Universal Codes For Minimizing Per-User Delay on Streaming Broadcast Channels

Emin Martinian, Gregory W. Wornell
Dept. of Electrical Engineering
Massachusetts Institute Of Technology
Cambridge, MA 02139
{emin,gww}@allegro.mit.edu

Abstract

We study the fundamental limits of decoding delay in providing reliable communication for real-time applications. After presenting a simple example to illustrate the difficulties in using traditional block codes, we propose a new communication model to study decoding delay for streaming broadcast channels. We prove a coding theorem establishing that, for certain collections of channel ensembles, delay-universal codes exist which simultaneously achieve the best delay for any channel in the collection. Such codes can reduce the latency in video conferencing, tele-medicine, remote monitoring, control, and related scenarios.

1 Introduction

Real-time applications such as voice and video conferencing, tele-medicine, remote monitoring, and control are highly sensitive to delay. Traditional methods of providing reliable communication usually rely on powerful error correcting codes, interleaving, or other forms of diversity to achieve robustness. While such traditional methods are highly suited to minimizing the resources required to deliver long, non-causal messages (*e.g.*, file transfers), they are not necessarily suitable for minimizing delay in streaming applications.

Consider the difference in transmitting a file versus broadcasting a lecture in real-time. One portion of a file may be useless without the rest. Hence, in sending a file, delay is naturally measured as the time between when transmission starts and ends. Performance for file transfers can thus be measured as a function of the total transmission time *e.g.*, as studied in the analysis of error exponents.

By contrast, a lecture is intended to be viewed or heard sequentially. Hence, delay is naturally measured as the lag between when the lecturer speaks and the listener hears. The minimum, maximum, or average lag is thus a more appropriate measure than the total transmission time. Furthermore, the best possible lag may be different for each user, depending on the received channel quality. Even for a single user, the best possible lag may vary throughout the transmission as the channel varies. Although traditional codes may be applied to real-time systems, they may be far from optimal.

Our goal is to compute the best possible lags and design systems which achieve them. Essentially, we wish to study transmission of a single message which can be received over a collection of channel ensembles denoted by $\{\Theta_i\}$ and defined in detail in Section 3. The decoding delay will of course depend upon the particular channel conditions encountered. But, codes which minimize delay for Θ may be poor for Θ'. Therefore, achieving good overall performance depends on designing codes which perform well for each $\Theta \in \{\Theta_i\}$. For example, in a multicast scenario, user i may receive the signal over channel ensemble Θ_i. Similarly, even with only a single receiver, there may be some uncertainty about the channel or the channel may act differently at different times and each Θ_i may correspond to one such possible channel. Ideally, we would like a system which achieves the minimum possible decoding delay for each possible channel condition.

To illustrate the shortcomings of using traditional block codes, consider a packet loss channel which behaves in one of two possible modes. In the first mode, denoted Θ_1, no more than 1 packet is lost within a given window of time. In the second mode, Θ_2, up to 3 packets may be lost in the time window of interest.

With a traditional block code, the transmitter could encode each group of 9 source packets, $\mathbf{s}[0]$, $\mathbf{s}[1]$, …, $\mathbf{s}[8]$ into 12 coded packets, $\mathbf{x}[0]$, …, $\mathbf{x}[11]$ using a systematic $(12, 9)$ Reed-Solomon (RS) code (or any other equivalent Maximum Distance Separable code). With this approach, any pattern of 3 (or less) packet losses occurring for channel Θ_2 can be corrected. But, on channel Θ_1, if only $\mathbf{x}[0]$ is lost, the soonest it can be recovered is when 9 more coded packets are received. Evidently the system may incur a decoding delay of 9 packets just to correct a single packet loss.

To decrease the delay, instead of using a $(12, 9)$ code, each block of 3 source packets could be encoded into 4 coded packets using a $(4, 3)$ RS code. Since one redundant packet is generated for every three source packets, this approach requires the same redundancy as the $(12, 9)$ code. With the $(4, 3)$ code, however, if the channel is in mode Θ_1 and only $\mathbf{x}[0]$ is lost, it can be recovered after the remaining 3 packets in the block are received. Thus the $(4, 3)$ system incurs a delay of only 3 to correct one lost packet. While this delay is much smaller than with the $(12, 9)$ code, if more than one packet in a block is lost on channel Θ_2, then decoding is impossible with the $(4, 3)$ code.

Both practical block codes as well as traditional information theoretic arguments are not designed for real-time systems. Thus we see that minimizing delay for minor losses from Θ_1 and maximizing robustness for major losses from Θ_2 are conflicting objectives. Is this trade-off fundamental to the nature of the problem, or is it an artifact of choosing a poor code structure? We show that in many cases of practical interest, there exist better code structures which are universally optimal for all channel conditions. Specifically, for the packet loss example, there exist codes with both the low decoding delay of the $(4, 3)$ code and the robustness of the $(12, 9)$ code. After discussing related work in Section 2, we motivate and define a new system model in Section 3 and present our main results on the existence of such codes in Section 4.

