ERASURE SOURCE-BROADCAST: INFORMATION-THEORETIC LIMITS

by

Louis Tan

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Graduate Department of Electrical and Computer Engineering
University of Toronto

© Copyright 2018 by Louis Tan
Abstract

Erasure Source-Broadcast: Information-Theoretic Limits

Louis Tan
Doctor of Philosophy
Graduate Department of Electrical and Computer Engineering
University of Toronto
2018

We study the information-theoretic limits of broadcasting a binary source over the erasure broadcast channel to $n$ receivers who have individual fractional distortion constraints. For the $n$-user problem without feedback, we derive an achievable coding scheme that is optimal under certain conditions. For the $n = 2$ and $n = 3$ user problem with feedback, we propose achievable coding schemes that are optimal for $n = 2$ and conjectured to be optimal for $n = 3$. In analyzing the $n = 3$ case, we also solve the more general problem of finding the expected accumulated reward before absorption of a Markov rewards process with impulse rewards and absorbing states. For the $n = 2$ case, we show that a feedback channel for both users is not necessary and that optimality can still be achieved if only the stronger user has a feedback channel. Finally, we derive an outer bound for the class of non-erasure-randomized codes for the case when $n = 2$. 
In memory of my father
Acknowledgements

I would like to first express my deep gratitude to my advisor, Professor Ashish Khisti, for his guidance and support throughout my academic career. It has been a great privilege to have had him as a mentor. He has always shown the utmost dedication to his students and committed his full time and effort in his desire to see his students develop. I have learned a great deal from Ashish, and I feel very fortunate to have had him as an advisor.

I would like to thank my committee members, Professors Baochun Li, Kostas Plataniotis, and Wei Yu for their help in evaluating and improving this thesis. I would also like to express my gratitude to my external examiner, Prof. Fady Alajaji, for dedicating his time and expertise in carefully reviewing and providing constructive feedback on this thesis.

During my studies, I have been fortunate enough to collaborate with many excellent researchers, and I would like to thank and express how much I enjoyed working with Prof. Emina Soljanin, Prof. Shirin Bidokhti, Dr. Yao Li, and Dr. Kaveh Mahdaviani.

I would also like to thank everyone at Rakuten USA in Boston for making my internship a great experience, namely, Pradipto Das, Ankur Datta, Giuseppe Di Fabbrizio, Aaron Levine, Allen Schmaltz, Arthur Toth, Natalija Ulemek, and Yandi Xia.

Lastly, I would like to thank my family and friends for their endless love and support. Most importantly, I would like to thank my father for a lifetime of love and sacrifice that he has provided. It is in his memory that I dedicate this thesis.
# Table of Contents

3.5 Individual Decoding Delays ............................................. 23
  3.5.1 Parity Segments Sent in Decreasing Order .................... 25
  3.5.2 Parity Segments Sent in Increasing Order ..................... 26
  3.5.3 A Numerical Comparison of Orderings .......................... 26
3.6 Conclusions ............................................................... 27

4 Converse via Multiple Descriptions .................................... 29
  4.1 Prior Work ............................................................... 29
  4.2 Main Idea ............................................................... 30
    4.2.1 A Toy Example .................................................. 30
    4.2.2 Outline of Proof .............................................. 31
  4.3 Virtual Channel Outputs ............................................. 32
    4.3.1 Construction of Virtual Channel Outputs .................... 32
    4.3.2 Properties of the Virtual Channel Outputs .................. 32
  4.4 Virtual Source Reconstructions .................................... 38
  4.5 Main Result ............................................................ 40
  4.6 Numerical Plot ......................................................... 42

5 Two-terminal Erasure Source-Broadcast with Feedback ............... 43
  5.1 Introduction ........................................................... 43
  5.2 System Model .......................................................... 44
  5.3 Coding for Two Users ................................................ 45
  5.4 Source-broadcast with One-sided Feedback ......................... 46
    5.4.1 Problem Formulation .......................................... 46
    5.4.2 Coding Scheme ................................................ 48
    5.4.3 Minmax Latency Optimality .................................. 51

6 Three-terminal Erasure Source-Broadcast with Feedback ............ 58
  6.1 Introduction ........................................................... 58
  6.2 System Model .......................................................... 59
  6.3 Coding for Three Users .............................................. 59
    6.3.1 Instantly Decodable, Distortion-Innovative Transmissions 59
    6.3.2 Non-Instantly-Decodable, Distortion-Innovative Coding .... 60
  6.4 Analysis ................................................................. 68
    6.4.1 Analysis of Coding Scheme of Section 6.3.1 ................ 68
    6.4.2 Analysis of Chaining Algorithm of Section 6.3.2 ........... 71
    6.4.3 Operational Significance of Theorem 6 ....................... 73
  6.5 Simulations ............................................................. 74
  6.6 Markov Rewards Processes with Impulse Rewards and Absorbing States ............................................. 75
    6.6.1 Related Work .................................................... 75
    6.6.2 Problem Formulation ............................................ 76
    6.6.3 Background for Absorbing Markov Chains ..................... 78
    6.6.4 Expected Rewards ............................................... 80
List of Tables

5.1 We justify the ordering of the region boundaries for $d_1$. In the left column, we have the ordering between two boundary points, and in the right column, we show the necessary and sufficient condition that justifies the ordering. ........................................... 54

5.2 We justify the ordering of the region boundaries of Figure 5.3 when $0 \leq d_1 \leq \epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12}$. In the left column, we have the ordering between two boundary points, and in the right column, we show the necessary and sufficient condition that justifies the ordering. . . . 55

5.3 We justify the ordering of the region boundaries of Figure 5.4 when $\epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12} < d_1 < \epsilon_1\epsilon_{12}$. In the left column, we have the ordering between two boundary points, and in the right column, we show the necessary and sufficient condition that justifies the ordering. 57

5.4 We justify the ordering of the region boundaries of Figure 5.5 when $\epsilon_1\epsilon_{12} \leq d_1 < \epsilon_1$. In the left column, we have the ordering between two boundary points, and in the right column, we show the necessary and sufficient condition that justifies the ordering. . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

6.1 A hypothetical sequence of transmissions and channel noises when targeting point-to-point optimal performance for user $j$ and user $k$. ................................................................. 62

6.2 A description of states in the Markov rewards process. Assume that $q_{i,j} \oplus q_{i,k}$ was sent at time $t - 1$. If the state indicated in the first column is entered, then the transmission indicated by the second column is sent at time $t$. We also show the precondition necessary to enter the state in the third column. .................................................. 63

6.3 A legend for the reward variables. ............................................................. 64

6.4 A detailed table of all outgoing transitions from State 1 based on the channel noise realization. The table includes the probability of each transition and the reward accumulated for each transition. ................................................................. 65

6.5 A detailed table of all outgoing transitions from State 2 based on the channel noise realization. The table includes the probability of each transition and the reward accumulated for each transition. ................................................................. 66

6.6 A detailed table of all outgoing transitions from State 4 based on the channel noise realization. The table includes the probability of each transition and the reward accumulated for each transition. ................................................................. 67
List of Figures

1.1 The general source-broadcast system model .................................................. 2

3.1 Broadcasting an equiprobable binary source over an erasure broadcast channel. .... 11

3.2 Distortion ratios plotted by user for \( n = 6 \) users, where user indices increase with user erasure rates. A user \( j \) is active if \( d_j / \epsilon_j < d_i / \epsilon_i \) for all \( i > j \). Active users are shown in bold. .................................................. 17

3.3 For \( n = 2 \) users, we show the demarcation of regions requiring distinct coding in the \((d_1, d_2)\)-plane. A region is shaded if its corresponding code is optimal. .................. 19

3.4 The latency plotted as more users are added to the system. The users are added in order of decreasing erasure rates for two different distortion constraints. We set \( \epsilon_i = (0.1, 0.2, 0.3, 0.4, 0.5) \) and take \( \mathbf{d}_1 = (0.01, 0.04, 0.09, 0.16, 0.25) \) in (a), while in (b) we have \( \mathbf{d}_2 = (0.01, 0.04, 0.13, 0.16, 0.25) \). .................................................. 20

3.5 The latency plotted as more users are added to the system. The users are added in order of decreasing erasure rates. We take \( \epsilon_i = c(i+1) \) and \( d_i = ci \) for \( c = 0.01 \) and \( i = 1, 2, \ldots, 80 \). 21

3.6 Latency versus \( d_2 \): \( d_1 = 1/10 \), \( \epsilon_1 = 0.3 \), \( \epsilon_2 = 0.4 \). ................................. 23

3.7 The systematic (cross-hatched) and parity components of the length-\( \mathbf{W}_i \) sequence \( \mathbf{X}_i \), which is the channel encoder output when given the length-\( \mathbf{N}_i \) source segment \( \mathbf{S}_i \). .............. 24

3.8 The ordering of \( \mathbf{P}_i \) for (a) decreasing and (b) increasing \( i \). In both cases, user \( k \) decodes the parities in Region I and ignores those in Region II. The latency of user \( k \) is \( w_k \). .............. 25

3.9 The individual latency of each user for the segment orderings of Sections 3.5.1 and 3.5.2. In (a), we set \( \epsilon = (0.1, 0.2, 0.3, 0.4, 0.5) \) and \( \mathbf{d} = \mathbf{d}_1 = (0.01, 0.04, 0.09, 0.16, 0.25) \). In (b), we take \( \epsilon = (0.31, 0.32, 0.33, 0.34, 0.35) \) and \( \mathbf{d}_3 = (0.155, 0.192, 0.231, 0.272, 0.315) \). The point-to-point optimal latency for each user, as given in (3.14), is also shown. .............. 27

4.1 A toy example involving two source symbols and three channel uses. Decoder 1 should recover the source losslessly while we wish to simultaneously minimize the distortion produced by Decoder 2. .................................................. 30

4.2 A numerical plot to illustrate the outer bound of Theorem 2. The stronger user achieves a point-to-point optimal distortion, and we plot outer and inner bounds for the weaker user’s distortion as a function of \( \epsilon_2 \). .................................................. 42

5.1 Broadcasting an equiprobable binary source over an erasure broadcast channel with a feedback channel asymmetrically available to only the stronger user. .............. 46
5.2 A tree diagram illustrating the relationship between the sets of source symbols. Each node represents a set of source symbols and a directed edge \((X, Y)\) indicates that set \(Y\) is a subset of \(X\). If edge \((X, Y)\) is also \textit{weighted}, then the weight represents the expected cardinality of set \(Y\). Only sets involved in the coding scheme have incoming \textit{weighted} edges. The direct successors of a node form a partition for the set representing the parent node. The root of the tree is the entire source sequence, which is subsequently partitioned at each depth of the tree. We also show set \(F\), which is the union of \(C\) and \(B_\theta\). 

5.3 The different regions requiring separate coding schemes when \(0 \leq d_1 \leq \epsilon_1(2\epsilon_1 - 1)/\epsilon_2\). 

5.4 The different regions requiring separate coding schemes when \(\epsilon_1(2\epsilon_1 - 1)/\epsilon_2 < d_1 < \epsilon_1\epsilon_2\). 

5.5 The different regions requiring separate coding schemes when \(\epsilon_1\epsilon_2 \leq d_1 < \epsilon_1\). 

6.1 The lower bound from Theorem 6 that delineates the boundary of distortion values for which the minmax optimal latency can be achieved. 

6.2 The total normalized number of instantly decodable, distortion-innovative packets that can be sent. 

6.3 The latency required to meet the distortion constraints.
List of Symbols

Greek

\( \gamma \)    fraction of source symbols taken from \( A \) to construct \( C \)
\( \delta \)    excess distortion of stronger user
\( \delta_R \) number of symbols taken from queue \( Q_R \) and placed in queue \( \tilde{Q}_T \)
\( \epsilon_i \) channel erasure rate for user \( i \)
\( \epsilon_{12} \) probability that an erasure event simultaneously occurs on both user 1 and user 2’s channel
\( \Theta \) impulse reward matrix
\( \theta \) fraction of source symbols taken from \( B \) to construct \( B_\theta \)
\( \Theta_1 \) submatrix of impulse rewards for transitioning from one transient state to another
\( \Theta_2 \) submatrix of impulse rewards for transitioning from a transient state to an absorbing state
\( \rho(A) \) spectral radius of matrix \( A \)
\( \rho_E \) reward matrix for number of equations received by user \( i \)
\( \rho_{ij} \) impulse reward for transitioning from state \( i \) to \( j \)
\( \rho_Q \) reward matrix for number of symbols placed in queue \( Q \)
\( \rho_u \) reward matrix for number of symbols decoded by user \( u \in \{j,k\} \)
\( \Omega \) state space for markov rewards process
\( \Omega_A \) set of absorbing states in markov rewards process
\( \Omega_T \) set of transient states in markov rewards process

Latin

\( a^1 \) critical boundary for \( d_2/\epsilon_2 \) that delineates separate coding regions
\( A \) set of source symbols received by stronger user after Phase I
\( a_i \) fraction of source symbols in source segment \( i \)
\( A_n \) upper-left submatrix of \( \hat{R}^n \)
\( b^1 \) critical boundary for \( d_2/\epsilon_2 \) that delineates separate coding regions
\( b \) latency, i.e., bandwidth expansion factor (channel uses per source symbol)
\( B \) set of source symbols containing \( \epsilon_1 - d_1 \) fraction of symbols from \( A^C \)
\( B_n \) upper-right submatrix of \( \hat{R}^n \)
\( B_\delta \) set of source symbols to be repeated containing \( 1 - \theta \) fraction of symbols from \( B \)
\( B_\theta \) set of source symbols containing \( \theta \) fraction of symbols from \( B \)
\( c^1 \) critical boundary for \( d_2/\epsilon_2 \) that delineates separate coding regions
\( C \) set of source symbols containing \( \gamma \) fraction of symbols from \( A \)
\( C(t) \) set of symbols appearing in linear combinations received by user \( i \) up to time \( t \)

\( C_{2,3} \) expected fraction of channel symbols received by weaker user in Phases II and III

\( d^1 \) critical boundary for \( d_2/\epsilon_2 \) that delineates separate coding regions

\( D_i \) distortion of user \( i \)

\( d_i \) distortion of user \( i \)

\( D_i^{*} \) point-to-point optimal distortion for user \( i \)

\( F \) set of source symbols constructed as the union \( C \cup B_\theta \)

\( H \) Hadamard product of impulse reward matrix and transition matrix

\( H_1 \) upper-left submatrix of \( H \)

\( H_2 \) upper-right submatrix of \( H \)

\( \mathcal{I}(\eta^n) \) set of indices in \( \eta^n \) that are nonzero

\( J \) set of active users

\( \mathcal{L}(\cdot) \) expected fraction of innovative source symbols received by weaker user from Phase II onwards

\( M \) number of virtual channels for weaker user

\( M^* \) number of chains built by user \( i \) before user \( j \) or \( k \) satisfies distortion constraint

\( M_i \) number of retransmissions needed for source symbol \( i \) to be received by stronger user

\( N \) fundamental matrix of absorbing Markov chain

\( N^n_i \) channel noise state sequence for user \( i \)

\( \tilde{N}^n_i \) the \( i \)th virtual channel noise state sequence

\( P \) transition matrix

\( P_i \) parity symbols of \( X_i \)

\( Q \) matrix of probabilities of transitioning from one transient state to another

\( Q^+ \) expected maximum normalized cardinality of queue \( Q \)

\( Q^* \) queue containing prioritized symbols for user \( i \)

\( q_U \) symbol taken from queue \( Q_U \)

\( \tilde{Q}_U \) preprocessed queue of symbols destined to all users in set \( U \)

\( q_U \) symbol taken from queue \( Q_U \)

\( Q_U \) queue containing symbols still required by all users in set \( U \)

\( R \) matrix of probabilities of transitioning from a transient state to an absorbing state

\( \hat{R}_\infty \) matrix of long term conditional expected accumulated reward

\( \tilde{R}_\infty \) matrix of long term scaled conditional expected accumulated reward

\( R(\tilde{n}_{00}^n) \) indices of source symbols reconstructed given channel noise state sequence \( \tilde{n}_{00}^n \)

\( \hat{R}_{E,l} \) expected number of symbols decoded by user \( i \) in \( l \)th iteration of Markov rewards process

\( \tilde{R}_n \) matrix of conditional expected accumulated reward at time \( n \)

\( \bar{R}_n \) matrix of scaled conditional expected accumulated reward at time \( n \)

\( \hat{R}_{a,l} \) expected number of symbols received by user \( u \) in \( l \)th iteration of Markov rewards process

\( S \) alphabet of source sequence

\( \hat{S} \) alphabet of source reconstruction

\( S_i \) state at time \( i \) in Markov rewards process

\( S_i \) source segment \( i \)

\( S_i \) set of random-variable source symbols not erased in \( \tilde{S}_i^{m} \)

\( \tilde{S}_i^{m} \) source reconstruction sequence corresponding to \( i \)th virtual channel output

\( \hat{S}_i^{N} \) source reconstruction sequence for user \( i \)
$S^N$  source sequence
$\mathcal{T}(t)$  set of symbols appearing in linear combination sent at time $t$
$\bar{T}_0$  expected latency to retransmit every source symbol until received by at least one user
$\bar{T}_i$  expected latency before no other transmissions of the form $q_i \oplus q_{j,k}$ can be sent
$\mathcal{U}$  set of all users
$v(t)$  random linear combination of source symbols sent at time $t$
$w$  latency (channel uses per source symbol)
$w^+$  minmax optimal latency
$w^*_i$  point-to-point optimal latency for user $i$
$\mathcal{X}$  alphabet of channel input 
$X_i$  channel input corresponding to source segment $i$
$X^W$  channel input sequence
$\mathcal{Y}$  alphabet of channel output 
$\hat{Y}_i^n$  the $i$th virtual channel output
$Y^W_i$  channel output sequence for user $i$
$\mathcal{Z}$  alphabet of channel noise state sequence
$Z^W_i$  channel noise state sequence for user $i$
Chapter 1

Introduction

The general source-broadcast problem is shown in Figure 1.1. In this problem, there is a block of $N$ source symbols we wish to communicate over $W \triangleq \lceil bN \rceil$ channel uses to $n$ users. The variable $b$ is called the latency, or alternatively, the bandwidth expansion factor. If $b > 1$, we say that there is bandwidth expansion, and alternatively, if $b < 1$, we say that there is bandwidth compression. Otherwise, we say that the bandwidth is matched if $b = 1$. The receivers have individual distortion constraints they must satisfy, and the problem is to find the achievable distortion region given $b$. Alternatively, given the distortion constraints, we may also ask what the minimum latency is so that all receivers achieve their distortion constraints.

