \frac{2}{M^2}\sum_{m=1}^{M}\frac{m(M-m)}{m+1} \\ &= \frac{(M+1)(M+2-2H(M+1))}{M^2}. \end{aligned}$$

■

Proof of Theorem 1

Just as we restricted the feasible set of (10) from $\{1,\ldots,M\}^M$ to $\mathcal{R}$, we first restrict the feasible set of (12). Define $\mathcal{D} = \{x : x \in \mathcal{C}, x'(t) \geq 0,\ t \in [0,1]\}$ as the set of continuous functions with non-negative slope; this is analogous to the restriction of $\{1,\ldots,M\}^M$ to tuples satisfying $1 \leq m_1 \leq \cdots \leq m_M \leq M$. Let x^* be an extremum of (12) over the set $\mathcal{C}$. Claim: $x^* \in \mathcal{D}$. Proof: suppose $x^* \in \mathcal{C} \setminus \mathcal{D}$. Let $\{(s_1,t_1),\ldots,(s_L,t_L)\}$ denote the set of intervals for which $x^{*'}(t) < 0$. For each interval (s_i,t_i) define $\tau_i = \inf\{t : x^*(t) \geq x^*(s_i)\}$, setting $\tau_i = 1$ if no such t exists. Form the function $\tilde{x}$ by setting $\tilde{x}(t) = x(s_i)$ for $t \in [s_i,\tau_i]$, otherwise set $\tilde{x}(t) = x(t)$. Note that $D(x^*) = D(\tilde{x})$, and $R(x)^{-1} > R(\tilde{x})^{-1}$. This contradicts the claim that an optimal $x^* \in \mathcal{C} \setminus \mathcal{D}$.

Define $\mathcal{E} = \{x : x \in \mathcal{C}, x(t) \geq t,\ t \in [0,1]\}$ as the set of continuous functions not falling below the line $x = t$; this is analogous to the restriction of $\{1,\ldots,M\}^M$ to tuples satisfying $m_k \geq k$. Let x^* be an extremum of (12) over the set $\mathcal{C}$. Claim: $x^* \in \mathcal{E}$. Proof: suppose $x^* \in \mathcal{C} \setminus \mathcal{E}$. Form the function $\tilde{x}$ as $\tilde{x}(t) = x^*(t) \vee t$. Note that $D(x^*) = D(\tilde{x})$, and $R(x^*)^{-1} > R(\tilde{x})^{-1}$. This contradicts the claim that an optimal $x^* \in \mathcal{C} \setminus \mathcal{E}$.

It follows that $x^* \in \mathcal{C} \cap \mathcal{D} \cap \mathcal{E}$; within this restricted set the delay function may be simplified to

$$D(x) = 2\int_0^1 (x(t)-t)\,\mathrm{d}t, \quad x \in \mathcal{C} \cap \mathcal{D} \cap \mathcal{E}. \tag{21}$$

Inverting the objective function to minimize the inverse rate, we combine the above observations to transform (12) into the following:

$$\min_{x\in\mathcal{C}}\left\{\int_0^1 \frac{1}{x(t)}\mathrm{d}t \;\middle|\; \begin{array}{c} \int_0^1 (x(t)-t)\,\mathrm{d}t = \frac{\delta}{2}, \\ x(t) \geq t,\ x'(t) \geq 0,\ t \in [0,1] \end{array}\right\}. \tag{22}$$

Form the Lagrangian with Lagrange multipliers λ, $\mu = \{\mu(t)\}$, and $\nu = \{\nu(t)\}$:

$$\begin{aligned} \mathcal{L} &= \int_0^1 \frac{1}{x(t)}\mathrm{d}t + \lambda\left(\int_0^1 (x(t)-t)\,\mathrm{d}t - \frac{\delta}{2}\right) \\ &\quad + \int_0^1 \mu(t)(x(t)-t)\mathrm{d}t + \int_0^1 \nu(t)x'(t)\mathrm{d}t \end{aligned}$$

The Euler-Lagrange equation giving a necessary condition for x to be an extremum of this functional is

$$\begin{aligned} &\frac{-1}{x(t)^2} + \lambda + \mu(t)\mathbf{1}_{x(t)=t} = \nu'(t)\mathbf{1}_{x'(t)=0} \iff \\ &x(t) = \left(\lambda + \mu(t)\mathbf{1}_{x(t)=t} - \nu'(t)\mathbf{1}_{x'(t)=0}\right)^{-\frac{1}{2}}. \end{aligned}$$

Note that $x(t)$ only depends upon t if either $x(t) = t$ or $x'(t) = 0$. But this specifies $x(t)$ to be either t or a constant respectively. Otherwise, $x(t)$ is a constant. Given that $x \in \mathcal{C} \cap \mathcal{D} \cap \mathcal{E}$ we conclude that

$$x^*(t) = \begin{cases} c, & 0 \leq t \leq c \\ t, & c < t \leq 1 \end{cases}. \tag{23}$$

for a c that may be computed through the delay constraint:

$$\int_0^c (c-t)\,\mathrm{d}t = \frac{\delta}{2} \iff c = \sqrt{\delta},\ c \in [0,1]. \tag{24}$$

Using the optimal x^* the associated maximum rate is

$$R(x^*) = \left(\int_0^{\sqrt{\delta}} \frac{1}{\sqrt{\delta}}\mathrm{d}t + \int_{\sqrt{\delta}}^1 \frac{1}{t}\mathrm{d}t\right)^{-1} = \frac{1}{1-\log\sqrt{\delta}}.$$ ■

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[2] T. M. Cover, "Broadcast channels," *IEEE Transactions on Information Theory*, vol. IT-18, no. 1, pp. 2–14, January 1972.

[3] ——, "Comments on broadcast channels," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2524–2530, October 1998.

[4] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Transactions on Information Theory*, vol. 42, no. 6, November 1996.

[5] S. Boucheron and M. R. Salamatian, "About priority encoding transmission," *IEEE Transactions on Information Theory*, vol. 46, no. 2, March 2000.

[6] M. G. Luby, M. Mitzenmacher, A. Shokrollahi, and D. A. Spielman, "Efficient erasure correcting codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, February 2001.

[7] R. Urbanke and A. Wyner, "Packetizing for the erasure broadcast channel with an internet application," 1997. [Online]. Available: citeseer.ist.psu.edu/urbanke97packetizing.html

[8] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, October 2006.

[9] J. M. Walsh and S. Weber, "A Concatenated Network Coding Scheme for Multimedia Transmission," in *Fourth Workshop on Network Coding, Theory, and Applications (Netcod 2008)*, 2008, submitted.

[10] D. Silva and F. R. Kschischang, "Rank-Metric Codes for Priority Encoding Transmission with Network Coding," in *Canadian Workshop on Information Theory*, Edmonton, AB, Canada, June 2007, p. 8184.