2 Previous Work

Researchers have explored channel uncertainty and transmission to multiple users by studying the broadcast channel, compound channel, and arbitrarily varying channel, [1] [2]. In these problems, each Θ_i represents a different channel probability law and the focus is on finding the capacity region. Usually there is a trade-off and increasing the rate received for a particular channel, Θ_i, requires decreasing the rate received over a different channel, $\Theta_{i'}$. In certain cases, however, universal systems exist [3].

In contrast to these static models where the goal is to find various capacity regions, we consider a dynamic model and study the delays or lags. Since we analyze the intramessage delays appropriate for real-time applications as opposed to the overall message delay appropriate for non real-time applications, we require a communication model where small chunks (or packets) of information are generated, sent, received, and decoded. We let each Θ_i represent an ensemble of channel conditions, and study the lag between when packets are generated and decoded.

An alternative model also concerned with delays in a somewhat different setting has been recently studied by Sahai [4]. Also, Shulman and Feder have considered a static broadcast channel where multiple receivers listen for a single long message (*e.g.*, a file transfer) [5]. They derive the trade-off between decreasing total delay for one receiver at the cost of increasing delay for another by studying the delay region. Similarly, Luby *et al.* construct codes to allow transmission of a file over a packet network such as the Internet [6] [7]. Once a number of bits corresponding to the total length of the file have been received, their codes allow the receiver to successfully recover the file with low computational delay. Techniques from this work may also prove useful for real-time applications (especially in the construction of practical codes).

3 Stream Coding System Model

We consider a streaming model where at each time step i a new source packet $\mathbf{s}[i]$ is revealed to the transmitter and encoded into a channel packet $\mathbf{x}[i]$. Each source packet consists of n_s samples from the alphabet $\mathcal{S}$ and each channel packet consists of n_c samples from the alphabet $\mathcal{X}$. A memory $\mathtt{M}$ packet encoder, $\mathcal{C}$, consists of a causal mapping from the past $\mathtt{M}$ source packets, $\mathbf{s}[i-\mathtt{M}]$, $\mathbf{s}[i-\mathtt{M}+1]$, $\ldots$, $\mathbf{s}[i-1]$ and the current source packet $\mathbf{s}[i]$ into the current channel input packet $\mathbf{x}[i]$:

$$\mathcal{C} : (\mathcal{S}^{n_s})^{\mathtt{M}+1} \mapsto \mathcal{X}^{n_c}. \tag{1}$$

For ease of analysis, we allow $\mathtt{M}$ and n_c to be as large as required. We define the rate of the system as

$$\mathtt{R} = (n_s/n_c) \log |\mathcal{S}|. \tag{2}$$

where $|\cdot|$ denotes cardinality of a set. Throughout the paper log refers to the natural logarithm and thus rate is measured in nats.

To study systems where the channel state changes relatively slowly we use a piece-wise constant or block-interference model [8] [9] [10]. Specifically, the channel output packet, $\mathbf{y}[i]$, obtained by the receiver is determined according to the channel law

$$\prod_{i=0}^{\infty} p_{\mathbf{y}[i]|\mathbf{x}[i];\boldsymbol{\theta}[i]}(\mathbf{y}[i]|\mathbf{x}[i];\theta[i]) = \prod_{i=0}^{\infty}\prod_{j=1}^{n_c} p_{y|x;\boldsymbol{\theta}}(y_j[i]|x_j[i];\theta[i]).$$

where $\boldsymbol{\theta}[i]$ denotes the channel state for packet i.

A packet decoder, $\mathcal{C}^{-1}$, is a mapping from a delayed sequence of channel outputs $\mathbf{y}[i+\mathtt{T}]$, $\mathbf{y}[i+\mathtt{T}-1]$, $\ldots$, and the corresponding channel states $\boldsymbol{\theta}[i+\mathtt{T}]$, $\boldsymbol{\theta}[i+\mathtt{T}-1]$, $\ldots$, to an estimate of a source packet, $\mathbf{s}[i]$. For a particular channel state sequence, θ, an encoder, $\mathcal{C}$, and the associated decoder $\mathcal{C}^{-1}$, we define the decoding delay, $\mathbf{T}(\theta, \mathcal{C}, n_c, \epsilon)$, as the minimum lag which guarantees successful decoding with probability of error at most ϵ when using packets of length n_c:

$$\mathbf{T}(\theta, \mathcal{C}, n_c, \epsilon) \overset{\Delta}{=} \min_{\mathtt{T}} : \left\{\Pr\left[\mathcal{C}^{-1}\left(\mathbf{y}\left[{}^{i+\mathtt{T}}_{-\infty}\right], \boldsymbol{\theta}\left[{}^{i+\mathtt{T}}_{-\infty}\right]\right) \neq \mathbf{s}[i]\right] \leq \epsilon\right\} \tag{3}$$

where we use the notation $\mathbf{a}\left[{}^{j}_{i}\right]$ to denote the subsequence $\mathbf{a}[i]$, $\mathbf{a}[i+1]$, $\ldots$, $\mathbf{a}[j]$.