In this dissertation, we particularize the source, channel and distortion measure of Figure 1.1 to that of a binary source, an erasure channel and an erasure distortion. That is, we study the information-theoretical limits for the problem of transmitting a binary source over an erasure broadcast channel subject to a fractional recovery constraint, which we term the erasure source-broadcast problem. The erasure channel model has been used to model packet losses due to fading or network congestion as experienced in higher layers of the protocol stack. As we discuss in Section 1.1, the fractional recovery requirement has applications in areas such as video coding. We consider the case when a feedback channel is available and also unavailable.

1.1 Motivation

The motivation for studying the source-broadcast problem is twofold. The first motivation is that of robust communications [1]. In digital communications systems, there is the inherent problem of both estimating the channel noise and the threshold effect [2]. In short, the threshold effect is the inherent problem that if a digital system is designed to perform at a certain estimated noise level, performance does not improve with more favourable channel conditions, e.g., improved signal-to-noise ratio. On the other hand, the system performance sharply degrades should the actual realized noise value be less favourable than the estimated noise value that the system targeted.

The source-broadcast problem formulation addresses this by considering multiple hypothetical receivers in a broadcast channel who have differing channel qualities and distortion requirements. We wish to find an encoding that can simultaneously satisfy all distortion requirements given the diverse channel conditions. By doing so, we can guarantee that if a receiver’s noise level were to fall off or
improve to a noise level equaling that of one of the hypothetical receivers, then he will at least achieve
the guaranteed distortion of that receiver. Thus, there will be graceful improvements, and graceful
degradation as the channel quality changes. The problem was first proposed in [1] for a Gaussian source
sent over a Gaussian broadcast channel to receivers with a mean-squared error distortion constraint.
Since then, there has been much research on this problem [3–5], however the optimal distortion region
remains unknown. Variations of this problem, including broadcasting a bivariate source [6], or having
side information available at the receivers, have also been studied [7–9] among many other variations.

The second motivation is that of broadcasting video content to diverse users. Consumers of video and
other content in today’s networks use very diverse video and computing equipment ranging from mobile
phones and handheld devices to desktops and HDTVs. When serving multiple diverse users, the most
straightforward approach is to establish independent unicast sessions. However, when a large number
of users require the same small content, (e.g., video clips at stadiums), or when a small number of users
require the same large content, (e.g., a large movie), the multiple-unicast approach clearly results in
highly inefficient use of overall network resources. In such applications, broadcast techniques can lead
to significant gains.

One important difference between point-to-point and broadcast/multicast applications lies in the
way packet losses are handled. In packet-based data networks, large files are usually segmented into
smaller blocks that are put into transport packets. Packet losses occur because of the physical channel
and other limitations such as processing power and buffer space. In point-to-point scenarios, the sender
can adjust its transmission/coding rate to avoid packet losses and retransmit lost packets according
to the feedback from the receiver through very efficient physical-layer schemes such as HARQ. In contrast,
in broadcast/multicast applications, it is costly for the sender to collect and respond to individual
receiver feedbacks, and thus HARQ schemes are disabled, and packet losses are inevitable. Forward
error correction coding provides a natural solution in such applications. A number of these schemes
have already been standardized and are being implemented. Rateless codes are a popular class of codes
that enable efficient communications over multiple unknown erasure channels at the packet level by
simultaneously approaching the channel capacity at all erasure rates. Raptor codes, a special class of
rateless codes, also have very low encoding and decoding complexity [10]. Because of these properties, Raptor codes have been standardized for Multimedia Broadcast/Multicast Service (MBMS) and are being deployed in applications such as LTE eMBMS. Raptor codes are essentially optimal for multicast over erasure channels where all receivers require identical content.

In certain applications however, the receivers may not require all the source packets and may not have identical demands. For example, in emerging eMBMS systems, there are two distinct phases of transmission. The first phase is a fixed-rate broadcast transmission, after which, each user is left with only a subset of source packets. Each user then recovers the remaining source packets through individual unicast from a dedicated repair server. Thus, during the broadcast phase, the server is required to only deliver a fraction of source packets to each user. As another example, consider a system that applies a multiple description code (MDC) [11–13] to an analog source sequence to generate a large number of MDC-coded packets. Here, the reconstruction quality depends on the number of MDC packets available for the destination. Thus, each user can have a different demand based on its screen resolution, channel condition, etc. In such scenarios where the user demands are not identical, finding both fundamental limits and practical coding schemes remains a fertile area of research to the best of our knowledge.

1.2 Related Work

The problem we study of minimizing the latency was also treated in [14] where a set of predetermined messages were required by each user such that the stronger users had to decode all the messages intended for the weaker users. Such a formulation is essentially a degraded message sets problem for which superposition coding is optimal for degraded broadcast channels. For the special case of packet erasure broadcast channels, the capacity can be achieved using optimal erasure codes. In contrast, we allow for a distortion between the source and the reconstruction. Since our distortion measure is a fractional requirement, we also allow for flexibility in which symbols are recovered so long as this number exceeds a certain threshold.

Our formulation can be viewed as a joint source-channel coding problem involving an equiprobable binary source and the erasure distortion measure. For \( s \in \mathcal{S} \), and \( \hat{s} \in \hat{\mathcal{S}} \), this distortion measure is given by

\[
d_E(s, \hat{s}) = \begin{cases} 
0 & \text{if } \hat{s} = s, \\
1 & \text{if } \hat{s} = \star \\
\infty & \text{otherwise.}
\end{cases}
\]  

The erasure distortion measure captures the fractional recovery requirement of our problem. It has found application in video coding. For example, the source symbols can be frames in a video file [15]. With fewer frames received, we may interpolate between them to view a low quality version of the video, and alternatively with many frames, we can expect better quality and less buffering time. Alternatively, multiple description video coding [11–13] has also been proposed in which the video quality depends on the number of coded packets received.

Although the separation principle of designing source codes independently of channel codes is often used in practice, joint source-channel coding has been used in many practical situations. For example, for fixed delay and complexity, designing a channel code with knowledge of the source’s properties can be used to gain significant improvements in speech coding [16]. It was also shown that joint source-channel
coding could provide an error exponent that is double that of the error exponent obtained from tandem coding in which source coding and channel coding are performed separately [17].

The source-broadcast problem we study has also been studied from a rateless coding perspective [18, 19]. Some related works that apply rateless codes to channels without state information and fading channels under delay constraints appear in [20, 21]. Based on the capacity region found in [14], the authors of [22] proposed and optimized a layered joint-source-channel-coding scheme over the binary erasure broadcast channel. While similar in spirit to this work, they do not consider partial recovery as is the focus of the present work.

As another alternative, multiple description coding has also been proposed within the literature as a method of addressing the problem we consider [23–25]. In this setup, n encoders map a source sequence into n descriptions that are to be sent over n bandwidth-constrained, errorless, parallel subchannels, each of which is equally likely to fail. In the event of a subchannel’s failure, the entire description sent over that channel is erased, whereas in the absence of a failure, the entire description is transmitted errorlessly. Given the rate of each encoder, the goal is then to find the set of \((2^n - 1)\) achievable distortions corresponding to the \((2^n - 1)\) possible subsets of descriptions that could be received errorlessly.

In [23], a symmetric version of this problem was studied, which considered a common encoder rate and distortions that depended only on the number of descriptions received. Hence, a reconstruction of the source at distortion level \(d_i\) would be expected with the reception of any \(i \in \{1, 2, \ldots, n\}\) descriptions. A coding scheme was proposed under the assumption that \(k\) out of \(n\) subchannels would not fail. The work in [24] then removed this assumption by building upon the coding scheme of [23] and successively concatenating different codes that used different values of \(k\).

Finally, a related problem to the source-broadcast problem is that of multi-level diversity coding [26], specifically that of symmetrical multi-level diversity coding [27, 28]. In this problem formulation, there are \(L\) parallel source sequences \(S^N_1, S^N_2, \ldots, S^N_L\) that are encoded by \(L\) encoders. Given a subset of \(U\) encoder outputs, the decoder is then expected to losslessly recover the sources \(S^N_1, S^N_2, \ldots, S^N_{|U|}\), where \(|U| \leq L\). The requirement that the decoder should reconstruct more data as he acquires more encoder outputs does allow for more robust communications as was our motivation in Section 1.1. However, we see that in this problem formulation, there is priority in which source sequences are recovered, e.g., \(S^N_1\) is always reconstructed if \(|U| \geq 1\). The relaxation of this requirement in our problem formulation could lead to more efficient use of channel resources based on opportunistically exploiting the channel noise. That is, by not forcing any receiver to decode specific packets and discard packets that are not useful, we can use the channel more efficiently by allowing flexibility in which packets are decoded.

### 1.3 Multiple Description Video Coding

In this section, we further elaborate on the use of multiple description coding in video applications. Such an approach has been proposed as an alternative to layered or progressive coding approaches in which a base layer, e.g., an I-frame, is encoded and subsequent refinement layers, e.g., P-frames, are conditionally encoded based on the base layer and other refinement layers. These types of encoding typically have greater compression efficiency at the cost of a more complex decoder that must receive the base and refinement layers in a fixed order. Consequently, the transmission of a progressively-coded video over a channel is typically done in a way such that the layers are retransmitted in order until an acknowledgement from the receiver is received at the transmitter through a feedback channel (see
Section 2.3 on the ARQ coding scheme). However, this may be prohibitive in live video applications such as video conferencing if the round trip time for a transmission and acknowledgment through the feedback channel is large [29]. Furthermore, when this approach is extended to a broadcast setting, forcing a receiver to decode specific packets and discard other packets could be inefficient. By avoiding this, we can use the channel more efficiently by allowing flexibility in which packets are decoded.

Multiple description coding has been proposed to circumvent some of the issues with the architecture of using progressive video codes with ARQ. The authors of [29] categorize the types of multiple description video codecs into one of five categories which include, among others, the categories of multiple description coding in the spatial domain, time domain, and frequency domain.

In multiple description coding in the spatial domain, each frame is typically subsampled at the pixel level to create descriptions. For example, every other pixel can be placed in one description while the remaining pixels are placed in another description. Variations of this technique have also involved subsampling the discrete Fourier transform (DCT) of the frame instead of the frame itself and multiple description codes operating in the spatial domain that are compliant with the H.264/AVC standards are reported in [30].

Finally, we mention that multiple description coding can also be used in conjunction with existing video coding standards. For example, in [31] the authors modify the syntax of the H.263 standard to protect more significant DCT coefficients with multiple description coding so that the coefficients can be reconstructed with better quality as the number of received descriptions increases. Alternatively, the authors of [32] have also modified the syntax of H.263 to protect motion vectors with multiple description coding.

### 1.4 Contributions

We go over the main contributions of this dissertation.

- For the erasure source-broadcast problem with \( n \) users and no feedback channel available, in Chapter 3, we propose an achievable coding scheme, i.e., an inner bound. We call the coding scheme a successive segmentation-based coding scheme and it is derived by splitting the source sequence into multiple segments and applying a systematic erasure code to each such segment. We formulate the problem of minimizing the transmission latency at the server as a linear programming problem and explicitly characterize an optimal choice for the code-rates and segment sizes. Furthermore, we give sufficient conditions for which the successive segmentation-based coding scheme is optimal. The results of this chapter have been partially presented at the 2013 IEEE Information Theory Workshop [33], the 2014 IEEE International Symposium on Information Theory [34], and published in the *IEEE Transactions on Information Theory* [35].

- For the erasure source-broadcast problem with \( n = 2 \) users where each user has a feedback channel available, we propose an optimal coding scheme in Section 5.3. We use a queue-based opportunistic network coding scheme, such that both users are sent only instantly-decodable [36], distortion-innovative packets. This result has been published in the 2015 IEEE International Symposium on Information Theory [37].

- For the erasure source-broadcast problem with \( n = 2 \) users where only the stronger user has a feedback channel available, we similarly propose an optimal coding scheme in Section 5.4. The
coding scheme makes use of uncoded transmissions, in particular, repetition coding, to avoid the threshold effect of coded transmissions. Repetition coding is, in general, inefficient in using channel resources when there is no feedback channel available to all users. However, we carefully construct our coding scheme so that we can still achieve optimality.

- For the erasure source-broadcast problem with $n = 3$ users where each user has a feedback channel available, we propose an achievable coding scheme in Section 6.3. We again use a queue-based opportunistic network coding scheme to send instantly-decodable, distortion-innovative packets. However, when sending such packets are no longer possible, we design a novel chaining algorithm that performs optimally for a wide range of channel conditions we have simulated. Furthermore, we give sufficient conditions for which the coding scheme is optimal. The results in this section have been partially published in the 2015 IEEE International Symposium on Information Theory [37].

- In analyzing the coding scheme of Section 6.3, we propose two new techniques for analyzing queue-based opportunistic network coding algorithms. The first is in deriving a linear program in Section 6.3.1 to solve for the number of instantly-decodable, distortion-innovative symbols that can be sent. The second technique is in using a Markov rewards process with impulse rewards and absorbing states to analyze queue-based algorithms in Section 6.4.2.

- In our analysis of the chaining algorithm of Section 6.4.2, we derive a more general result for explicitly finding the expected accumulated reward for a discrete-time Markov rewards process with impulse rewards and absorbing states in Section 6.6. An impulse reward is a reward $\rho_{i,j}$ that is accumulated when transitioning from state $i$ to $j$. In contrast, a rate-based reward $r_i$ is the reward for simply occupying state $i$. To the author’s best knowledge, there has yet to be an explicit equation derived for the expected reward until absorption for a discrete-time Markov rewards process with impulse rewards. Rather, the only known expression is for a discrete-time Markov rewards process with rate-based rewards and absorbing states.

- For the erasure source-broadcast problem with $n = 2$ users and no feedback channel available, we propose an outer bound (converse) for a class of non-erasure randomized codes. Our outer bound improves upon the cutset bound and is shown to coincide with the segmentation-based coding scheme of Chapter 3 when the stronger user achieves no excess distortion.
Chapter 2

Background

In this chapter, we review known results in the topics of source coding, channel coding, and joint source-channel coding. The results we review are in the context of the erasure distortion and the erasure channel. For both single and multiterminal settings, we review achievability and converse results and the effects of introducing a feedback channel.

2.1 Single-terminal Source Coding

Consider a binary, memoryless source whose symbols are independent and identically distributed (i.i.d.) according to a Bern(1/2) distribution. If the source is reconstructed subject to an erasure distortion of $D \in [0, 1]$, the rate-distortion function, $R(D)$, is given by

$$R(D) = 1 - D.$$  \hspace{1cm} (2.1)

The rate-distortion function represents the minimum bits per source symbol needed to reconstruct the source to an average distortion of $D$. This lower bound is easily achievable for a Bern(1/2) source. We simply retain the first $(1 - D)$ fraction of symbols in the source sequence, and discard the remaining $D$ fraction.

2.2 Point-to-point Channel Coding

The channel capacity for a binary erasure channel with erasure rate $\epsilon$ is given by $C$ where

$$C = 1 - \epsilon.$$  \hspace{1cm} (2.2)

The channel capacity represents the maximum rate, in bits per channel use, for which information can be reliably sent over the erasure channel. That is, we do not allow for any distortion in the information sent over the channel.

The capacity can be achieved in a variety of manners including random codes, Reed-Solomon codes, LT or rateless codes etc. Rateless codes enable efficient communications over multiple unknown erasure channels by simultaneously approaching the channel capacity at all erasure rates. We review their properties in Section 3.4. In that section, we also go over modifications to the design of rateless codes.
in order to relax the condition of reliable communications and allow for distortion in the information transmitted over the channel.

2.3 Point-to-point Channel Coding with Feedback

It is a classical result that feedback does not increase the capacity of a memoryless channel for point-to-point communications [38]. However, feedback can reduce the complexity of the achievability scheme. In particular, an automatic repeat request (ARQ) scheme can be used to achieve the capacity of an erasure channel. In such a scheme, we simply retransmit each information symbol until we receive an acknowledgement over the feedback channel that the symbol has been received.

2.4 Channel Coding for the Erasure Broadcast Channel

The erasure broadcast channel is a stochastically degraded broadcast channel and its capacity is achieved with a timesharing scheme. That is, the capacity of the erasure broadcast channel can be achieved by using a capacity-achieving point-to-point channel code to send each private message to each receiver.