3.1 The Achievable Delay Region

In a broadcast or multicast scenario, different users receive the signal over different channel conditions. Similarly, even with only a single receiver, there may be some uncertainty about the channel. Traditionally in a static problem, this is modeled by defining a different channel probability law for each user or each possible channel. Instead, to obtain a dynamic model, we define a channel ensemble Θ_i as a set of possible channel state sequences, $\boldsymbol{\theta}[i]$. The packet loss channel in the introduction provides one example of such a model. For another example, the channel ensemble Θ_1 could consist of all channel state sequences where five of the received packets have a signal-to-noise ratio up to 10 dB below nominal. Similarly, the channel ensemble Θ_2 could consist of all sequences where seven of the received packets have a signal-to-noise ratio up to 3 dB below nominal.

One natural performance measure for a code $\mathcal{C}$ on the channel ensemble Θ is the maximum decoding delay for $\mathcal{C}$ on any state sequence in Θ:

$$\mathbf{T}_{\max}(\Theta, \mathcal{C}, n_c, \epsilon) \triangleq \max_{\theta \in \Theta} \mathbf{T}(\theta, \mathcal{C}, n_c, \epsilon). \tag{4}$$

To model scenarios where the channel may behave in one of N distinct modes (*e.g.*, N users each receiving the transmission over different physical conditions), we can consider a collection of N channel ensembles $\{\Theta_1, \Theta_2, \ldots, \Theta_N\}$. In this case, performance can be measured by the delay tuple $(\mathtt{T}_1, \mathtt{T}_2, \ldots, \mathtt{T}_N)$ where $\mathtt{T}_i = \mathbf{T}_{\max}(\Theta_i, \mathcal{C}, n_c, \epsilon)$.

We define the set of all possible delay tuples as the achievable delay region, as illustrated in Fig. 1 for $N = 2$. Determining the N-dimensional achievable delay region for arbitrary channel ensembles would provide the most complete understanding of delay in the packet-streaming model. Codes which achieve the best delay trade-offs characterized by such a region would be useful building blocks in designing practical systems. In this paper, we focus on simpler performance measures corresponding to particular delay tuples and defer more detailed exploration of delay regions [11].

To obtain a simple performance measure, we need to map the collection of channel ensembles $\{\Theta_i\}_{i=1}^{N}$ to a single measure of overall delay efficiency. How should we construct this mapping? First, the measure should not completely ignore performance for any channel ensemble Θ_i. Second, the measure should lead to the design of codes which are good for a large collection of channel conditions. Finally, the measure should be reasonably simple to analyze.

Worst-case performance would seem to be a natural metric. Specifically, define the minimax-delay for a channel ensemble Θ as the worst-case delay for the best code:

$$\mathbf{T}_{\max}^{\min}(\Theta, n_c, \epsilon) \triangleq \min_{\mathcal{C}} \max_{\theta \in \Theta} \mathbf{T}(\theta, \mathcal{C}, n_c, \epsilon). \tag{5}$$

The codes labeled $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_{12}$ in Fig. 1 correspond to points achieving the minimax-delays $\mathbf{T}_{\max}^{\min}(\Theta_1, n_c, \epsilon)$, $\mathbf{T}_{\max}^{\min}(\Theta_2, n_c, \epsilon)$ and $\mathbf{T}_{\max}^{\min}(\Theta_1 \cup \Theta_2, n_c, \epsilon)$ respectively.

The drawback of using $\mathbf{T}_{\max}^{\min}(\Theta_i, n_c, \epsilon)$ as a performance measure is that it completely ignores performance on all channel ensembles except Θ_i. The drawback of using $\mathbf{T}_{\max}^{\min}\left(\cup_{i=1}^{N}\Theta_i, n_c, \epsilon\right)$ is that this quantity is essentially determined by the worst channel in the collection $\cup_{i=1}^{N}\Theta_i$. Hence codes designed to optimize $\mathbf{T}_{\max}^{\min}\left(\cup_{i=1}^{N}\Theta_i, n_c, \epsilon\right)$ will not necessarily perform well on most of the ensembles in the collection.