2.5 Channel Coding for the Erasure Broadcast Channel with Feedback

It is a classical result that feedback does not increase the capacity of a memoryless physically degraded broadcast channel [39]. However, in general, feedback does increase the capacity of a non-physically-degraded erasure broadcast channel [40,41]. In [40], the authors use a queue-based coding scheme to track which packets arrive at an unintended destination. In doing so, network coding opportunities are created, which increases the capacity of the broadcast channel. We explain and generalize this approach for joint source-channel coding in Section 5.3.

2.6 Single-terminal Joint Source-channel Coding

The source-channel separation theorem provides a necessary condition for a source is to be communicated over a channel with capacity $C$ subject to a distortion constraint of $D$. The theorem states that

$$R(D) \leq bC,$$

(2.3)

where $b$ is the latency, i.e., the number of channel symbols per source symbol. If we specialize for the case of a Bern(1/2) source, the erasure distortion, and an erasure channel, we get that

$$(1 - D) \leq b(1 - \epsilon).$$

(2.4)

We may use (2.4) to give a lower bound on the latency or the distortion. When doing so, we will occasionally refer to this lower bound as the point-to-point bound, the Shannon bound, or the cutset bound.
2.7 Single-terminal Joint Source-channel Coding with Feedback

It is a known result that the source-channel separation theorem still holds if a feedback channel is available [42, Remark 3.12], [43]. However, as discussed in Section 5.3, we may again reduce complexity by employing an ARQ scheme.

2.8 Multi-terminal Joint Source-channel Coding

In this thesis, we study the erasure source-broadcast problem and give inner and outer bounds when there is no feedback channel available. When a feedback channel is available, we given inner bounds for the case of two and three users.
Chapter 3

Erasure Source-Broadcast without Feedback

Motivated by error correction coding in multimedia applications, we study the problem of broadcasting a single common source to multiple receivers over heterogeneous erasure channels. Each receiver is required to partially reconstruct the source sequence by decoding a certain fraction of the source symbols. We propose a coding scheme that requires only off-the-shelf erasure codes and can be easily adapted as users join and leave the network. Our scheme involves splitting the source sequence into multiple segments and applying a systematic erasure code to each such segment. We formulate the problem of minimizing the transmission latency at the server as a linear programming problem and explicitly characterize an optimal choice for the code-rates and segment sizes. Through numerical comparisons, we demonstrate that our proposed scheme outperforms both separation-based coding schemes, and degree-optimized rateless codes and performs close to a natural outer (lower) bound in certain cases. We further study individual user decoding delays for various orderings of segments in our scheme. We provide closed-form expressions for each individual user’s excess latency when parity checks are successively transmitted in both increasing and decreasing order of their segment’s coded rate and also qualitatively discuss the merits of each order.

3.1 Introduction

In this chapter, we propose a coding scheme for transmitting to multiple receivers with heterogeneous channels and demands. Our scheme relies only on off-the-shelf erasure codes. The key idea in our scheme is to partition the source sequence into multiple non-overlapping segments and to apply a systematic erasure code to each segment. We formulate the problem of selecting the segment lengths and code rates that minimize the transmission latency as a linear programming problem and characterize an explicit solution. We discuss how the solution naturally evolves as users join or leave the network. We further compare our scheme numerically with separation-based schemes, and degree-optimized rateless codes and demonstrate that significant performance gains are possible. We also discuss how a tradeoff between the latencies of individual users can be attained by selecting various transmission orders for the parity checks.

Throughout this chapter, we adhere to the notation defined herein. The sample space of a random
3.2 System Model and Prior Work

3.2.1 System Model

The problem is illustrated in Fig. 3.1. We consider a binary memoryless source \( \{S(t)\}_{t=1,2,...} \) that produces equiprobable symbols in the alphabet \( \mathcal{S} = \{0, 1\} \) and that we wish to communicate to \( n \) users over an erasure broadcast channel. The source is communicated by a block-encoding function that maps a length-\( N \) source sequence, \( S^N \), to a length-\( W \) channel input sequence, \( X^W = (X(1), X(2), \ldots, X(W)) \), where \( X(t) \) denotes the \( t \)th channel input taken from the alphabet \( \mathcal{X} = \{0, 1\} \).

Let \( Y_i(t) \) be the channel output observed by user \( i \) on the \( t \)th channel use for \( i \in [n] \) and \( t \in [W] \). Our channel model is a binary erasure broadcast channel as shown in Fig. 3.1. In particular, let \( \epsilon_i \) denote the erasure rate of the channel corresponding to user \( i \), where we assume that \( 0 < \epsilon_1 < \epsilon_2 < \ldots < \epsilon_n < 1 \). This is without loss of generality since we can address all users that experience identical erasure rates by serving the one with the most stringent distortion requirement. Our model specifies that \( Y_i(t) \) exactly reproduces the channel input \( X(t) \) with probability \( (1 - \epsilon_i) \) and otherwise indicates an erasure event, which happens with probability \( \epsilon_i \). We let \( Y_i(t) \) take on values in the alphabet \( \mathcal{Y} = \{0, 1, \star\} \) so that an erasure event is represented by ‘\( \star \)’, the erasure symbol. We assume that erasure events on each channel occur independently of each other. That is, we assume the conditional probability mass function of the channel outputs satisfies
\[
\Pr(Y_1^W = y_1^W, \ldots, Y_n^W = y_n^W | X^W = x^W) = \prod_{i=1}^{n} \Pr(Y_i^W = y_i^W | X^W = x^W) = \prod_{i=1}^{n} \prod_{t=1}^{W} \Pr(Y_i(t) = y_i(t) | X(t) = x(t)),
\]

where \( \Pr(Y_i(t) = \cdot | X(t) = \cdot) \) is the channel transition probability for the \( i \)th erasure channel. Note that in our setup, the erasure rates for each user are assumed to be known. However, our setup also models the compound channel where the erasure rate is not known. Instead, the erasure rate belongs to a collection of possible states with each state corresponding to one virtual user in our system [38, p. 562].

Having observed his channel output, user \( i \) then uses it to reconstruct the source as a length-\( N \) sequence, denoted as \( \hat{S}_i^N \). We will be interested in a fractional recovery requirement so that each symbol in \( \hat{S}_i^N \) either faithfully recovers the corresponding symbol in \( S_i^N \), or otherwise a failure is indicated with an erasure symbol, i.e., we do not allow for any bit flips.

More precisely, we choose the reconstruction alphabet \( \hat{S} \) to be an augmented version of the source alphabet so that \( \hat{S} = \{0, 1, *\} \), where the additional ‘*’ symbol indicates an erasure symbol. We then express the constraint that an achievable code ensures that each user \( i \in [n] \) achieves a fractional recovery of \( 1 - d_i \), where \( d_i \in [0, 1] \), with the following definition.

**Definition 1.** An \((N, W, d_1, d_2, \ldots, d_n)\) code for source \( S \) on the erasure broadcast channel consists of

1. an encoding function \( f_N : S^N \rightarrow X^W \) such that \( X^W = f_N(S^N) \), and
2. \( n \) decoding functions \( g_{i,N} : Y^W \rightarrow \hat{S}_i^N \) such that \( \hat{S}_i^N = g_{i,N}(Y_i^W) \) and for each \( i \in [n] \),

   (a) \( \hat{S}_i^N \) is such that for \( t \in [N] \), if \( \hat{S}_i(t) \neq S(t) \), then \( \hat{S}_i(t) = * \),

   (b) \( \mathbb{E} \left| \{t \in [N] \mid \hat{S}_i(t) = *\} \right| \leq N d_i \),

where \( \mathbb{E}(\cdot) \) is the expectation operation and \( |A| \) denotes the cardinality of set \( A \).

For a given code, we next define the **latency** that the code requires before all users can recover their desired fraction of the source. Finally, we then state our problem as characterizing the achievable latencies under a prescribed distortion vector as per the following definitions.

**Definition 2.** The latency, \( w \), of an \((N, W, d_1, d_2, \ldots, d_n)\) code is the number of channel uses per source symbol that the code requires to meet all distortion demands, i.e., \( w = W/N \).

**Definition 3.** Latency \( w \) is said to be \((d_1, d_2, \ldots, d_n)\)-achievable over the erasure broadcast channel if for every \( \delta > 0 \), there exists for sufficiently large \( N \), an \((N, wN, \hat{d}_1, \hat{d}_2, \ldots, \hat{d}_n)\) code such that for all \( i \in [n] \), \( d_i + \delta \geq \hat{d}_i \).

**Remark 1.** Throughout this chapter we will assume that for each user \( i \in [n] \), we have that \( d_i < \epsilon_i \). Any user \( j \) with \( d_j \geq \epsilon_j \) will be trivially satisfied by the systematic portion of our segmentation-based coding scheme. Furthermore, we will show in Lemma 19 that within our class of coding schemes, such a systematic portion can be transmitted without loss of optimality when at least one user satisfies \( d_i < \epsilon_i \). Finally, if every user satisfies \( d_i \geq \epsilon_i \), a simple uncoded transmission scheme is easily shown to be optimal.
Remark 2. While our system model has assumed binary alphabets for both source and channel input sequences, our results can easily be extended to larger alphabet sizes. Provided we keep source and channel input alphabets identical in size, our results could then extend to packet erasure networks.

A code that satisfies the content demands of a set of users may in fact afford different users the ability to finish receiving their content at intervals so that some users require only a short latency while others require longer ones (e.g., see [44]). In particular, we can also define what we will call a discretized code, which accounts for users’ separate decoding latencies.

Definition 4. An \((N, W_1, W_2, \ldots, W_n, d_1, d_2, \ldots, d_n)\) discretized code for source \(S\) on the erasure broadcast channel consists of

1. an encoding function \(f_N : S^N \to X^W\) such that \(X^W = f_N(S^N)\), and \(W = \max_{i \in [N]} W_i\),
2. \(n\) decoding functions \(g_{i, N} : Y^{W_i} \to \hat{S}_i^N\) such that \(\hat{S}_i^N = g_{i, N}(Y_i^{W_i})\) and for each \(i \in [n]\),
   
   \(a\) \(\hat{S}_i^N\) is such that for \(t \in [N]\), if \(\hat{S}_i(t) \neq S(t)\), then \(\hat{S}_i(t) = \star\),
   
   \(b\) \(\mathbb{E}\{t \in [N] \mid \hat{S}_i(t) = \star\} \leq N d_i\),

Using Definition 4, we can similarly define what it means when latency tuple \((w_1, w_2, \ldots, w_n)\) is \((d_1, d_2, \ldots, d_n)\)-achievable as in Definition 3.

Clearly, if we let \(W = \max_{i \in [N]} W_i\), we see that an \((N, W_1, W_2, \ldots, W_n, d_1, d_2, \ldots, d_n)\) discretized code is also an \((N, W, d_1, d_2, \ldots, d_n)\) code. Definition 4 is of interest from the perspective of content consumers as it concerns both the latencies that they will each have to endure for their content requirements and also the possible tradeoffs amongst themselves. Alternatively, Definition 3 is unconcerned with individual latencies and instead provides us with the minmax latency metric by taking the maximum over all user latencies. In this way, the minmax latency metric is of interest from a content provider’s perspective as it will allow the provider to compare codes based on which ones minimize the overall transmission time that is required from the provider.

The focus in this chapter will primarily be the minmax latency metric, and the solution that we propose is a code that is \((\text{minmax})\) latency-optimal within the class of segmentation-based codes. Our discussion of individual latencies will be limited to Section 3.5 where given a segmentation-based code, we consider different transmission orderings of the segments for individual latency considerations.

3.2.2 Prior Work

As mentioned earlier, our formulation can be viewed as a joint source-channel coding problem involving an equiprobable binary source and the erasure distortion measure. The erasure distortion measure has been studied in the related context of multiple description coding in [25]. In our intended context of joint source-channel coding, it has also been studied for the case of two users in [33, 45]. The coding schemes in these two works involved adaptations of techniques used in the Gaussian models (see e.g., [1, 3, 4, 46–48] and references therein). To the best of our knowledge, such schemes do not attain smaller latencies than the scheme proposed in the present chapter. Furthermore, such schemes involve joint source-channel code designs and do not have the practical advantages of the proposed scheme that were discussed previously.

An “erasure” version of the symmetric multiple-description problem was also studied in [25], which considered an erasure distortion as well as a no-excess rate constraint for every \(m\) out of \(n\) descriptions.
Interestingly, the coding scheme used in [25] built upon the ideas of [23,24] and resulted in a segmentation-based scheme similar to ours where the source was segmented into equal segments that were then each encoded with a systematic erasure code. In contrast, their work, however, did not involve any optimization over segment sizes. While these works do have high-level similarities and draw upon common practical motivations, there is another important distinction between our work and multiple description coding. This is that, fundamentally, the problem we consider is a joint-source channel coding problem. That is, in our formulation, the size of each channel symbol is fixed, while the number of channel uses approaches infinity. In contrast, in multiple description coding, the number of channels remains fixed, whereas the number of bits sent over each channel approaches infinity.

The segmentation-based code we present is also related to the coding scheme recently proposed in [49], which was studied independently of our work and presented alongside it at a recent conference. In this work, the authors consider combining a successive refinement code with a timesharing strategy that individually channel codes messages intended for different users listening over the broadcast channel. As we will see, the code we present is similar in its use of a successive refinement code and a timesharing strategy. However, we will also see that our particular distortion measure is matched with the erasure channel in such a way that we are also able to benefit from the use of uncoded transmissions.

Finally, it is also worth mentioning that in terms of an outer bound, techniques that involve auxiliary channels have been developed for both the Gaussian model [4] and a more general model of a discrete memoryless source sent over a discrete memoryless broadcast channel [50]. While the techniques and inequalities used in [4] can be adapted for the erasure channel [51,52], we have found that doing so does not result in an outer bound that improves upon the point-to-point outer bound in the present setup. The difficulty encountered is in defining a suitable auxiliary channel that would lead to a non-trivial bound. For the closely related problem involving the erasure broadcast channel and a Hamming distortion, an auxiliary channel can be chosen in order to derive a non-trivial outer bound. [50,53]. For our problem involving the erasure distortion, in Chapter 4 we will derive a non-trivial outer bound for a class of non-randomized-erasure codes with techniques that do not require the introduction of an auxiliary channel.

### 3.3 Segmentation-Based Coding

#### 3.3.1 The Main Idea

Let \( v \) denote the user with the highest erasure rate, and consider the case when this user is the only one in our system. It is well known [38] that the optimal latency of \( (1 - d_v)/(1 - \epsilon_v) \) can be achieved by, e.g., first compressing the source with distortion \( d_v \) and then losslessly transmitting the compressed version of the source with a channel code of rate \( (1 - \epsilon_v) \). The compression process is particularly simple in our case; we simply retain the first \( N(1 - d_v) \) source sequence symbols and discard the remaining symbols. Note that this (separation) scheme can also be decoded by any user \( s \) with erasure rate \( \epsilon_s \leq \epsilon_v \) and results in the same distortion \( d_v \). Thus, if \( d_s \geq d_v \), the introduction of user \( s \) into the system does not modify the code since user \( s \) does not require any dedicated coding. Consider, however, when \( d_s < d_v \).

We accommodate user \( s \) by incrementally modifying our coding; in addition to transmitting the first \( N(1 - d_v) \) source symbols as before, we also transmit the following \( N(d_v - d_s) \) source symbols with a channel code of rate \( (1 - \epsilon_s) \). Thus, if \( d_s < d_v \), the addition of user \( s \) does modify the code since user \( s \)
does require dedicated coding. It is not hard to generalize this type of coding for \( n \) users. We simply identify the users that require dedicated coding and code for only these users by following the procedure mentioned above. In general, we see that for \( 1 \leq i < j \leq n \), user \( i \) is able to decode whatever was channel coded for user \( j \) since we have assumed that user indices increase with erasure rates. Therefore, user \( i \) requires dedicated coding only if whatever was already sent to users with worse channel qualities is not sufficient for his own distortion requirement, i.e., if \( d_i < d_j \) for \( j \in \{i+1, i+2, \ldots, n\} \). For future reference, we will call this a layered coding scheme.

We observe that whenever a user does not require dedicated coding, he achieves the same distortion as some user \( j \) who has a worse channel quality and who does require dedicated coding. Thus, this coding does not allow for graceful improvements in distortion for increasingly favourable channel qualities. We circumvent this by modifying our coding. Consider again the case when user \( v \) is the only user in the system. Instead of the separation-based scheme, we now split the source sequence into two segments. The first segment consists of a fraction of \( a_0 \) source symbols and is transmitted uncoded, while the second segment consists of a fraction of \( a_v \) source symbols and is transmitted using a systematic channel code of rate \((1 - \epsilon_v)\). Note that the latency in this scheme is \( a_0 + a_v/(1 - \epsilon_v) \), while the fraction of symbols received is \( a_0(1 - \epsilon_v) + a_v \). By setting \( a_0 = d_v/\epsilon_v \) and \( a_v = 1 - a_0 \), we achieve the same latency as the (optimal) separation-based scheme while satisfying the distortion constraint.

Fundamentally, this approach functions by first ensuring that user \( v \) losslessly recovers all but a fraction of \( d_v/\epsilon_v \) source symbols via a channel code. By construction, the positions of the missing \( Nd_v/\epsilon_v \) symbols are known. Therefore, if they are transmitted uncoded in a second step, we expect that a reduced number of only \( N(d_v/\epsilon_v) \cdot \epsilon_v = Nd_v \) symbols will be missing afterwards.

In what follows, we will extend this approach to the case of \( n \) receivers. For \( i \in [n] \), instead of guaranteeing user \( i \)’s recovery of all but the last \( Nd_i \) source symbols as in the layered approach, we will instead guarantee his recovery of all but the last \( Nd_i/\epsilon_i \) symbols. Each user can then recover what he additionally requires by listening to uncoded transmissions or the systematic portions of the channel codes used. For the layered scheme, we saw that if user \( i \) recovered all but the last \( Nd_i \) symbols, he would require dedicated coding if \( d_i < d_j \) for all \( j > i \). Since we guarantee the recovery of all but the last \( Nd_i/\epsilon_i \) symbols in our new coding, we will analogously see in Section 3.3.2, when defining active users, that a user \( i \) requires dedicated coding in our proposed code if \( d_i/\epsilon_i < d_j/\epsilon_j \) for all \( j > i \).

### 3.3.2 Scheme Description

In this section, we formally discuss the class of segmentation-based schemes and formulate the problem of selecting optimal segment sizes and channel code rates. We then present an analytical solution and discuss connections with the scheme presented in the previous subsection.

The source sequence \( S_N \) is divided into \( K + 1 \) non-overlapping subsequences, \( S_0, S_1, \ldots, S_K \), where for \( k = 0, 1, \ldots, K \), \( S_k \) carries \( a_k \) fraction of source bits and \( \sum_{k=0}^K a_k \leq 1 \). For each \( k \), the segmentation encoder maps subsequence \( S_k \) into channel input \( X_k \) by using a rate-\( r_k \) systematic erasure code. We take \( r_0 = 1 \) so that \( X_0 = S_0 \), i.e., \( S_0 \) is sent uncoded. The broadcast channel input sequence \( X^W \) is obtained by concatenating the segments \( X_0, X_1, \ldots, X_K \).

User \( i \) observes the channel input through a channel with erasure probability \( \epsilon_i \) and can therefore completely recover all source segments that are coded at rates \( r_k \leq 1 - \epsilon_i \). He further recovers an additional \((1 - \epsilon_i)\) fraction of source segments coded at rates \( r_k > 1 - \epsilon_i \) due to the systematic (uncoded) part of the channel code used. This is formally stated in the following claim, which directly follows from...
Definition 3 and by construction of the scheme.

Claim 1. The above segmentation-based coding scheme has latency

$$a_0 + \frac{a_1}{r_1} + \cdots + \frac{a_K}{r_K}, \quad (3.3)$$

and the fraction of source symbols recovered at user $i$ is

$$\left\{ (1 - \epsilon_i) \sum_{0 \leq j \leq K} a_j + \sum_{0 \leq k \leq K} a_k \right\}. \quad (3.4)$$

Remark 3. It is interesting to note that by (3.4), the source symbols that are ultimately not recovered by a weaker user are concentrated in segments coded for stronger users. Furthermore, $S_n$ is recovered by all users. Although exploring these properties is beyond the scope of our work, we mention that it may be useful in certain applications.

Remark 4. As mentioned earlier, the segmentation-based scheme requires only off-the-shelf erasure codes to be separately applied to non-overlapping source segments. Its computational complexity is therefore no worse than that of its highest-complexity constituent erasure code.

Note that in our formulation so far, the segment sizes, $a_i$, the associated code-rates, $r_i$, as well as the number of segments, $K$, need to be specified. Our optimization problem involves selecting these parameters such that the latency in (3.3) is minimized and for each user $i$, the received fraction of symbols in (3.4) is at least equal to $1 - d_i$. We first show that the choice of optimal rates, $r_i$, admits a natural solution that significantly simplifies our optimization problem.

Claim 2. The latency of a segmentation-based scheme can be reduced with no penalty in achievable distortion by modifying its segment lengths, $a_0, a_1, \ldots, a_K$, and code rates, $r_1, \ldots, r_K$, s.t. the rates belong to the set $\mathcal{R} = \{1\} \cup \{1 - \epsilon_i, i \in [n]\}$.

Proof. The proof is given in Appendix A.1.

With Claims 1 and 2 in hand, we can formulate an optimization problem to minimize the system latency over the segment lengths $\mathbf{a} = (a_0, a_1, \ldots, a_n)$ given the distortion constraints as follows.

$$\min_{\mathbf{a}} \quad a_0 + \frac{a_1}{1 - \epsilon_1} + \cdots + \frac{a_n}{1 - \epsilon_n}$$

subject to

$$a_0 + a_1 + \cdots + a_n \leq 1,$$

$$\left(1 - \epsilon_i\right) \sum_{j=0}^{i-1} a_j + \sum_{j=i}^{n} a_j \geq 1 - d_i, \quad \text{for } i \in [n]$$

$$a_j \geq 0, \quad \text{for } j = 0, 1, \ldots, n.\quad (3.5)$$

One may wonder whether it suffices to replace the inequality constraint $a_0 + a_1 + \ldots + a_n \leq 1$ in (3.5) with a strict equality constraint. It is not obvious a priori if this can be done. Indeed, there can exist optimal solutions to (4) where the inequality is strict. However, in our proof of Theorem 1, (more specifically in Lemma 19 of Appendix A.2), we establish that there exists an optimal solution that satisfies the aforementioned constraint with equality.
Chapter 3. Erasure Source-Broadcast without Feedback

Figure 3.2: Distortion ratios plotted by user for \( n = 6 \) users, where user indices increase with user erasure rates. A user \( j \) is active if \( d_j/\epsilon_j < d_i/\epsilon_i \) for all \( i > j \). Active users are shown in bold.

We provide an explicit solution to (3.5) below. We first define the set of active users, \( J \), as the set containing users whose distortion-erasure ratio is smaller than that of every user with a higher erasure rate (see Fig. 3.2 for an illustration), i.e.,

\[
J = \{ j_1, j_2, \ldots, j_l \} = \{ j \in [n] : d_j/\epsilon_j < d_i/\epsilon_i, \forall i > j \}.
\]  

(3.6)

Note that from the above definition, it immediately follows that if \( J = \{ j_1, j_2, \ldots, j_l \} \) and \( j_1 < j_2 < \cdots < j_l \), then \( d_{j_1}/\epsilon_{j_1} < d_{j_2}/\epsilon_{j_2} < \cdots < d_{j_l}/\epsilon_{j_l} < 1 \). Moreover, we can easily see that \( J \) is non-empty since \( n \in J \) and \( j_l = n \).

**Theorem 1.** Let \( 0 < \epsilon_1 < \epsilon_2 < \cdots < \epsilon_n < 1 \), \( (d_1, d_2, \ldots, d_n) \) be a distortion vector, and \( J \) be defined as above. Then the optimal solution to (3.5) gives a latency of

\[
\frac{d_{j_1}}{\epsilon_{j_1}} + \sum_{m=1}^{l-1} \frac{1}{1 - \epsilon_{j_m}} \left( \frac{d_{j_{m+1}}}{\epsilon_{j_{m+1}}} - \frac{d_{j_m}}{\epsilon_{j_m}} \right) + \frac{1}{1 - \epsilon_{j_l}} \left( 1 - \frac{d_{j_l}}{\epsilon_{j_l}} \right),
\]

which is \( (d_1, d_2, \ldots, d_n) \)-achievable by a segmentation-based coding scheme with \( |J| + 1 = l + 1 \) segments of normalized segment lengths

\[
a_0 = \frac{d_{j_1}}{\epsilon_{j_1}}, \quad a_{j_l} = 1 - \frac{d_{j_l}}{\epsilon_{j_l}},
\]

\[
a_{j_m} = \frac{d_{j_{m+1}}}{\epsilon_{j_{m+1}}} - \frac{d_{j_m}}{\epsilon_{j_m}} \quad \text{for} \ 1 \leq m < l,
\]

(3.7)

and corresponding code rates

\[
r_0 = 1 \quad \text{and} \quad r_{j_m} = 1 - \epsilon_{j_m} \quad \text{for} \ 1 \leq m \leq l.
\]

**Proof.** The proof is given in Appendix A.2.

It is interesting to ask if and how the scheme has to be redesigned if another user, \( s \), joins the system. If the new user arrives prior to the start of the transmission block, only an incremental adjustment to the original code is needed. Clearly, the scheme will be affected only if the user’s parameters place him in \( J \). Suppose this is so. In this case, the arrival of user \( s \) will have two effects. Firstly, user \( s \) may displace stronger users from \( J \). Specifically, for each user \( i \in J \) with a better channel than user \( s \), we
must re-evaluate whether or not user \( i \) still belongs in \( J \), i.e., if \( d_i/\epsilon_i < d_s/\epsilon_s \). Secondly, for those users who no longer meet this condition, we merge the segments that were originally channel coded for each of them and subsequently split the resulting combined segment into two new segments. Suppose that after re-evaluating the set \( J \), user \( s \) is adjacent to users \( r \) and \( t \) in \( J \) where \( \epsilon_r < \epsilon_t \). Then the two new segments that replace the merged segments consist of one that is of size \( d_t/\epsilon_t - d_s/\epsilon_s \) and protected by a channel code of rate \( 1 - \epsilon_s \), and another that is of size \( d_s/\epsilon_s - d_r/\epsilon_r \) and protected by a channel code of rate \( 1 - \epsilon_r \). The departure of user \( s \) reverses this process. Note that this scheme scales easily with the number of users.

Alternatively, if the new user joins midway during the transmission block, his distortion for that particular block will depend on when he joins. The system, however, would be able to adjust at the start of the next block to accommodate the new user.

### 3.3.3 Special Cases

We now consider several interesting values for the erasure rates and distortion constraints and interpret the segmentation-based coding scheme in these special cases.

#### Uniform Channel Condition

When all users are subject to the same channel erasure rate, \( \epsilon_1 \), we effectively have \( n = 1 \). As in Section 3.3.1, we simply set \( a_0 = \frac{d_0}{\epsilon_1} \) and \( a_1 = 1 - \frac{d_0}{\epsilon_1} \), where \( d_0 \) is the minimum distortion of all users. The latency achieved equals \( \frac{1 - d_0}{1 - \epsilon_1} \), which is easily seen as optimal as it coincides with the separation-based outer (lower) bound.

#### Uniform Distortion

When all users have the same distortion constraint, \( d \) but experience different channel erasure rates, we have that \( J = \{ n \} \) so that we encode for the weakest user by setting \( a_0 = \frac{d}{\epsilon_n} \) and \( a_n = 1 - \frac{d}{\epsilon_n} \). All stronger users achieve progressively better distortions, and the latency achieved is \( \frac{1 - d}{1 - \epsilon_n} \), which is optimal.

#### Constant \( \frac{d_i}{\epsilon_i} \)

If \( \frac{d_i}{\epsilon_i} = c < 1 \) for each user \( i \in [n] \), we again have that \( J = \{ n \} \). Thus, \( a_0 = c \), \( a_n = 1 - c \), and we achieve a latency of \( w = \frac{1 - cc_n}{1 - \epsilon_n} = \frac{1 - d_0}{1 - \epsilon_n} \), which is again optimal.

\( d_i = \epsilon_i^2 \)

When user distortions are quadratic in their erasure rates, we have \( \frac{d_i}{\epsilon_i} = \epsilon_i \), and hence \( J = [n] \). Thus, \( a_0 = \epsilon_1 \), \( a_i = \epsilon_{i+1} - \epsilon_i \) for \( i \in [n-1] \), and \( a_n = 1 - \epsilon_n \). We refer to this as the “proportional allocation scheme.” The amount of bits allocated to the segment protected with an erasure code of rate \( (1 - \epsilon_i) \) is the difference in the channel capacity between user \( i \) and the next weakest user, user \( i + 1 \). The latency achieved in this case is \( w = 1 + \epsilon_1 + \sum_{i=1}^{n-1} \frac{\epsilon_{i+1} - \epsilon_i}{1 - \epsilon_i} \).
Chapter 3. Erasure Source-Broadcast without Feedback

Figure 3.3: For \( n = 2 \) users, we show the demarcation of regions requiring distinct coding in the \((d_1, d_2)\)-plane. A region is shaded if its corresponding code is optimal.

**Two Users**

When there are only two users in the system, we can partition the \((d_1, d_2)\)-plane into distinct regions that each have a separate encoding scheme (see Fig. 3.3). Region I is where \( d_i \geq \epsilon_i \) for both \( i = 1, 2 \). Clearly, an uncoded transmission strategy is optimal in this case, and so we shade this region in Fig. 3.3 to indicate that we have matching inner and outer bounds. Similarly, in Region II, where \( d_2 \geq \epsilon_2 \) but \( d_1 < \epsilon_1 \), we can also be optimal, albeit this time with a segmentation-based code. The segmentation is done as if user 1 is the only user in the network, and the systematic portion of the code is sufficient for user 2 as each source symbol is eventually sent uncoded over the channel (see Remark 1 and Lemma 19). An analogous argument can be made for Region III where we would code as if user 2 is the only user in the network. Next, Region IV(a) illustrates the final region where we obtain optimality, which happens when \( d_2/\epsilon_2 \leq d_1/\epsilon_1 \leq 1 \). In this case, only user 2 is active (see (3.6)), and the coded/uncoded transmissions for user 2 is also sufficient for user 1 (see Section 3.3.1). Region IV(b) is the final region and represents the only region in which there is a tension between user needs. In this region, both users are active, i.e., \( d_1/\epsilon_1 < d_2/\epsilon_2 \), and the coding must account for the presence of both users.

**3.3.4 Numerical Comparisons**

We compare the latency achievable by our segmentation scheme of Theorem 1 against some baseline coding schemes. The comparison is done in a way that parallels the discussion in Section 3.3.1. We first consider a single user and successively add additional users to see how the overall latency changes as a function of the number of users in the network. The users are added in decreasing order of erasure rates. The first coding scheme we compare Theorem 1 to is a separation-based approach which, for example, may be implemented with a random linear network code (RLNC) [54]. In our context, an RLNC coding scheme refers to a point-to-multipoint coding scheme that creates random linear combinations of source symbols at the transmitter. It does not incorporate any combining of channel packets at intermediate network nodes. With an RLNC scheme, we satisfy all user demands by sending a common message that is intended for everyone to decode. The common message is a compressed version of the source at a distortion equal to the minimum of all user distortion constraints. It is channel coded at a rate that the weakest user can decode. This scheme achieves an overall latency of \( w_{\text{RLNC}} = \frac{1-\min_{i \in [n]} d_i}{1-\epsilon_n} \). The reader may verify that the RLNC scheme is also optimal in cases (1) and (2) in Section 3.3.3 but will lead to...
higher latencies in the remaining cases.

In order to understand the importance of choosing segment sizes, another coding scheme we consider is a simplified version of the optimization problem in (3.5) where all the non-zero segment sizes are forced to be identical. In particular, each segment, $a_i$, for $i \in \{0, 1, \ldots, n\}$, can be either zero or take a fixed value. We note that the RLNC scheme is a special case of this scheme when only $a_n$ is non-zero. Finally, the layered coding scheme of Section 3.3.1 is the last baseline coding scheme we consider.

The numerical comparisons are shown in Fig. 3.4 where we have taken $n = 5$. Let $\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_5)$ and $d_1 = (d_1, d_2, \ldots, d_5)$. In the first plot of Fig. 3.4, we set $\epsilon$ and $d_1$ so that for $i \in \{1, 2, \ldots, 5\}$, $\epsilon_i = 0.1 \times i$ and $d_i = \epsilon_i^2$. For the sake of clarity, $\epsilon = (0.1, 0.2, 0.3, 0.4, 0.5)$ and $d_1 = (0.01, 0.04, 0.09, 0.16, 0.25)$. In this case, it can be seen that the addition of each user expands the set $J$ and thus leads to an increase in latency. In the second plot of Fig. 3.4, we have slightly modified only the third component of the distortion vector so that now the third user requires a higher distortion of 0.13 instead of 0.09. For clarity, we now have that $d_2 = (0.01, 0.04, 0.13, 0.16, 0.25)$. In this case, we see that for our proposed scheme, when the third user is added to the network, his distortion is sufficiently high so that he can simply meet his distortion constraint by virtue of his better channel quality and from what is already sent over the channel (cf. Section 3.3.1). The latency does not increase in this step. In all cases, we see that our proposed coding scheme performs much better than the other baseline schemes.

Finally, we further highlight the potential benefits of Theorem 1 by plotting a larger example with 80 users. In Fig. 3.5, we take $\epsilon_i = c(i + 1)$ and $d_i = ci$ for $c = 0.01$ and $i = 1, 2, \ldots, 80$. Note that Fig. 3.5 again adds users in order of decreasing erasure rates so that, in fact, user 80 is added first. For this example, we see that all users require dedicated coding for both Theorem 1 and the layered scheme. As described in Section 3.3.1, the layered scheme channel codes a fraction of $d_{i+1} - d_i = c$ source symbols for user $i$, which is constant among all users. In contrast, Theorem 1 channel codes a fraction $d_{i+1}/\epsilon_{i+1} - d_i/\epsilon_i = 1/(i+1)(i+2)$ for user $i$. Thus, we see that the longest segments are sent to better users for Theorem 1. In addition, for Theorem 1, the size of the segment coded for user $i$ decreases
Figure 3.5: The latency plotted as more users are added to the system. The users are added in order of
decreasing erasure rates. We take $\epsilon_i = c(i + 1)$ and $d_i = ci$ for $c = 0.01$ and $i = 1, 2, \ldots, 80$.
quickly as $i$ increases. On the other hand, it stays constant for the layered scheme and Fig. 3.5 reflects
this advantage.