To illustrate the drawback of worst-case delay for a concrete example, consider a congestion model where packet losses may occur. Let each Θ_i represent the ensemble of packet loss patterns with exactly i lost packets. The worst-case delay will always be

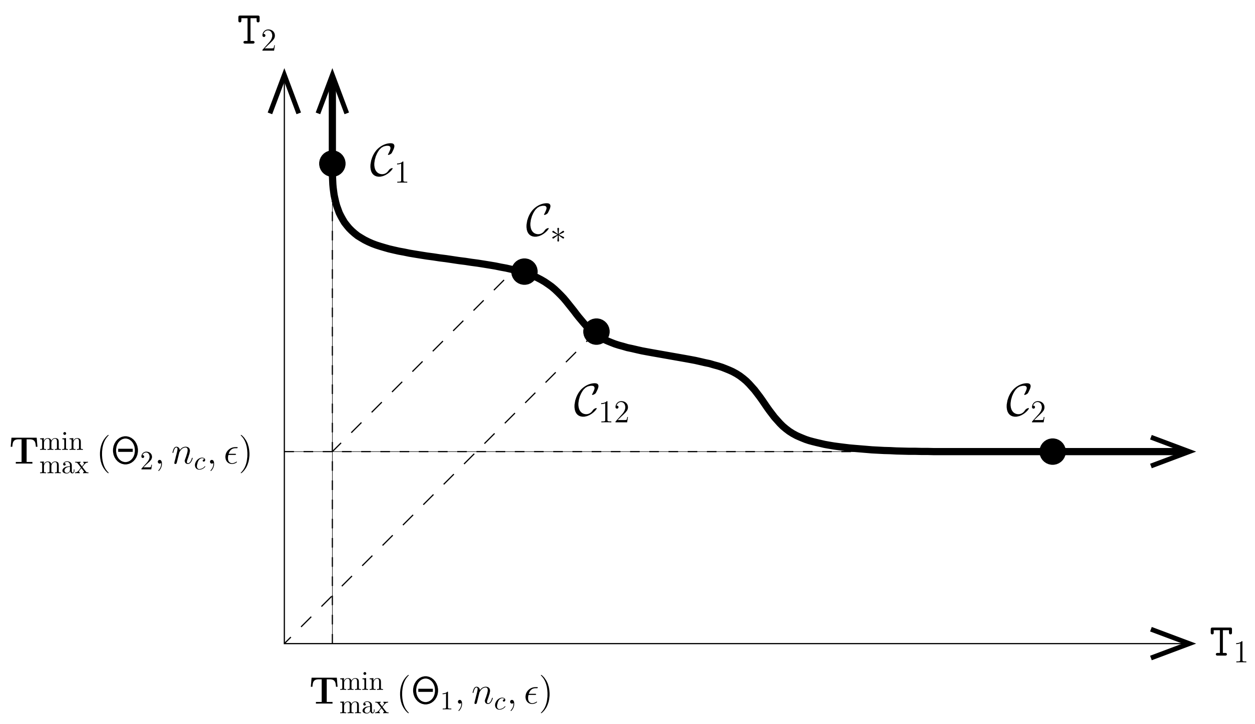

Figure 1: A Possible delay region. The axes represent delay for two channels ensembles Θ_1 and Θ_2. Points above and to the right of the solid curve correspond to achievable delays. The points marked $\mathcal{C}_1$ and $\mathcal{C}_2$ correspond to codes achieving the minimum delay for Θ_1 and Θ_2, while the code $\mathcal{C}_{12}$ minimizes the maximum delay. The code $\mathcal{C}_*$ minimizes the maximum difference between the delay for a fixed code and the best code for each channel ensemble.

determined by a packet loss pattern consisting of the maximum number of lost packets, *i.e.*,

$$\max_{\theta \in \cup_{i=1}^{N} \Theta_i} \mathbf{T}(\theta, \mathcal{C}, n_c, \epsilon) = \max_{\theta \in \Theta_N} \mathbf{T}(\theta, \mathcal{C}, n_c, \epsilon). \tag{6}$$

Hence $\mathbf{T}_{\max}^{\min}\left(\cup_{i=1}^{N}\Theta_i, n_c, \epsilon\right)$ completely ignores performance for every channel ensemble except Θ_N and only focuses on the worst channel conditions. Specifically, the worst-case delay does not consider performance when fewer losses occur: if only $N/2$ packet losses occur, then $\mathcal{C}$ could have a much lower decoding delay than $\mathcal{C}'$, but this is not captured by our performance metric. Ideally, we desire a code which can correct the loss of fewer packets with correspondingly shorter delay.

Intuitively, we would like a "universal" code which has the same delay as an optimal code designed for $\mathtt{B}$ packet losses when $\mathtt{B}$ packet losses occur and has the same delay as an optimal code designed for $\mathtt{B}'$ packet losses when $\mathtt{B}'$ occur. Thus, we believe that a valuable performance measure for a code is the codes deviation from being universal. We define the excess delay $\mathbf{T}_{\text{ex}}(\mathcal{C}, \{\Theta_i\})$ of a code $\mathcal{C}$ as how much more delay $\mathcal{C}$ requires for each ensemble than a minimax delay optimal code designed specifically for that ensemble:

$$\mathbf{T}_{\text{ex}}(\mathcal{C}, \{\Theta_i\}, n_c, \epsilon) \triangleq \max_i \left[\max_{\theta \in \Theta_i} \mathbf{T}(\theta, \mathcal{C}, n_c, \epsilon) - \mathbf{T}_{\max}^{\min}(\Theta_i, n_c, \epsilon)\right]. \tag{7}$$