3.4 A Comparison to Rateless Codes

In this section, we compare our segmentation-based scheme with rateless codes optimized for unequal
user demands. For simplicity, we make our comparison for the case of $n = 2$ users (see Section 3.3.3 for a
discussion of this special case). As discussed earlier, rateless codes provide near-optimal, low-complexity
performance when the users are interested in identical content.

A rateless code maps $N$ binary source symbols, $\{u_1, \ldots, u_N\}$, into a potentially infinite sequence of
binary code symbols $\{v_l\}_{l=1}^{\infty}$, where $v_l$ are linear combinations of $\{u_1, \ldots, u_N\}$, i.e.,
$v_l = \theta_1^l u_1 + \cdots + \theta_N^l u_N$, $\theta_j^l \in \{0, 1\}$. The coefficients $\theta_j^l$ are generated in the following way: (1) we select a degree
distribution $\{p_1, \ldots, p_N\}$ for the code, and for each $v_l$, we sample the associated degree $M$ from this
distribution; (2) we randomly and uniformly select $M$ elements from the set $\{\theta_1^l, \ldots, \theta_N^l\}$ to be non-
zero and let the remaining entries be zero. In the classical rateless code design [55, 56], the degree
distribution is selected such that the overhead when recovering all $N$ source symbols is kept as small as
possible. However, in our present setup, where each receiver requires different demands, such a degree
distribution will not be suitable. Building upon the approach taken in [18], [19], we briefly discuss how
a suitable degree distribution can be obtained for our setup and then compare the performance with our
segmentation-based scheme.

We note in advance that codes designed this way do not include the segmentation-based scheme of
Theorem 1 as a special case. This is because in our rateless code construction, the choice of non-zero
source symbol coefficients is done uniformly over the entire source sequence. In contrast, each parity bit
in the segmentation-based scheme is generated from source bits restricted to a certain segment.

There are, however, existing rateless codes that send batches of linear combinations generated from
select subsets of source packets, such as the chunked codes of [57]. However, this work’s consideration
in restricting the scope of linear combinations is largely due to concerns in computational complexity
rather than optimizing non-uniform partial recovery constraints. To the best of our knowledge, the
LT-based rateless code designs investigated in [18, 19] are the only ones in the existing literature for the
scenario we consider. Hence, we only include an enhanced version of the designs given in [18, 19] as a
representative design for comparison with the segmentation-based scheme.

3.4.1 Rateless Coding Approach

In this subsection, we describe the main difference in our present approach compared to [18], [19], which is
the way we handle degree-1 symbols. In previous works, the degree-1 symbols were sampled uniformly at
random. This resulted in many repetitions where the same source symbol would be transmitted multiple
times while others would not be transmitted at all. Thus, our current work proposes an alternative
that chooses these symbols deterministically in a round-robin fashion. Suppose that Nz transmissions
of source symbols have finished, where Nz is a natural number and z is a positive, rational number. If
a single source symbol is sent uncoded T times over a channel with erasure rate ϵ, the probability that
it is recoverable after these transmissions is (1 − ϵT). We have that after Nz round-robin
transmissions of source symbols, a fraction of (z − \lfloor z \rfloor) source symbols were transmitted
(\lfloor z \rfloor + 1) times, while the remaining (1 − (z − \lfloor z \rfloor)) fraction was transmitted only \lfloor z \rfloor times. The average fraction of recovered
symbols is therefore given by φ(z, ϵ),

\[
φ(z, ϵ) = (1 − (z − \lfloor z \rfloor))(1 − ϵ^{\lfloor z \rfloor}) + (z − \lfloor z \rfloor)(1 − ϵ^{\lfloor z \rfloor + 1}).
\] (3.8)

Following [19], we can express the optimal degree distribution that minimizes the maximum latency
as follows.

\[
\min_{w, p_1, \ldots, p_N} w
\]

subject to

\[
\log(1 − x) − \log(1 − φ(wp_1, ϵ_i)) + (1 − ϵ_i)w \sum_{j>1} j p_j x^{j−1} > 0,
\]

∀x ∈ (0, 1 − di), i = 1, 2,

where the probabilities satisfy \sum_j p_j = 1 and p_j ≥ 0, and we recall that di, and ϵi denote the distortion
and erasure probabilities for the two users. To interpret the above expression, note that the left-hand-side, when multiplied by 1 − x, is proportional to the size of the ripple [58] induced in the belief
propagation decoding process when a fraction of x source symbols have been recovered. Hence, the
constraint ensures that the ripple remains non-empty until a fraction of 1 − di source symbols have
been recovered, which in turn ensures a distortion smaller than di. Using the approach in [19], we can
numerically compute the optimal degree distribution by using a linear programming approach. We omit
the details due to space constraints.

3.4.2 Numerical Results

Fig. 3.6 plots the latency vs. d2 with the rest of the parameters, i.e., d1, ϵ1, and ϵ2 fixed. We plot the
outer (lower) bound w_M = \max{\frac{1−d_1}{1−\epsilon_1}, \frac{1−d_2}{1−\epsilon_2}} together with the latency achieved by the segmentation-based scheme of Theorem 1, and the optimal latency achievable by a code designed through (3.9). We refer to this plot as the LT-based scheme due to the similarities with LT codes [55]. Alongside these
curves, we plot the convex hull of the latencies achieved with the LT-based scheme and denote this as
the “timesharing” curve in Fig. 3.6.
We observe that there are two regions where Theorem 1 meets the outer bound. The first is where \( d_2 \geq \epsilon_2 = 0.4 \), and the other is where \( d_2 \leq d_1\epsilon_2/\epsilon_1 \approx 0.13 \) (see Section 3.3.3 for a more detailed discussion of these regions). Note that there is a considerable gap between the degree-optimized rateless codes and the segmentation-based scheme. The LT-based scheme forces the code to have a single degree distribution from which each coded bit is sampled. The segmentation-based scheme, however, applies a different code to each of the segments and hence provides greater flexibility to simultaneously satisfy each user’s demand. Note that in Fig. 3.6, the LT-based scheme is optimal as \( d_2 \to 0 \), but the gap increases as the distortion increases. We also observe in numerical experiments that for small \( d_2 \) (up to around 0.2 in Fig. 3.6), the optimal latency of (3.9) is achieved when the degree distribution is designed for user 2 only, oblivious of user 1. This, to some extent, echoes the segmentation-based scheme when \( d_2 \leq d_1\epsilon_2/\epsilon_1 \approx 0.13 \) as discussed above.

### 3.5 Individual Decoding Delays

In this section, we consider possible orderings for the transmission of the (minmax) latency-optimal segments given in Theorem 1. In doing so, we will observe the subsequent effect this has on individual decoding delays, which is of practical interest. For clarity of exposition, we do this by revisiting the numerical example given in Sections 3.3.3 and 3.3.4 and comparing two possible segment orderings. We mention, however, that the procedure we follow for our derivation is not dependent on this example and is easily generalizable. The intention of comparing these orderings is to illustrate just how challenging it can be to schedule the transmission of segments for a particular metric. While it is not an exhaustive treatment of all possible orderings, our hope is that the insight gained through this example will trigger further interest and research.

Now, recall that for this example, we have that \( \mathcal{J} = [n] \). In turn, this implies that each user’s distortion constraint in (3.5) is tight since in general, any user in \( \mathcal{J} \) will have their distortion constraint met with equality. This fact can be verified by combining (3.5) and (3.7). The consequence of this is that each user will need to receive a portion of every segment, a fact which will be taken into account.
when considering possible segment orderings.

Before we begin discussing some possible orderings, however, let us first consider the process involved in transmitting a length-$N a_i$ source segment $S_i$ for $i \in [n]$. Given that this segment is channel coded with a rate-$(1 - \epsilon_i)$ code to obtain the channel input $X_i$, we see that $W_{ii}$ channel uses are required to transmit the segment where

$$Na_i = W_{ii}(1 - \epsilon_i). \quad (3.10)$$

Since the channel code is systematic, the $W_{ii}$ channel uses consists of a length-$N a_i$ portion of the original source symbols in $S_i$ followed by a length-$p_{ii}$ portion of parity symbols where

$$p_{ii} = W_{ii} - Na_i = \frac{N \epsilon_i}{1 - \epsilon_i} a_i. \quad (3.11)$$

We denote the length-$p_{ii}$ portion of parity symbols in $X_i$ as $P_i$. This partitioning into systematic and parity components is depicted in Fig. 3.7.

Notice, however, that user $i$ is the only user who must listen for the entire $W_{ii}$ channel uses. For $j \in \{i + 1, i + 2, \ldots, n\}$, user $j$ in fact cannot decode the entire segment $S_i$ and instead relies only on what he can obtain from the systematic portion. He can therefore stop listening after $N a_i$ channel uses. On the other hand, for $k \in \{1, 2, \ldots, i - 1\}$, since $\epsilon_k < \epsilon_i$, user $k$ can decode segment $S_i$ by listening to only $W_{ik} < W_{ii}$ channel uses where

$$W_{ik}(1 - \epsilon_k) = W_{ii}(1 - \epsilon_i). \quad (3.12)$$

The earlier decoding times $W_{i1}$ and $W_{i2}$ for users 1 and 2 are also shown in Fig. 3.7.

In light of these facts, we will treat the systematic portion of each channel coded segment as a common requirement for all users. In the next two subsections, we will therefore consider orderings that begin with uncoded transmissions. That is, we will first send the length-$N a_0$ segment $S_0$ uncoded and subsequently isolate and transmit the systematic component of $X_i$ for $i \in [n]$. This requires a total of $N(a_0 + a_1 + \ldots + a_n) = N$ transmissions, where we have used the fact that the source segments partition the entire source sequence (see (3.7) and Lemma 19).

The entire source sequence is therefore sent over the first $N$ channel uses, and the only remaining task is to determine the subsequent ordering of the $n$ parity components $P_i$ for $i \in [n]$. This option of ordering parity components provides much flexibility to a content provider. For example, he can make any user $k \in [n]$ able to decode at a latency that is point-to-point optimal. We again note that for $i \in \{k + 1, k + 2, \ldots, n\}$, user $k$ does not have to receive the entire $p_{ii}$ parity symbols of $X_i$. He can
3.5.1 Parity Segments Sent in Decreasing Order

In this subsection, we consider the case when $P_i$, the parity for segment $S_i$, is sent in decreasing order of $i$. That is, we first transmit $P_n$ followed by $P_{n-1}$ to $P_1$ (see Fig. 3.8a). We will calculate the excess latency each user experiences with this ordering. The excess latency is defined relative to the point-to-point optimal latency, $w_k^*$, which is given for user $k$ by

$$w_k^* = 1 - d_k 1 - \epsilon_k.$$  \hfill (3.14)

Given that user $k$ achieves a latency of $w_k$, we then define his excess latency, $\delta_k$, to be

$$\delta_k = w_k - w_k^*.$$  \hfill (3.15)

To calculate $\delta_k$, we first remind the reader that for the example we are considering, user $k$ requires parities from $P_i$ for $i \in \{k, k+1, \ldots, n\}$ but does not require any parities from $P_j$, $j \in \{1, 2, \ldots, k-1\}$, since they are intended for users with better channel qualities. He can therefore meet his distortion constraint after $P_k$ is sent (see Fig. 3.8a). We recall from the previous section that user $k$ needs to listen to only $p_{ik}$ of the $p_{ii}$ symbols in $P_i$. By combining (3.11) and (3.13), we see that the excess latency incurred by listening to the full $p_{ii}$ parity symbols is therefore cumulatively given by
\[ \delta_k = \frac{1}{N} \sum_{i=k+1}^{n} (p_{ii} - p_{ik}) \]  
(3.16)

\[ = \sum_{i=k+1}^{n} \left( \frac{\epsilon_i}{1 - \epsilon_i} - \frac{\epsilon_k}{1 - \epsilon_k} \right) a_i. \]  
(3.17)

Hence, the latency tuple \((w_i^* + \delta_1, w_i^* + \delta_2, \ldots, w_i^* + \delta_n)\) is \((d_1, d_2, \ldots, d_n)\)-achievable, where the \(a_i\)'s that appear in (3.17) are given by Theorem 1 for \(i \in \{0, 1, \ldots, n\}\).

### 3.5.2 Parity Segments Sent in Increasing Order

In this subsection, we consider the case when \(P_i\), the parity for segment \(S_i\), is sent in increasing order of \(i\). That is, we first transmit \(P_1\) followed by \(P_2\) to \(P_n\) (see Fig. 3.8b). We will again calculate the excess latency user \(k\) experiences with this ordering, which we will denote this time by \(\Delta_k\).

In calculating \(\Delta_k\), we again observe that the first \(k-1\) parities, \(P_1, P_2, \ldots, P_{k-1}\), are useless to user \(k\) since they are intended for users with better channel qualities. For \(j \in \{1, 2, \ldots, k-1\}\), the excess latency for each of these segments is thus \(p_{j,k}\).

In contrast, user \(k\) does require parities from \(P_i\) for \(i \in \{k, k+1, \ldots, n\}\). For each of these parities, we can again derive the excess latency incurred as being \((p_{ii} - p_{ik})\). Notice, however, that user \(k\) is not forced to listen to the full amount of parity symbols for \(P_n\). Since this is the last parity segment sent, he can actually decode after listening to \(p_{nk}\) of these symbols, and so there is no excess latency incurred from \(P_n\) (see Fig. 3.8b). The cumulative excess latency is therefore given by

\[ \Delta_k = \frac{1}{N} \left( \sum_{i=1}^{k-1} p_{ii} + \sum_{i=k+1}^{n-1} (p_{ii} - p_{ik}) \right) \]  
(3.18)

\[ = \sum_{i=1}^{k-1} \frac{\epsilon_i}{1 - \epsilon_i} a_i + \sum_{i=k+1}^{n-1} \left( \frac{\epsilon_i}{1 - \epsilon_i} - \frac{\epsilon_k}{1 - \epsilon_k} \right) a_i. \]  
(3.19)

Again, the latency tuple \((w_i^* + \Delta_1, w_i^* + \Delta_2, \ldots, w_i^* + \Delta_n)\) is therefore \((d_1, d_2, \ldots, d_n)\)-achievable, where the \(a_i\)'s that appear in (3.19) are given by Theorem 1 for \(i \in \{0, 1, \ldots, n\}\).

### 3.5.3 A Numerical Comparison of Orderings

We now compare the individual latencies achieved with the orderings proposed in Sections 3.5.1 and 3.5.2. We do the comparison for the example discussed in Sections 3.3.3 and 3.3.4, where each user \(i\)'s distortion is quadratic in his erasure rate, i.e., \(d_i = \epsilon_i^2\) for \(i \in \{1, 2, \ldots, 5\}\).

Let \(\epsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_5)\) and \(d_1 = (d_1, d_2, \ldots, d_5)\). In the first example of Fig. 3.9a, we again take \(\epsilon = (0.1, 0.2, 0.3, 0.4, 0.5)\) and \(d_1 = (0.01, 0.04, 0.09, 0.16, 0.25)\). In this figure, each user is shown on the horizontal axis and the individual latency he achieves is plotted on the vertical axis. Each user’s point-to-point optimal latency, as given by (3.14), is also shown so that the excess latency can easily be inferred.

From this figure, we see that the sum excess latency is lower when the parities are sent in increasing order. At first, this may seem counterintuitive since when \(P_i\) are transmitted in increasing order of \(i\), a user \(k\) has no use for parities \(P_j\) for \(j \in \{1, 2, \ldots, k-1\}\) and essentially postpones the decoding process until the transmission of these parities is complete (see Fig. 3.8b). On the other hand, when \(P_i\) are sent
the lengths of all coded segments are equal. Specifically, we set \( \epsilon = (0.1, 0.2, 0.3, 0.4, 0.5) \) and \( d = d_1 = (0.01, 0.04, 0.09, 0.16, 0.25) \). In (b), we take \( \epsilon = (0.31, 0.32, 0.33, 0.34, 0.35) \) and \( d_3 = (0.155, 0.192, 0.231, 0.272, 0.315) \). The point-to-point optimal latency for each user, as given in (3.14), is also shown.

Figure 3.9: The individual latency of each user for the segment orderings of Sections 3.5.1 and 3.5.2. In (a), we set \( \epsilon = (0.1, 0.2, 0.3, 0.4, 0.5) \) and \( d = d_1 = (0.01, 0.04, 0.09, 0.16, 0.25) \). In (b), we take \( \epsilon = (0.31, 0.32, 0.33, 0.34, 0.35) \) and \( d_3 = (0.155, 0.192, 0.231, 0.272, 0.315) \). The point-to-point optimal latency for each user, as given in (3.14), is also shown.

In *decreasing* order of \( i \), user \( k \) has already finished decoding by the time any parities \( P_j, j \in [k - 1] \), are sent (see Fig. 3.8a). The lengths of the parities in Figures 3.8a, and 3.8b were drawn only for convenience, however, as \( p_{ii} \), the number of parity symbols in \( P_i \), will generally vary depending on \( i \) (see (3.11)). As discussed in Section 3.5.2, the ability for certain users to avoid receiving the entire \( p_{55} \) parities of \( P_5 = P_n \) is the other important benefit in this example as \( P_5 \) happens to be the longest of all parity segments.