The code $\mathcal{C}_*$ in Fig. 1 is an example of a code which minimizes the excess delay. While a code minimizing $\mathbf{T}_{\max}^{\min}\left(\cup_{i=1}^{N}\Theta_i, n_c, \epsilon\right)$ is found by drawing a 45 degree line from the origin to the frontier of the achievable delay region, a code minimizing $\mathbf{T}_{\text{ex}}\left(\mathcal{C},\{\Theta_i\},n_c,\epsilon\right)$ can be found by drawing a 45 degree line from the intersection of the $\mathbf{T}_{\max}^{\min}\left(\Theta_i,n_c,\epsilon\right)$ hyper-planes to the frontier of the achievable region.

If the excess delay for a code $\mathcal{C}$ is zero, we call $\mathcal{C}$ delay-universal over the collection of channel ensembles $\{\Theta_i\}$. Of course, whether such a universal code exists depends upon the collection of channel ensembles. In the sequel, we show that in many practical cases of interest such codes do exist and lead to systems which are good for a variety of channel conditions.

4 Coding Theorems

In order to state our main result about the existence of delay-universal codes, we need the following definitions.

Definition 1. *A collection of channel ensembles $\{\Theta_i\}$ is defined as* **input compatible** *if there exists an input probability distribution which simultaneously maximizes the mutual information $I(\mathsf{x};\mathsf{y}|\theta[i])$ for every state $\theta[i]$ in every sequence $\theta \in \Theta_i$.*

A collection of channel ensembles containing additive white Gaussian noise (AWGN) channels with different signal-to-noise ratios is an example of an input compatible collection while one containing both additive white Gaussian noise channels and additive exponential noise channels is not.[1] Intuitively, this property is important if universal codes are desired. Otherwise a code matched to one channel ensemble will have a sub-optimal mutual information on another channel ensemble and hence fail to be universally optimal. For non input compatible collections, techniques from the theory of compound channels and broadcast channels will be required.

Definition 2. *A channel ensemble Θ is defined as* **permutation invariant** *if, for every state sequence $\theta \in \Theta$, every permutation of θ is also in Θ.*

A channel ensemble where any `B` packets may be completely lost while the others are received error free and an ensemble where any `B` packets may be received with half the normal signal-to-noise ratio are both permutation invariant. A channel ensemble where any consecutive `B` packets may be completely lost or an ensemble where any consecutive `B` packets may be received with half the normal signal-to-noise ratio are not permutation invariant.

Our main result, proved on page 10, is the following theorem:

Theorem 1. *If $\{\Theta_i\}$ is a collection of input compatible, permutation invariant channel ensembles, then there exists a delay-universal code for $\{\Theta_i\}$. Specifically, there exists a code, $\mathcal{C}$, such that for every $\epsilon > 0$, the excess delay is exactly zero for large enough packets:*

$$\forall n_c > n_0(\epsilon),\ \mathbf{T}_{\text{ex}}\left(\mathcal{C},\{\Theta_i\},n_c,\epsilon\right) = 0. \tag{8}$$

[1]AWGN channels are also stochastically degraded [12]. In general, stochastically degraded channels are not necessarily input compatible, and input compatible channels are not necessarily stochastically degraded.

A Packet Loss Example To apply this result to streaming in a packet network, we can choose Θ_i as the ensemble where at most i packets are lost. For any N, $\{\Theta_i\}_{i=1}^N$ is a collection of input compatible, permutation invariant channel ensembles and therefore Theorem 1 guarantees the existence of a universal code. In practice the minimum packet size, $n_0(\epsilon)$, need not be excessively large and codes requiring packet sizes proportional to $N/\log\epsilon$ with encoding and decoding times polynomial in the packet size can be easily constructed [11]. Such lengths are well within the packet sizes used for Internet transmission as well as various wireless standards. For arbitrary channel models, tools from the traditional analysis of error exponents can be used to bound the required packet size.

The construction of the desired code is straightforward and uses standard random coding arguments. The key variation is that the code is designed to work in a packet-streaming model instead of a block model. The main effort required to prove the theorem is showing that no code designed for a specific Θ_i can do better. To show this we need a tool to study the decoding delay of a code as a function of the channel mutual information.

4.1 Information Debt

We define the mutual information debt at time i, $I_d[i, \mathtt{R}|\theta]$, according to the recursion

$$I_d[i, \mathtt{R}|\theta] = n_c \cdot \mathtt{R} - I(\mathbf{x}[i]\,; \mathbf{y}[i]\,|\theta[i]) + \max\{I_d[i-1, \mathtt{R}|\theta], 0\} \tag{9}$$

with the added condition that $I_d[i_{\text{start}}, \mathtt{R}|\theta] = 0$ where i_{start} is the time of the initial transmission. The information debt can be thought of as how much more information we need to receive about the current messages before successful decoding can occur. This interpretation can be immediately translated into the following simple bound on decoding delay.