In contrast, Figure 3.9b plots when the distortions and erasure rates have been chosen such that the lengths of all coded segments are equal. Specifically, we set \( \epsilon = (0.31, 0.32, 0.33, 0.34, 0.35) \) and \( d_3 = (0.155, 0.192, 0.231, 0.272, 0.315) \) so that \( (a_1, a_2, a_3, a_4, a_5) = (0.1, 0.1, 0.1, 0.1, 0.1) \). The erasure rates were also chosen within a short interval so that users experience similar channel qualities. In turn, the excess latency that stronger users incur when listening to parities of weaker users is small. Thus, each term in (3.16) is small and the excess latency for sending parities in decreasing order is minimal. On the other hand, differing channel qualities does not account for the entire excess latency when sending parities in *increasing* order. There is also the excess latency incurred by beginning transmission with parities that are not decodable for certain users, which is represented by the first summation in (3.18). We see then that in Figure 3.9b, the sum excess latency is lower when sending parities in *decreasing* order.

### 3.6 Conclusions

In this chapter, we proposed a successive segmentation-based coding scheme for broadcasting a binary source over a multi-receiver erasure broadcast channel. Each receiver has individual distortion constraints and experiences distinct channel erasure rates. The proposed scheme partitions the source sequence into multiple segments and applies a systematic erasure code to each segment. We provided optimal choices for segment sizes and code rates for each segment, which were based on the users’ channel erasure rates, and distortion constraints.

Not only does this proposed scheme outperform rateless codes and network coding, it also has two
other practical advantages, namely simplicity and scalability. Firstly, it uses only off-the-shelf systematic erasure codes rather than a joint source-channel code, which would otherwise be required for optimality. Secondly, it can easily be adjusted as users are added or deleted from the system and thus scales to an arbitrary number of users while retaining optimality.

We also discussed the effects that segment transmission orderings has on the decoding latencies of individual users. We provided closed-form expressions for each individual user’s excess latency when parity check bits are successively transmitted in both increasing and decreasing order of their segment’s coded rate. We then demonstrated how each of the two orderings could be more favourable than the other in terms of incurring a smaller average individual latency.

For future work, it is our interest to conduct a thorough analysis of individual latencies achieved by users in our segmentation-based scheme. We would also like to analyze the segmentation-based scheme for finite block-lengths and extend the scheme for multiple-description-coded Gaussian sources.
Chapter 4

Converse via Multiple Descriptions

We consider the problem of deriving an outer bound (converse), for the erasure source-broadcast problem without feedback when there are only two receivers. The system model we study is the same as in Chapter 3, however our derivation makes two additional assumptions. The first assumption is that we consider the case when \( M \triangleq 1/(1-\epsilon_2) \) is an integer. The second assumption is that we consider only the class of non-erasure-randomized codes. As we explain in Section 4.4, this is the class of codes for which the positions of erasures in the source reconstruction is determined only by the channel noise realization.

The outer bound we present is parameterized by the distortion of user 1, the stronger user. We assume that \( D_1 \), the distortion achieved by user 1, is given by \( D_1 = D^*(\epsilon_1) + \delta \), where \( \delta \in [0, \epsilon_1 - D^*(\epsilon_1)) \), and \( D^*(\epsilon_i) \) is the point-to-point optimal distortion for user \( i \), given by

\[
D^*(\epsilon_i) = 1 - b(1 - \epsilon_i),
\]

where \( b = n/m \) is the number of channel uses per source symbol, i.e., the bandwidth expansion factor.

4.1 Prior Work

The initial progress in deriving an outer bound for the source-broadcast problem was in the seminal work of [4]. In their work, the authors considered a Gaussian source being sent over a two-receiver, Gaussian broadcast channel with bandwidth expansion and a mean-squared error distortion constraint. In deriving the outer bound, an auxiliary random variable was introduced. The technique of introducing an auxiliary random variable to derive an outer bound was previously used for the problem of quadratic Gaussian multiple description coding [59]. The source-broadcast outer bound was parameterized by the variance of the auxiliary random variable and by optimizing over the domain of the variance, an outer bound could be found that improved on the cutset bound.

By using the techniques of [4], an analogous outer bound was derived for the problem of transmitting a binary source over an erasure broadcast channel with Hamming distortion in [53]. However, when the Hamming distortion constraint is replaced with that of an erasure distortion constraint, the introduction of an auxiliary random variable as in [4] did not result in an outer bound that improved on the cutset bound.

The interpretation of the auxiliary random variable of [4] was given an operational meaning in [50] where the authors generalized the source-broadcast outer bound of [4] for general sources, channels,
4.2 Main Idea

4.2.1 A Toy Example

We first give a toy example that outlines a plausible argument against the possibility of two users simultaneously achieving point-to-point optimal distortions for the erasure source-broadcast problem of broadcasting a binary source over an erasure broadcast channel with bandwidth mismatch and an erasure distortion constraint.

Problem Definition

The problem is shown in Figure 4.1. We consider two source symbols \((S_1, S_2)\) sampled i.i.d. from a Bern(1/2) distribution that are to be sent to two users over three channel uses \((X_1, X_2, X_3)\), of an erasure broadcast channel. The first user, who will be called the stronger user, listens over a channel and distortion measures. In their work, the authors first found the distortion region for broadcasting a bivariate source over a broadcast channel given that one of the receivers had one of the source sequences as side information. They then used a reduction argument to derive an outer bound for the source-broadcast problem, which when particularized for the quadratic Gaussian case, coincided with the outer bound of [4]. Similarly, this outer bound coincided with the one in [53] when specialized for the binary source, erasure broadcast channel and Hamming distortion and did not improve upon the cutset bound if the Hamming distortion was replaced with the erasure distortion. The authors in [60] have also extended the outer bound of [4] to general sources, distortion measures, and broadcast channels, however their techniques again do not provide a non-trivial outer bound for the erasure source-broadcast problem.

In subsequent work to generalize the results of [4], the authors of [5] also extended the outer bound for the case of \(N\) receivers in the Gaussian broadcast channel. Finally, although matching inner and outer bounds have not been derived for the source-broadcast problem, the authors of [49] have bounded the loss in optimality of simply using a separation-based time-sharing inner bound.

In terms of outer bounds for problems related to the source-broadcast problem, the authors of [6] found an exact characterization for broadcasting a bivariate Gaussian source over the Gaussian broadcast channel without any receiver side information. Finally, there have also been outer bounds derived for the related problems mentioned in Chapter 1 of symmetrical multilevel diversity coding [27, 28], and erasure multiple descriptions when considering the worst-case distortion [25].
that is guaranteed to have exactly one of the channel symbols erased. The identification of which symbol
\((X_1, X_2 \text{ or } X_3)\) is actually erased is not known \textit{a priori}, however the probability of each of the three
outcomes is equally likely. Similarly, the second user, who will be called the weaker user, listens over a
channel that is guaranteed to have two of the three symbols erased, i.e., only one symbol is not erased.
The probability that \(X_i\) is not erased is equally likely for all \(i \in \{1, 2, 3\}\). The stronger user requires the
two source symbols to be recovered losslessly while the weaker user is only interested in obtaining the
minimum possible distortion given the stronger user’s lossless requirement.

**Intuition**

Since the stronger user requires lossless reconstruction, we intuitively expect the transmitter to encode
the symbols in a manner similar to having

\[
\begin{bmatrix}
X_1 \\
X_2 \\
X_3
\end{bmatrix} = \begin{bmatrix}
S_1 \\
S_2 \\
S_1 \oplus S_2
\end{bmatrix},
\]

(4.2)

so that two source symbols can be reconstructed if any two channel symbols are received. However,
if this is the case, we see that in one of the three equiprobable erasure patterns on the \textit{weaker} user’s
channel, there is a possibility that both of the symbols \(X_1\) and \(X_2\) in (4.2) are erased so that only
\(X_3 = S_1 \oplus S_2\) is received. In this case, the weaker user can neither reconstruct \(S_1\) nor \(S_2\) and so
must output an erasure at both positions. It is trivial to see that in the two remaining channel erasure
patterns in which only \(X_1\) or only \(X_2\) is received, the weaker user can reconstruct one of the two source
symbols. Conditioned on these three erasure patterns, the average distortion achieved by the weaker
user is therefore \(D_2 = (1/3) \cdot 1/2 + (1/3) \cdot 1/2 + (1/3) \cdot 1 = 2/3\), which is greater than the point-to-point
optimal distortion of \(D^*_2 = 1/2\).

**4.2.2 Outline of Proof**

In the following sections, we extend this argument to the general erasure source-broadcast problem for
non-erasure-randomized codes. Our argument is inspired by the techniques of [61], where the authors
studied the modulo-sum capacity of an erasure multiple-access channel. In their argument, the authors
partitioned the channel input into sets and derived necessary conditions for the individual sets. They
then argued by symmetry that any achievable coding scheme should also be functional if the partitioned
sets were interchanged, which lead to the derivation of another group of non-trivial necessary conditions.

In our work, we similarly first partition the source sequence \(X^n\) into \(M\) disjoint sets of channel
symbols that could hypothetically be received in a virtual channel output. Each virtual channel output
contains \(n(1 - \epsilon_2)\) channel symbols from \(X^n\) on average and has the same marginal distribution as the
weaker user’s actual channel output. We then derive necessary conditions for the virtual channel outputs
individually as well as for when we combine virtual channel outputs. In particular, the weaker user is
able to reconstruct the source given any virtual channel output, and the stronger user should reconstruct
the source given the channel input symbols contained in a \textit{set} of virtual channel outputs.

In Lemma 4 we bound the conditional mutual information between any two virtual channel outputs
as a function of the stronger user’s distortion. If the stronger user’s distortion is small, we show that the
conditional mutual information between virtual channel outputs should also be small. This is because if
the virtual channel outputs are highly correlated, there is redundancy between them and if the stronger user receives a superset of the channel input symbols in the two channel outputs, he will not be able to receive enough new information to achieve a small distortion.

To illustrate, in our example above, we have $M = 3$, and we see that if the stronger user is lossless, $X_i$ must be independent of $X_j$ for all $i \neq j$. Otherwise, the result of receiving any two channel symbols will not give enough information to reconstruct two source symbols.

In our proof, we define a non-erasure-randomized code such that given the channel erasure pattern, the positions of the erasures in the source reconstruction are determined. In Lemma 7, we then show that such codes have the property that the conditional mutual information between one virtual channel reconstruction and a set of other virtual channel reconstructions is upper-bounded by a sum of pairwise mutual information terms.

In our main result, we then sum $M$ versions of the rate-distortion theorem inequality, one for each virtual channel output in (4.70). Hypothetically, the incorporation of each inequality should increase the number of symbols able to be reconstructed on the right-hand-side of (4.70). We then use the chain rule to represent the sum as a joint entropy term and a summation of conditional mutual information terms in (4.71). We then use the property of non-erasure-randomized codes to upper-bound the sum of mutual informations terms with respect to a sum of pairwise mutual information terms. Finally, we use Lemma 4 to show that if the pairwise mutual information terms are small, as required by the stronger user’s distortion, then some virtual channel outputs cannot be used to reconstruct the source for the weaker user, as in the toy example given in Section 4.2.1.

4.3 Virtual Channel Outputs

4.3.1 Construction of Virtual Channel Outputs

Definition 5. Let $U^n = (U_1, U_2, \ldots, U_n)$ be a sequence of i.i.d. random variables where each $U_i$ is uniformly distributed over the set $[M] \triangleq \{1, 2, \ldots, M\}$.

We define a virtual channel noise realization, $\tilde{N}_i^n$, based on $U^n$ as follows.

Definition 6. Let $U^n$ be a random variable as defined in Definition 5. For every $i \in [M]$, we define the virtual channel noise realization $\tilde{N}_i^n$ as an $n$-symbol vector such that for all $k \in [n]$, $\tilde{N}_i^n(k) = 0$ if $U^n(k) = i$, else $\tilde{N}_i^n(k) = 1$ if $U^n(k) \neq i$.

We associate a virtual channel output $\tilde{Y}_i^n$ with any virtual channel noise realization as per the following definition.

Definition 7. Let $\tilde{N}_i^n$ be a random variable as defined in Definition 6 for some $i \in [M]$. We define the virtual channel output $\tilde{Y}_i^n$ as an $n$-symbol vector such that for all $k \in [n]$, $\tilde{Y}_i^n(k) = X^n(k)$ if $\tilde{N}_i^n(k) = 0$, else $\tilde{Y}_i^n(k) = \star$ if $\tilde{N}_i^n(k) = 1$.

Definition 8. Let $\eta^n \in \{0, 1\}^n$. We define $\mathcal{I}(\eta^n) = \{k \in [n] \mid \eta(k) = 0\}$ as the set of indices in $\eta^n$ that indicate an erasure did not occur.

4.3.2 Properties of the Virtual Channel Outputs

We derive some properties of the $\tilde{Y}_i^n$ below.
Proposition 1. Let $Y^n_2$ be the channel output of user 2, and let $i \in \{1, 2, \ldots, M\}$. Then for all $x^n \in \{0, 1\}^n$, and $y^n \in \{0, 1, \ast\}^n$, $\Pr(Y^n_2 = y^n | X^n = x^n) = \Pr(\tilde{Y}^n_i = y^n | X^n = x^n)$.

Proof. Given $X^n$, by construction, $\tilde{Y}^n_i$ is an i.i.d. sequence where, similar to $Y^n_2$, for every $k \in [n]$, $\tilde{Y}^n_i(k)$ is not erased if $U^n(k) = i$, which happens with probability $1/M = 1 - \epsilon_2$.

Lemma 1. Let $\tilde{N}^n_i$ and $\tilde{Y}^n_i$ be defined as in Definitions 6, and 7 respectively for some $i \in [M]$, and let $\bar{U}$ be a set of arbitrarily correlated random variables. Then $H(\tilde{Y}^n_i | \tilde{N}^n_i, \bar{U}) \leq n(1 - \epsilon_2)$.

Proof.

$$H(\tilde{Y}^n_i | \tilde{N}^n_i, \bar{U}) = \sum_{j=1}^{n} H(\tilde{Y}^n_i(j) | \tilde{N}^n_i(1), \tilde{Y}^n_i(2), \ldots, \tilde{Y}^n_i(j-1), \tilde{N}^n_i, \bar{U})$$

$$(a) \leq \sum_{j=1}^{n} H(\tilde{Y}^n_i(j) | \tilde{N}^n_i(j))$$

$$= \sum_{i=1}^{n} H(X^n(j) | \tilde{N}^n_i(j) = 0)^{1/M}$$

$$(b) \leq \sum_{i=1}^{n} H(X^n(j) | \tilde{N}^n_i(j) = 0)(1 - \epsilon_2)$$

$$(c) \leq \sum_{i=1}^{n} (1 - \epsilon_2)$$

$$= n(1 - \epsilon_2)$$

where

(a) follows from the fact that conditioning reduces entropy

(b) follows from the fact that $M = 1/(1 - \epsilon_2)$

(c) follows from the fact that the uniform distribution maximizes the entropy of a discrete random variable.

Definition 9. Let $A \subseteq [n]$. We define $X_A = \{X^n(j) | j \in A\}$.

Definition 10. Let $A_i$ be random variables for $i \in [n]$, and let $B \subseteq [n]$. We define $\cup_{i \in B} A_i = \{A_i | i \in B\}$.

Definition 11. Let $A_i$ be random variables for $i \in [n]$, and let $B \subseteq [n]$. We define $H(\cup_{i \in B} A_i)$ as the joint entropy of the random variables in $\cup_{i \in B} A_i$.

Lemma 2. Let $B \subseteq [M]$, and $\hat{n}^n_j \in \{0, 1\}^n$ for all $j \in B$. Let $C = \{k \in [n] | \hat{n}^n_j(k) = 1 \text{ for all } j \in B\}$, i.e., $C$ is the set of indices for which there was an erasure for all $\tilde{Y}^n_i$ where $i \in B$. Then $H(\cup_{j \in B} \tilde{Y}^n_j | \cup_{j \in B} \tilde{N}^n_j = \hat{n}^n_j) = H(X_{\overline{C}})$, where $\overline{A}$ denotes the complement of set $A$.

Proof.

$$H(\cup_{j \in B} \tilde{Y}^n_j | \cup_{j \in B} \tilde{N}^n_j = \hat{n}^n_j) = H(X_{\overline{C}} | \cup_{j \in B} \tilde{N}^n_j = \hat{n}^n_j)$$

$$(a) = H(X_{\overline{C}})$$
Proof. Given any $k$ is neither placed in $\tilde{W}_i$, which are determined from $U^n$. \hfill \square

Lemma 3. Let $P^n = (P_1, P_2, \ldots, P_n)$ and $Q^n = (Q_1, Q_2, \ldots, Q_n)$ each be a sequence of i.i.d. random variables where for $i \in [n]$, $P_i$ and $Q_i$ are Bernoulli random variables over the support $\{0, 1\}$ such that $\Pr(P_i = 1) = \hat{p}$, and $\Pr(Q_i = 1) = \hat{q}$, where $0 \leq \hat{p} \leq \hat{q} \leq 1$.

Let $E_{P^n}X = \sum_{p^n \in \{0,1\}^n} H(X_{\mathcal{I}(p^n)}) \Pr(P^n = p^n)$ and $E_{Q^n}X = \sum_{q^n \in \{0,1\}^n} H(X_{\mathcal{I}(q^n)}) \Pr(Q^n = q^n)$ where $\mathcal{I} : \{0,1\}^n \to \mathcal{P}([n])$ is defined in Definition 8. Then (a) $E_{P^n}X \leq E_{Q^n}X + n(\hat{q} - \hat{p})$, (b) $E_{Q^n}X \leq E_{P^n}X$.