Lemma 1. *Consider a channel state sequence where the mutual information debt becomes positive at time i_s and stays positive until at least time i_f. If N is the largest integer below $I_d[i_f, \mathtt{R}|\theta]/(n_c\mathtt{R})$, then at least one of the packets $\mathbf{s}[i_s]$, $\mathbf{s}[i_s+1]$, $\ldots$, $\mathbf{s}[i_f-N]$, transmitted by a causal, rate $\mathtt{R}$ system will fail to be decoded by time i_f.*

Proof. We have the following chain of inequalities:

$$\begin{aligned}
N \cdot n_c \cdot \mathtt{R} &< I_d[i_f, \mathtt{R}|\theta] && (10)\\
&= \sum_{i=i_s}^{i_f} [n_c\mathtt{R} - I(\mathbf{x}[i]\,; \mathbf{y}[i]\,|\boldsymbol{\theta}[i])] && (11)\\
&= (i_f - i_s + 1)n_c\mathtt{R} - I\left(\mathbf{x}\left[{}^{i_f}_{i_s}\right]; \mathbf{y}\left[{}^{i_f}_{i_s}\right] |\boldsymbol{\theta}\left[{}^{i_f}_{i_s}\right]\right) && (12)\\
&= (i_f - i_s + 1)n_c\mathtt{R} - I\left(\mathbf{x}\left[{}^{i_f}_{i_s}\right]; \mathbf{y}\left[{}^{i_f}_{-\infty}\right] |\boldsymbol{\theta}\left[{}^{i_f}_{-\infty}\right]\right) && (13)\\
&\le (i_f - i_s + 1)n_c\mathtt{R} - I\left(\mathbf{s}\left[{}^{i_f}_{i_s}\right]; \mathbf{y}\left[{}^{i_f}_{-\infty}\right] |\boldsymbol{\theta}\left[{}^{i_f}_{-\infty}\right]\right) && (14)\\
&\le (i_f - i_s + 1)n_c\mathtt{R} - I\left(\mathbf{s}\left[{}^{i_f-N}_{i_s}\right]; \mathbf{y}\left[{}^{i_f}_{-\infty}\right] |\boldsymbol{\theta}\left[{}^{i_f}_{-\infty}\right]\right) && (15)\\
I\left(\mathbf{s}\left[{}^{i_f-N}_{i_s}\right]; \mathbf{y}\left[{}^{i_f}_{-\infty}\right] |\boldsymbol{\theta}\left[{}^{i_f}_{-\infty}\right]\right) &< (i_f - i_s - N + 1)n_c\mathtt{R}. && (16)
\end{aligned}$$

Equations (10) and (11) follow from the assumption that $I_d[i, \mathtt{R}|\theta] > n_c\mathtt{R}N$ for $i_s \le i \le i_f$. The chain rule for mutual information and the block memoryless structure of the channel yield (12) and (13). Equation (14) follows from the data processing inequality and (15) follows from the chain rule for mutual information.

Finally, we can consider $\mathbf{s}[i_s]$, $\mathbf{s}[i_s+1]$, …, $\mathbf{s}[i_f-N]$ as a message of rate $(i_f - i_s - N + 1)\cdot \mathtt{R}$. Therefore Fano's inequality combined with (16) implies that the decoder can not recover this message with small probability of error. $\square$

Note that Lemma 1 is a somewhat weak characterization of the decoding delay. Specifically, it tells us that as long as the information debt is positive at least *some* source packet can not be decoded, but it does not tell us exactly *which* packet can not be decoded. For example, imagine that a number of packets are lost starting at time i_s and thus the information debt becomes positive at time i_s and stays positive until time i_f. If the encoder somehow anticipated the packet losses it may be possible that at time i_f, the decoder can reconstruct all packets except the one at time $i_f - N$ and hence the decoding delay may be only N instead of $i_f - i_s$.

Thus, in general, it is possible for the decoding delay required by a code to be less than the time the information debt is positive (*e.g.*, with burst erasure correction [13]). For a permutation invariant channel ensemble, however, this is not the case.

Lemma 2. *For any rate* $\mathtt{R}$ *code,* $\mathcal{C}$*, and permutation invariant channel ensemble,* Θ*, the worst-case decoding delay is at least the maximum possible time the information debt can be positive:*

$$\max_{\theta\in\Theta}\min_{\mathcal{C}} \mathtt{T}(\theta,\mathcal{C}) \geq \max_{\theta\in\Theta}\sum_{i=-\infty}^{\infty}\Big[I_d[i,\mathtt{R}|\theta]\Big]_1 \tag{17}$$

where $\Big[t\Big]_1$ *is 1 for* $t>0$ *and 0 otherwise.*

Due to space constraints, we provide a sketch of the argument for Lemma 2 only for packet loss channels. The full proof for arbitrary channels is available in [11].

Proof sketch for Lemma 2 on packet loss channels: Let Θ consist of all channel state sequences where exactly $\mathtt{B}$ packets may be completely lost and all others are correctly received. Assume for the sake of contradiction that Lemma 2 is false and the decoding delay for some code $\mathcal{C}$ is strictly less than the right hand side of (17) which we denote as $\mathtt{T}$. Consider the channel state sequence

$$\theta^* = \theta^*[0], \theta^*[1], \ldots, \theta^*[\mathtt{T}-1] = 1^{\mathtt{B}}0^{\mathtt{T}-\mathtt{B}}$$

where $\theta^*[i]=1$ indicates packet i was erased.

By assumption, $\mathbf{s}[0]$ can be recovered by time $\mathtt{T}-1$ even though the information debt is positive at time $\mathtt{T}-1$. Now imagine that $\theta^*[\mathtt{T}]$ is erased. Since, $\mathbf{s}[0]$ is already recovered, the decoding delay for θ^* is the same as for

$$\theta^{**} = \theta^{**}[0], \theta^{**}[1], \ldots, \theta^{**}[\mathtt{T}] = 01^{\mathtt{B}-1}0^{\mathtt{T}-\mathtt{B}}1.$$

But since Θ is permutation invariant $\theta^{**}\in\Theta$ and hence, by assumption, for θ^{**}, $\mathbf{s}[1]$ can be decoded at time $\mathtt{T}$. Since the decoding delay for θ^* must be no worse than for, θ^{**}, $\mathbf{s}[1]$ can be decoded at time $\mathtt{T}$ for θ^* even though

$$I_d[\mathtt{T},\mathtt{R}|\theta^*] > I_d[\mathtt{T}-1,\mathtt{R}|\theta^*] > 0.$$

By continuing this argument, we can construct a channel state sequence with information debt greater than $\mathtt{T}\cdot n_c\cdot\mathtt{R}$ at time i^*, which, by assumption, can be decoded with delay $\mathtt{T}-1$. This contradicts Lemma 1 and shows our assumption that Lemma 2 was false is incorrect. $\square$

4.2 Random Code Constructions

Lemma 2 provides a lower bound on the decoding delay via the information debt. We present a random code construction which results in successful decoding whenever the mutual information debt is non-positive. Together these results can be used to characterize the connection between information debt and delay.

Encoder Construction: To construct an encoder we require a random partition, $\mathcal{H}(\cdot)$, mapping arbitrary length sequences from the alphabet $\mathcal{S}^{n_s}$ to random i.i.d. sequences of length n_c from the alphabet $\mathcal{X}$. Specifically, $\mathcal{H}(\cdot)$ is generated according to the following procedure. For each element of the alphabet $\mathcal{S}^{n_s}$, independently select n_c elements from the alphabet $\mathcal{X}$ using the distribution $p(x)$. Repeat this procedure for each element of the alphabets $\mathcal{S}^{2n_s}$, $\mathcal{S}^{3n_s}$, etc.

Encoding and Decoding: The initial source packet, $\mathbf{s}[0]$, is encoded to form $\mathbf{x}[0] = \mathcal{H}(\mathbf{s}[0])$. The next source packet, $\mathbf{s}[1]$, is encoded to form $\mathbf{x}[1] = \mathcal{H}(\mathbf{s}[0], \mathbf{s}[1])$, the one after that is encoded to form $\mathbf{x}[2] = \mathcal{H}(\mathbf{s}[0], \mathbf{s}[1], \mathbf{s}[2])$, etc. For any $\epsilon > 0$, to decode $\mathbf{s}[i]$ (assuming previous packets have been already decoded), the decoder waits until $I_d[i+j, \mathtt{R}+\epsilon|\theta] \leq 0$. Then the decoder searches for a unique $\mathbf{s}[i]$ followed by a sequence of packets $\mathbf{s}[i+1]$, $\mathbf{s}[i+2]$, ..., $\mathbf{s}[i+j]$ such that ..., $\mathbf{x}[i]$, $\mathbf{x}[i+1]$, ..., $\mathbf{x}[i+j]$ are jointly typical with ..., $\mathbf{y}[i]$, $\mathbf{y}[i+1]$, ..., $\mathbf{y}[i+j]$. If no such unique $\mathbf{s}[i]$ is found, the decoder declares a decoding failure.