Proof. Given any $p^n$, let $\hat{U}^n(p^n) = (\hat{U}_1, \hat{U}_2, \ldots, \hat{U}_n)$ be a sequence of random variables where for $k \in [n]$, each $\hat{U}_k \in \{0,1,2\}$ is a random variable whose distribution depends only on $p^n(k)$, the specifics of which will be described shortly. Given $\hat{U}^n(p^n) = u^n$, we use $u^n$ to partition the set $\mathcal{I}(p^n)$ into sets $\mathcal{W}_1(u^n) \subseteq [n]$, and $\mathcal{W}_2(u^n) \subseteq [n]$. Specifically, for $k \in [n]$, if $p^n(k) = 0$, we have that $k \in \mathcal{I}(p^n)$, and conditioned on this fact, we place $k$ in $\mathcal{W}_1(u^n)$ if $u^n(k) = 1$, else, we place $k$ in $\mathcal{W}_2(u^n)$ if $u^n(k) = 2$. We do not place $k$ in either $\mathcal{W}_1(u^n)$ or $\mathcal{W}_2(u^n)$ if $u^n(k) = 0$. We set the probability of these events to be

\begin{align*}
\Pr(\hat{U}_k = 0 | p^n(k) = 0) &= 0 \quad (4.11) \\
\Pr(\hat{U}_k = 1 | p^n(k) = 0) &= (1 - \hat{q})/(1 - \hat{p}) \quad (4.12) \\
\Pr(\hat{U}_k = 2 | p^n(k) = 0) &= 1 - (1 - \hat{q})/(1 - \hat{p}). \quad (4.13)
\end{align*}

On the other hand, if $p^n(k) = 1$, we have that $k \not\in \mathcal{I}(p^n)$, and we set $\Pr(\hat{U}_k = 0 | p^n(k) = 1) = 1$ so that $k$ is neither placed in $\mathcal{W}_1(u^n)$ nor $\mathcal{W}_2(u^n)$.

Given $p^n$, we can view $\hat{U}^n(p^n)$ as a sequence of biased coin flips, where the bias is determined by $p^n$. If we were to first choose which biased coin to flip by selecting $p^n$ at random, the resulting process would yield a sequence of coin flips that we define as $U^n$. That is, $U^n$ is a sequence of random variables where the sequence is determined by first realizing a value for $P^n$ and then realizing a value for $\hat{U}^n(p^n)$ given $P^n = p^n$. 

(a) follows from the fact that the channel input is independent of the virtual channel noises $\hat{N}_i^n$, which
With this notation, we can write that

\[
\mathbb{E}_{P^n}X = \sum_{p^n \in \{0,1\}^n} \left( H(X_{I(p^n)}|\hat{U}^n(p^n)) + I(X_{I(p^n)};\hat{U}^n(p^n)) \right) \Pr(P^n = p^n)
\]

\[= \sum_{p^n \in \{0,1\}^n} H(X_{I(p^n)}|\hat{U}^n(p^n)) \Pr(P^n = p^n) \quad (4.14) \]

\[= \sum_{p^n \in \{0,1\}^n} \sum_{u^n \in \{0,1,2\}^n} H(X_{I(p^n)}|\hat{U}^n(p^n) = u^n) \Pr(P^n = p^n) \Pr(\hat{U}^n(p^n) = u^n|P^n = p^n) \quad (4.15) \]

\[= \sum_{p^n \in \{0,1\}^n} \sum_{u^n \in \{0,1,2\}^n} H(X_{W_1(u^n)}, X_{W_2(u^n)}) \Pr(\hat{U}^n = u^n) \quad (4.16) \]

\[= \sum_{u^n \in \{0,1,2\}^n} H(X_{W_1(u^n)}, X_{W_2(u^n)}) \Pr(\hat{U}^n = u^n) \quad (4.17) \]

\[= \sum_{u^n \in \{0,1,2\}^n} H(X_{W_1(u^n)}) \Pr(\hat{U}^n = u^n) + \sum_{u^n \in \{0,1,2\}^n} H(X_{W_2(u^n)}|X_{W_1(u^n)}) \Pr(\hat{U}^n = u^n) \quad (4.18) \]

\[= \sum_{\tilde{q}^n \in \{0,1\}^n} H(X_{I(\tilde{q}^n)})f(\tilde{q}^n) + \sum_{u^n \in \{0,1,2\}^n} H(X_{W_2(u^n)}|X_{W_1(u^n)}) \Pr(\hat{U}^n = u^n), \quad (4.19) \]

where

(a) follows from the fact that conditioned on \(p^n\), \(\hat{U}^n(p^n)\) is a random sequence independent of the channel input \(X^n\)

(b) follows from the fact that \(W_1(u^n)\) and \(W_2(u^n)\) partition \(I(p^n)\) given \(u^n\)

(c) follows from the definition of \(\hat{U}^n\)

(d) we have defined \(f(\tilde{q}^n) = \sum_{u^n \in \mathcal{U}(\tilde{q}^n)} \Pr(\hat{U}^n = u^n)\) and \(\mathcal{U}(\tilde{q}^n) = \{ u^n \in \{0,1,2\}^n \mid I(\tilde{q}^n) = W_1(u^n) \}\).

We now show that the first summation of (4.21) is equal to \(\mathbb{E}_{Q^n}X\) by showing that \(f(\tilde{q}^n) = \Pr(Q^n = \tilde{q}^n)\). By construction, \(k \in W_1(u^n)\) if \(\hat{U}^n(k) = 1\), which happens with probability

\[
\Pr(\hat{U}^n(k) = 1) = \Pr(\hat{U}_k = 1|p^n(k) = 0) \Pr(p^n(k) = 0) \quad (4.22)
\]

\[= 1 - \tilde{q}, \quad (4.23) \]

where (4.23) follows from (4.12). On the other hand, if \(k \notin W_1(u^n)\), then we must have \(\hat{U}^n(k) = 0\), or \(\hat{U}^n(k) = 2\), which happens with probability \(\tilde{q}\). Since \(\hat{U}^n(k)\) is i.i.d. for all \(k \in [n]\), it is not hard to see that the summation in the expression for \(f(\tilde{q}^n)\) factorizes such that \(f(\tilde{q}^n) = \Pr(Q^n = \tilde{q}^n)\). We can therefore continue from (4.21) by writing

\[
\mathbb{E}_{P^n}X = \mathbb{E}_{Q^n}X + \sum_{u^n \in \{0,1,2\}^n} H(X_{W_2(u^n)}|X_{W_1(u^n)}) \Pr(\hat{U}^n = u^n) \quad (4.24)
\]

\[\leq \mathbb{E}_{Q^n}X + \sum_{u^n \in \{0,1,2\}^n} |W_2(u^n)| \cdot \Pr(\hat{U}^n = u^n) \quad (4.25)\]

\[= \mathbb{E}_{Q^n}X + \mathbb{E}_{\hat{U}^n}|W_2(\hat{U}^n)| \quad (4.26)\]
where

(a) follows from the fact that the uniform distribution maximizes the entropy of a discrete random variable.

Finally, we evaluate the second term in (4.26). By construction, element \( k \) is placed in \( W_2(\hat{U}^n) \) if \( \hat{U}^n(k) = 2 \), and since \( \hat{U}^n \) is an i.i.d. sequence, for every \( k \in [n] \) this happens with probability

\[
\Pr(\hat{U}^n(k) = 2) = \Pr(\hat{U}_k = 2 | p^n(k) = 0) \Pr(p^n(k) = 0) \tag{4.27}
\]

\[
= \left( 1 - \frac{1 - \bar{q}}{1 - \bar{p}} \right) (1 - \bar{p}), \tag{4.28}
\]

where (4.28) follows from (4.13). The random variable \(|W_2(\hat{U}^n)|\) represents the number of successes in \( n \) Bernoulli trials where the probability of success is given by Equation (4.28). We can therefore calculate that

\[
E_{\hat{U}^n}|W_2(\hat{U}^n)| = n \Pr(\hat{U}^n(k) = 2). \tag{4.29}
\]

By combining (4.29), (4.28) and (4.26), we can therefore derive part (a) of the Lemma. In order to also prove part (b), we simply note that the last term on the right-hand side of (4.24) is non-negative.

\[\square\]

**Lemma 4.** Let \( \tilde{Y}_i^n \) and \( \tilde{Y}_j^n \) be defined as in Definition 7 and \( \tilde{N}_i^n \) and \( \tilde{N}_j^n \) be defined as in Definition 6 for some \( i, j \in [M] \) such that \( i \neq j \). If \( D_1 = D^*(\epsilon_1) + \delta \), then

\[
I(\tilde{Y}_i^n; \tilde{Y}_j^n | \tilde{N}_i^n, \tilde{N}_j^n) \leq m(\delta + K(\epsilon_1, \epsilon_2)),
\]

where

\[
K(\epsilon_1, \epsilon_2) = \begin{cases} 
0 & \text{if } (1 - \epsilon_1) \geq 2(1 - \epsilon_2), \\
b(2(1 - \epsilon_2) - (1 - \epsilon_1)) & \text{if } (1 - \epsilon_1) < 2(1 - \epsilon_2). 
\end{cases} \tag{4.30}
\]

**Proof.**

\[
I(\tilde{Y}_i^n; \tilde{Y}_j^n | \tilde{N}_i^n, \tilde{N}_j^n) = H(\tilde{Y}_i^n | \tilde{N}_i^n, \tilde{N}_j^n) + H(\tilde{Y}_j^n | \tilde{N}_i^n, \tilde{N}_j^n) - H(\tilde{Y}_i^n, \tilde{Y}_j^n | \tilde{N}_i^n, \tilde{N}_j^n) \tag{4.32}
\]

\[
\leq (a) 2n(1 - \epsilon_2) - H(\tilde{Y}_i^n, \tilde{Y}_j^n | \tilde{N}_i^n, \tilde{N}_j^n) \tag{4.33}
\]

\[
= (b) 2n(1 - \epsilon_2) - \sum_{\tilde{\tilde{n}}_i^n, \tilde{\tilde{n}}_j^n} H(X_{\tilde{\tilde{n}}_i^n}(\tilde{\tilde{n}}_i^n)) \Pr(\tilde{\tilde{n}}_i^n = \tilde{\tilde{n}}_i^n, \tilde{\tilde{n}}_j^n = \tilde{\tilde{n}}_j^n) \tag{4.34}
\]

\[
= (c) 2n(1 - \epsilon_2) - \sum_{p^n \in \{0,1\}^n} H(X_{\tilde{\tilde{n}}_i^n}(p^n)) \Pr(P^n = p^n) \tag{4.35}
\]

\[
= (d) 2n(1 - \epsilon_2) - \sum_{p^n \in \{0,1\}^n} H(X_{\tilde{\tilde{n}}_i^n}(p^n)) \Pr(P^n = p^n) \tag{4.36}
\]

where

(a) follows from Lemma 1

(b) follows from Lemma 2, where \( C(\tilde{\tilde{n}}_i^n, \tilde{\tilde{n}}_j^n) = \{ k \in [n] \mid \tilde{\tilde{n}}_i^n(k) = 1 \text{ and } \tilde{\tilde{n}}_j^n(k) = 1 \} \), and \( \mathcal{V} = \{0,1\}^n \times \{0,1\}^n \).
(c) we have defined $g(p^n) = \sum_{(\tilde{n}_i^n, \tilde{n}_j^n) \in \mathcal{V}(p^n)} \Pr(\tilde{N}_i^n = \tilde{n}_i^n, \tilde{N}_j^n = \tilde{n}_j^n)$ and $\mathcal{V}(p^n) = \{(\tilde{n}_i^n, \tilde{n}_j^n) \in \mathcal{V} \mid I(p^n) = \mathcal{C}(\tilde{n}_i^n, \tilde{n}_j^n)\}$, i.e., $\mathcal{V}(p^n)$ is the set of $(\tilde{n}_i^n, \tilde{n}_j^n)$ such that the unerased indices of $\tilde{n}_i^n$ and $\tilde{n}_j^n$ correspond to $I(p^n)$

(d) we will justify this step in the next paragraph, and have defined $P^n = (P_1, P_2, \ldots, P_n)$ as a sequence of i.i.d. random variables where for all $i \in [n]$ $P_i$ is a Bernoulli random variable over the support $\{0, 1\}$ such that $\Pr(P_i = 1) = 1 - 2(1 - \epsilon_2)$.

We justify (4.36) by evaluating $g(p^n)$ in (4.35). For every $k \in I(p^n)$, by construction, we have that a necessary condition for $(\tilde{n}_i^n, \tilde{n}_j^n) \in \mathcal{V}(p^n)$ is that $\tilde{n}_i^n(k) = 0$ or $\tilde{n}_j^n(k) = 0$. In turn, this event happens if $U^n(k) = i$ or $U^n(k) = j$, which happens with probability $\Pr(U^n(k) \in \{i, j\}) = 2/M = 2(1 - \epsilon_2)$. Similarly, if $k \not\in I(p^n)$, we require that $U^n(k) \not\in \{i, j\}$, which happens with probability $\Pr(U^n(k) \not\in \{i, j\}) = 1 - 2(1 - \epsilon_2)$. Since $U^n$ is an i.i.d. sequence, we therefore have that for any $p^n \in \{0, 1\}^n$, $g(p^n)$ factorizes so that $g(p^n) = \Pr(P^n = p^n)$.

Now, consider user 1, the stronger user, and let $Y^n_1$, and $N^n_1$ be his channel output, and channel noise realization respectively. We write

$$m(1 - D_1) = mR(D_1) \quad (4.37)$$

$$\leq I(S^n; \hat{S}_1^n) \quad (4.38)$$

$$\leq I(X^n; Y^n_1) \quad (4.39)$$

$$= I(X^n; Y^n_1, N^n_1) \quad (4.40)$$

$$= I(X^n; Y^n_1 | N^n_1) \quad (4.41)$$

$$= H(Y^n_1 | N^n_1) \quad (4.42)$$

$$= \sum_{q^n \in \{0, 1\}^n} H(X^n_{I(q^n)}) \Pr(N_1^n = q^n) \quad (4.43)$$

where

(a) follows from the definition of the rate-distortion function

(b) follows from the data processing inequality, and the Markov Chain $S^n - X^n - Y^n_1 - \hat{S}_1^n$

(c) follows from the fact that the channel noise realization, $N^n_1$, is a function of $Y^n_1$

(d) follows from the fact that the channel noise realization, $N^n_1$ is independent of the channel input, $X^n$

(e) follows from the fact that $Y^n_1$ is a function of $X^n$ and $N^n_1$.

Finally, by substituting $D_1 = D^*(\epsilon_1) + \delta$ and applying Part (a) of Lemma 3 to the last terms on the right-hand sides of (4.43), and (4.36), we can therefore derive the upper bound implied by (4.30) when $(1 - \epsilon_1) \geq 2(1 - \epsilon_2)$. On the other hand, when $(1 - \epsilon_1) < 2(1 - \epsilon_2)$, we similarly derive the upper bound implied by (4.31) by invoking Part (b) of Lemma 3 on the right-hand sides of (4.43), and (4.36).
4.4 Virtual Source Reconstructions

Definition 12. Let \( \tilde{S}_i^m \triangleq \tilde{S}_2^m (\tilde{Y}_i^n) \) be the reconstruction for user 2 given channel output \( \tilde{Y}_i^n \), where \( i \in [M] \).

In our outer bound, we consider the class of non-erasure-randomized codes, which assumes that the positions of non-erased symbols in the reconstruction of \( \tilde{S}_2^m \) given any \( \tilde{Y}_i^n \), depends only on the channel noise realization \( \tilde{N}_i^n \), and not on the actual channel output \( \tilde{Y}_i^n \). We formally define this below.

Definition 13. Let \( Y_2^n \) be the channel output of user 2 and \( N_2^n \) be the associated channel noise sequence. A code is said to be non-erasure-randomized if for every \( \tilde{n}_0^n \in \{0,1\}^n \), there exists a predetermined set \( R(\tilde{n}_0^n) \subseteq [m] \) such that the reconstruction \( \tilde{S}_0^m = \tilde{S}_2^m (Y_2^n) \) satisfies

1. \( \tilde{S}_0^m (k) = \ast \) for all \( k \in [m] \setminus R(\tilde{n}_0^n) \)
2. \( \tilde{S}_0^m (j) = S^m (j) \) for all \( j \in R(\tilde{n}_0^n) \).

Definition 14. Let \( \mathcal{P}([m]) \) be the power set of \([m]\) and let \( R : \{0,1\}^n \to \mathcal{P}([m]) \) be defined as in Definition 13. For \( i \in [M] \), we define \( S_i \) as the set of random-variable source symbols not erased in \( \tilde{S}_i^m \), i.e., \( S_i = \{ S^m (k) \in S^m \mid k \in R(\tilde{N}_i^n) \} \).

For non-erasure-randomized codes, given the erasure pattern of the channel output, the position of erasures in the source reconstruction is determined by \( R(\cdot) \). We show in the following lemma that given \( R(\cdot) \) and the positions of the erasure symbols in the reconstruction \( \tilde{S}_i^m \), the only source of randomness in \( \tilde{S}_i^m \) comes from the value of the source symbols at the non-erased positions of \( \tilde{S}_i^m \).