Probability of Error:

Lemma 3. *For any $\epsilon > 0$, there exists an n_0 such that for all $n_c > n_0$, the probability of decoding failure is less than ϵ.*

Proof. We prove the Lemma using standard arguments. Let $\mathcal{E}$ be the event that the decoder fails. The event $\mathcal{E}$ can be separated into the event that the transmitted codeword is not jointly typical with the received sequence (denoted by $\mathcal{E}_1$) and the event that an incorrect codeword is jointly typical with the received sequence (denoted by $\mathcal{E}_2$). The union bound implies $\Pr[\mathcal{E}] \leq \Pr[\mathcal{E}_1] + \Pr[\mathcal{E}_2]$. By the law of large numbers, $\Pr[\mathcal{E}_1] \to 0$ as $n_c \to \infty$, so all that remains is to bound $\Pr[\mathcal{E}_2]$.

Note that if $I_d[i+j, \mathtt{R}+\epsilon|\theta] \leq 0$, then according to (9),

$$I\left(\mathbf{x}\begin{bmatrix} i+j \\ 0 \end{bmatrix}; \mathbf{y}\begin{bmatrix} i+j \\ 0 \end{bmatrix} \middle| \boldsymbol{\theta}\begin{bmatrix} i+j \\ 0 \end{bmatrix}\right) \geq (i+j-1) \cdot n_c(\mathtt{R}+\epsilon). \tag{18}$$

This leads to the following chain of inequalities

$$\begin{aligned}
\Pr[\mathcal{E}] &\leq |\mathcal{S}|^{n_s \cdot (i+j-1)} \cdot \exp\left\{-I\left(\mathbf{x}\begin{bmatrix} i+j \\ 0 \end{bmatrix}; \mathbf{y}\begin{bmatrix} i+j \\ 0 \end{bmatrix} \middle| \boldsymbol{\theta}\begin{bmatrix} i+j \\ 0 \end{bmatrix}\right)\right\} &(19)\\
&= \exp\left\{(i+j-1) \cdot n_s \log|\mathcal{S}| - I\left(\mathbf{x}\begin{bmatrix} i+j \\ 0 \end{bmatrix}; \mathbf{y}\begin{bmatrix} i+j \\ 0 \end{bmatrix} \middle| \boldsymbol{\theta}\begin{bmatrix} i+j \\ 0 \end{bmatrix}\right)\right\} &(20)\\
&\leq \exp\left\{(i+j-1) \cdot \frac{n_s}{n_c} \cdot n_c \log|\mathcal{S}| - n_c \cdot (i+j-1)(\mathtt{R}+\epsilon)\right\} &(21)\\
&= \exp\left\{-n_c \cdot (i+j-1) \cdot \epsilon\right\}. &(22)
\end{aligned}$$

The right hand side of (19) consists of the number of possible codewords times a bound on the probability that any incorrect codeword is typical with the received sequence. We obtain (21) from (18), and (22) comes from our definition of rate (2). □

We can now use these Lemmas to prove Theorem 1.

proof of Theorem 1: Since the collection of channel ensembles is input compatible, the preceding code construction yields a code where the decoding delay is given by the length of time the information debt stays positive. According to Lemma 2, for any particular channel ensemble, Θ_i, the minimax decoding delay for any code is given by the maximum time the information debt can stay positive. Hence the preceding code construction is universal. $\square$

References

[1] T. M. Cover, "Comments on broadcast channels," *IEEE Transactions on Information Theory*, vol. 44, pp. 2524–2530, Oct 1998.

[2] A. Lapidoth and P. Narayan, "Reliable communication under channel uncertainty," *IEEE Transactions on Information Theory*, vol. 44, pp. 2148–2177, Oct 1998.

[3] A. Lapidoth and J. Ziv, "On the universality of the LZ-based decoding algorithm," *IEEE Transactions on Information Theory*, vol. 44, pp. 1746–1755, Sep 1998.

[4] A. Sahai, "Evaluating channels for control: capacity reconsidered," in *American Control Conference*, vol. 4, pp. 2358–2362, 2000.

[5] N. Shulman and M. Feder, "Static broadcasting," in *International Symposium on Information Theory*, p. 23, June 2000.

[6] M. Luby, "LT codes," *Foundations of Computer Science*, November 2002.

[7] J. W. Byers, M. Luby, and M. Mitzenmacher, "Accessing multiple mirror sites in parallel: using tornado codes to speed up downloads," *INFOCOM*, pp. 275–283, March 1999.

[8] R. McEliece and W. Stark, "Channels with block interference," *IEEE Transactions on Information Theory*, vol. 30, pp. 44–53, Jan 1984.

[9] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: information-theoretic and communications aspects," *IEEE Transactions on Information Theory*, vol. 44, pp. 2619–2692, October 1998.

[10] R. Knopp and P. A. Humblet, "On coding for block fading channels," *IEEE Transactions on Information Theory*, vol. 46, pp. 189–205, Jan 2000.

[11] E. Martinian. PhD thesis, Massachusetts Institute of Technology. in preparation.

[12] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* New York: John Wiley & Sons, Inc., 1991.

[13] E. Martinian and C.-E. Sundberg, "Low delay burst erasure correction codes," in *International Conference on Communications*, vol. 3, pp. 1736–1740, 2002.