Lemma 5. Let \( \tilde{S}_i^m \) and \( S_i \) be defined as in Definitions 12 and 14 respectively, and let \( U \) be an arbitrary collection of random variables. Then for non-erasure-randomized codes, \( H(\tilde{S}_i^m | \tilde{N}_i^n, U) = H(S_i | \tilde{N}_i^n, U) \)

Proof.

\[
H(\tilde{S}_i^m | \tilde{N}_i^n, U) = H(\tilde{S}_i^m | R(\tilde{N}_i^n), \tilde{N}_i^n, U) + I(R(\tilde{N}_i^n); \tilde{S}_i^m | \tilde{N}_i^n, U)
\]

\[
\overset{(a)}{=} H(\tilde{S}_i^m | R(\tilde{N}_i^n), \tilde{N}_i^n, U) + I(R(\tilde{N}_i^n); \tilde{S}_i^m | \tilde{N}_i^n, U)
\]

\[
\overset{(b)}{=} H(\bigcup_{k \in R(\tilde{N}_i^n)} \tilde{S}_i^m (k) | R(\tilde{N}_i^n), \tilde{N}_i^n, U)
\]

\[
\overset{(c)}{=} H(S_i | R(\tilde{N}_i^n), \tilde{N}_i^n, U)
\]

\[
\overset{(d)}{=} H(S_i | \tilde{N}_0^n, U)
\]

where

(a) follows from the assumption that \( R(\tilde{N}_i^n) \) is a deterministic function of \( \tilde{N}_i^n \)

(b) follows from Definition 13 whereby given \( \tilde{N}_i^n \) and \( R(\tilde{N}_i^n) \), for all \( k \in [m] \setminus R(\tilde{N}_i^n) \), \( \tilde{S}_i^m (k) = \ast \), i.e., \( \tilde{S}_i^m (k) \) is a constant

(c) follows from the fact that if a finite distortion is achieved, then \( \tilde{S}_i^m (k) = S^m (k) \) for all \( k \in R(\tilde{N}_i^n) \)

(d) follows from the assumption that \( R(\tilde{N}_i^n) \) is a deterministic function of \( \tilde{N}_i^n \).
Proof.
\[ I(\tilde{S}_i^m; \cup_{j \in B} \tilde{S}_j^m | \mathcal{N}) = H(\tilde{S}_i^m | \mathcal{N}) - H(\tilde{S}_i^m | \cup_{j \in B} \tilde{S}_j^m, \mathcal{N}) \]
\begin{align*}
(a) & \quad H(S_i | \mathcal{N}) - H(S_i | \cup_{j \in B} \tilde{S}_j^m, \mathcal{N}) \\
(b) & \quad H(S_i | \mathcal{N}) - H(S_i | \cup_{j \in \mathcal{A}} S^m(j), \mathcal{N}) \\
(c) & \quad H(S_i | \mathcal{N}) - H(S_i | \cup_{j \in \mathcal{A} \cap \mathcal{B}(\tilde{N}_i^m)} S^m(j), \cup_{k \in \mathcal{A} \cap R(\tilde{N}_i^m)} S^m(k), \mathcal{N}) \\
(d) & \quad + H(\cup_{j \in \mathcal{A} \cap \mathcal{B}(\tilde{N}_i^m)} S^m(j), \cup_{k \in \mathcal{A} \cap R(\tilde{N}_i^m)} S^m(k), \mathcal{N}) \\
(e) & \quad + H(S_i | \mathcal{N}) - H(S_i | \cup_{k \in \mathcal{A} \cap \mathcal{B}(\tilde{N}_i^m)} S^m(k), \mathcal{N}) \\
(f) & \quad H(\cup_{j \in \mathcal{A} \cap \mathcal{B}(\tilde{N}_i^m)} S^m(j), \mathcal{N}) \\
(g) & \quad \mathbb{E}_\mathcal{N}[A \cap R(\tilde{N}_i^m)] \\
(h) & \quad \mathbb{E}_\mathcal{N}[A \cap R(\tilde{N}_i^m)]
\end{align*}

where

(a) follows from Lemma 5

(b) follows from the definition of a non-erasure-randomized code, and \( \mathcal{A} \triangleq \cup_{k \in \mathcal{B}} R(\tilde{N}_k^m) \)

(c) we denote \( \bar{R}(\tilde{N}_i^m) = [m] \setminus R(\tilde{N}_i^m) \)

(d) follows from the chain rule for entropy

(e) follows from the fact that for any \( j \in \mathcal{A} \cap R(\tilde{N}_i^m), S^m(j) \in \mathcal{S}_i \)

(f) follows from the fact that the source is i.i.d. and for any \( k \in \mathcal{A} \cap \bar{R}(\tilde{N}_i^m), S^m(k) \not\in \mathcal{S}_i \)

(g) follows from the fact that the source is i.i.d. and the sets \( \mathcal{A} \cap R(\tilde{N}_i^m) \), and \( \mathcal{A} \cap \bar{R}(\tilde{N}_i^m) \) are disjoint

\[ \square \]
(h) follows from the fact that the source is binary, equiprobable, i.i.d. and independent of $\mathcal{N}$.

If merely an upper bound is needed for the mutual information term in Lemma 6, we can use the algebraic properties of sets to upper bound the right-hand-side of (4.59) in terms of the constituent elements of $\mathcal{A}$.

Lemma 7. Let $i \in [M], \mathcal{B} \subseteq [M]$ and $\mathcal{A} = \cup_{j \in \mathcal{B}} R(\tilde{\mathcal{N}}_n^m)$. Let $\mathcal{C} \subseteq [M]$ be any superset such that $\{i\} \cup \mathcal{B} \subseteq \mathcal{C}$ and let $\mathcal{N} = \{\tilde{\mathcal{N}}_k^n \mid k \in \mathcal{C}\}$ be a set of virtual channel noise sequences. Then $I(\tilde{\mathcal{S}}_m^i; \cup_{j \in \mathcal{B}} \tilde{\mathcal{S}}_m^j | \mathcal{N}) \leq \sum_{j \in \mathcal{B}} I(\tilde{\mathcal{S}}_m^i; \tilde{\mathcal{S}}_m^j | \mathcal{N})$.

Proof.

\begin{align}
I(\tilde{\mathcal{S}}_m^i; \cup_{j \in \mathcal{B}} \tilde{\mathcal{S}}_m^j | \mathcal{N}) & \overset{(a)}{=} E_{\mathcal{N}}[A \cap R(\tilde{\mathcal{N}}_n^m)] \\
& \overset{(b)}{=} E_{\mathcal{N}}[\cup_{j \in \mathcal{B}} (R(\tilde{\mathcal{N}}_j^n) \cap R(\tilde{\mathcal{N}}_i^n))] \\
& \overset{(c)}{\leq} \sum_{j \in \mathcal{B}} E_{\mathcal{N}}[R(\tilde{\mathcal{N}}_j^n) \cap R(\tilde{\mathcal{N}}_i^n)] \\
& \overset{(d)}{=} \sum_{j \in \mathcal{B}} I(\tilde{\mathcal{S}}_m^i; \tilde{\mathcal{S}}_m^j | \mathcal{N})
\end{align}

where

(a) follows from Lemma 6

(b) follows from the distributive property of set intersection and set union operations

(c) follows from the linearity of the expectation operator and the fact that $|A \cup B| \leq |A| + |B|$ for any sets $A$ and $B$

(d) follows from Lemma 6.

4.5 Main Result

Proposition 1 states that the marginal distributions of the virtual channel outputs are the same as the marginal distribution for the actual channel output of user 2. Since the capacity region depends only on the marginals of the channel output, we can therefore reconstruct the source within an expected distortion of $D_2$ given any $\tilde{\mathcal{Y}}_n^m$. We bound $D_2$, the distortion for user 2, by writing

\begin{align}
m(1 - D_2) &= mR(D_2) \\
& \leq I(S_m^m; \tilde{S}_m^i) \\
& \overset{(a)}{=} I(S_m^m; \tilde{S}_m^i, \mathcal{N}) - I(S_m^m; \mathcal{N} | \tilde{S}_m^i) \\
& \leq I(S_m^m; \tilde{S}_m^i, \mathcal{N}) \\
& \overset{(b)}{=} I(S_m^m; \tilde{S}_m^i | \mathcal{N}) \\
& \leq H(\tilde{S}_m^i | \mathcal{N})
\end{align}

where
(a) $\mathcal{N} = \{\tilde{N}_k \mid k \in [M]\}$ is the set of all virtual channel state sequences

(b) follows from the independence of the source sequence and the channel noise

Since (4.69) is valid for all $i \in [M]$, we sum these $M$ equations to get that

$$Mm(1 - D_2) = \sum_{i=1}^{M} H(\tilde{S}_i^m | \mathcal{N})$$

$$= H(\tilde{S}_1^m, \tilde{S}_2^m, \ldots, \tilde{S}_M^m | \mathcal{N}) + \sum_{i=1}^{M} I(\tilde{S}_i^m; \cup_{j=1}^{i-1} \tilde{S}_j^m | \mathcal{N})$$

$$= H(S_1, S_2, \ldots, S_M | \mathcal{N}) + \sum_{i=1}^{M} I(\tilde{S}_i^m; \cup_{j=1}^{i-1} \tilde{S}_j^m | \mathcal{N})$$

$$\leq m + \sum_{i=1}^{M} I(\tilde{S}_i^m; \cup_{j=1}^{i-1} \tilde{S}_j^m | \mathcal{N})$$

$$\leq m + \sum_{i=1}^{M-1} \sum_{j=1}^{i} I(\tilde{S}_i^m; \tilde{S}_j^m | \tilde{N}_i^m, \tilde{N}_j^m)$$

$$\leq m + \sum_{i=1}^{M} \sum_{j=1}^{i-1} I(\tilde{Y}_i^n, \tilde{Y}_j^n | \tilde{N}_i^n, \tilde{N}_j^n)$$

$$\leq m + \sum_{i=1}^{M} \sum_{j=1}^{i-1} m(\delta + K(\epsilon_1, \epsilon_2))$$

$$= m + m \frac{(M - 1)}{2} (\delta + K(\epsilon_1, \epsilon_2))$$

where

(a) follows from Lemma 20 in Appendix B

(b) follows from Lemma 5

(c) follows from the fact that there are at most $m$ symbols to reconstruct, the source is binary and i.i.d., and the uniform distribution maximizes the entropy of a discrete random variable

(d) follows from Lemma 7

(e) follows from the data processing inequality and the Markov chain $\tilde{S}_i^m - \tilde{Y}_i^n - \tilde{Y}_j^n - \tilde{S}_j^m$

(f) follows from Lemma 4

By solving for $D_2$ in (4.77) and recognizing the fact that $\epsilon_2$ can be expressed as $\epsilon_2 = 1 - 1/M$, we get that

$$D_2 \geq \epsilon_2 - \frac{(M - 1)}{2} (\delta + K(\epsilon_1, \epsilon_2)).$$

**Theorem 2.** Let $\epsilon_2 = 1 - 1/M$ for some integer $M > 1$, and let $D_1 = D^*(\epsilon_1) + \delta$ for some $\delta > 0$. Then for all non-erasure-randomized codes, $D_2$ satisfies (4.78), where $K(\epsilon_1, \epsilon_2)$ is given in Lemma 4.
We mention the special case in which 

\((1 - \epsilon_1) \geq 2(1 - \epsilon_2)\) and 

\(D_1 = D^*(\epsilon_1) + \delta_m\) where \(\delta_m \to 0\) as 

\(m \to \infty\). In this case, \(K(\epsilon_1, \epsilon_2) = 0\) and we have that 

\(D_2 \geq \epsilon_2 - \delta'_m\), where \(\delta'_m \to 0\) as \(m \to \infty\). In 

fact, this lower bound can be achieved with a systematic code that meets user 1’s point-to-point-optimal 

distortion constraint, which leads to the following corollary.

**Corollary 1.** Let \(\epsilon_2 = 1 - 1/M\) for some integer \(M > 1\), and let 

\((1 - \epsilon_1) \geq 2(1 - \epsilon_2)\). Then if user 1 

meets their point-to-point optimal distortion, 

\(D_1 = D^*(\epsilon_1)\), the distortion for user 2 given by 

\(D_2 = \epsilon_2\) is 

optimal for all non-erasure-randomized codes.

### 4.6 Numerical Plot

In Figure 4.2, we give a numerical plot to illustrate the result in Theorem 2. In this plot, we fix \(\epsilon_1 = 0.85\), 

\(b = 4.67\), and set the stronger user’s distortion to be 

point-to-point optimal, i.e., \(D_1 = D^*(\epsilon_1)\). We then 

vary \(\epsilon_2\) on the x-axis and plot, in circles, the outer bound for the distortion of the weaker user from 

Theorem 2. We compare this to the Shannon outer bound plotted with the dash-dot line, and we see 

that for non-erasure-randomized codes, Theorem 2 gives a greater lower-bound in some regions. The 

solid vertical line indicates the boundary of \(\epsilon_2\) for which the region to the left has values of \(\epsilon_2\) that satisfy 

\((1 - \epsilon_1) < 2(1 - \epsilon_2)\), while the region to the right has values of \(\epsilon_2\) that satisfy 

\((1 - \epsilon_1) \geq 2(1 - \epsilon_2)\) We see that in both regions, Theorem 2 can still give a better outer bound than the Shannon bound.

Finally, we also plot the successive segmentation inner bound as a dashed line. As mentioned in 

Corollary 1, we see that the outer and inner bounds coincide when \((1 - \epsilon_1) \geq 2(1 - \epsilon_2)\).
Chapter 5

Two-terminal Erasure Source-Broadcast with Feedback

We study the effects of introducing a feedback channel in the erasure source-broadcast problem for the case of two receivers. The receivers each require a certain fraction of a source sequence, and we are interested in the minimum latency, or transmission time, required to serve them all. We first show that for a two-user broadcast channel, a point-to-point outer bound can always be achieved. In Section 5.4, we also show that optimal performance can be achieved if only one of the users, the stronger user, has a feedback channel.

5.1 Introduction

The presence of a feedback channel can have many benefits. In addition to practical issues such as reducing complexity, it can also increase fundamental communication rates in multiuser networks (see, e.g., [62, 63]). In particular, feedback has been shown to increase the channel capacity of the erasure broadcast channel [40, 41]. In contrast, we study a joint source-channel coding problem of broadcasting an equiprobable binary source over an erasure broadcast channel with feedback. Each receiver demands a certain fraction of the source, and we look to minimize the overall transmission time needed to satisfy all user demands.

We are motivated by heterogeneous broadcast networks, where we encounter users with very different channel qualities, processing abilities, mobility, screen resolutions, etc. In such networks, the user diversity can translate to different distortion requirements from the broadcaster since, e.g., a high-quality reconstruction of a video may not be needed by a user with a limited-capability device. In addition, if the source we wish to reconstruct is the output of a multiple description code, then the fraction of the source that is recoverable is of interest.

A related problem has been studied for the case without feedback in [34]. There are also many other variations of the problem (see [64] for a thorough literature review). A channel coding version of the problem was studied in [40], which proposed a general algorithm for sending a fixed group of messages to \( n \) users over an erasure broadcast channel with feedback. In contrast, our formulation allows flexibility in which messages are received at a user so long as the total number received exceeds a certain threshold. The variant of the index coding problem of [65] is similar in this respect in that given \( n \) users, each
already possessing a different subset of \( m \) messages, the goal is to minimize the number of transmissions before each user receives any additional \( t \) messages.

In our work, we utilize uncoded transmissions that are instantly-decodable, and distortion-innovative. The zero latency in decoding uncoded packets has benefits in areas in which packets are instantly useful at their destination such as applications in video streaming and disseminating commands to sensors and robots \([36, 66]\). We show that when feedback is available from both users, we can always send instantly-decodable, distortion-innovative transmissions. While this may not necessarily be the case if a feedback channel is available from only the stronger user, we will show that in this case, the optimal minmax latency can still be achieved by using repetition coding in tandem with the transmission of random linear combinations.

### 5.2 System Model

The general problem we consider involves communicating a binary memoryless source \( \{S(t)\}_{t=1,2,\ldots} \) to \( n \) users over an erasure broadcast channel with feedback. The source produces equiprobable symbols in \( \mathcal{S} = \{0, 1\} \) and is communicated by an encoding function that produces the channel input sequence \( X^W = (X(1), \ldots, X(W)) \), where \( X(t) \) denotes the \( t \)th channel input taken from the alphabet \( \mathcal{X} = \{0, 1\} \).

We assume that \( X(t) \) is a function of the source as well as the channel outputs of all users prior to time \( t \).

Let \( Y_i(t) \) be the channel output observed by user \( i \) on the \( t \)th channel use for \( i \in [n] \triangleq \{1, 2, \ldots, n\} \).

We let \( Y_i(t) \) take on values in the alphabet \( \mathcal{Y} = \{0, 1, \ast\} \) so that an erasure event is represented by \( \ast \).

We associate user \( i \) with the state sequence \( \{Z_i(t)\}_{t \in [W]} \), which represents the noise on user \( i \)'s channel, where \( Z_i(t) \in \mathcal{Z} \triangleq \{0, 1\} \), and \( Y_i(t) \) will be erased if \( Z_i(t) = 1 \) and \( Y_i(t) = X(t) \) if \( Z_i(t) = 0 \). The channel we consider is memoryless in the sense that \( Z_i(t) \) is drawn i.i.d. from a Bern(\( \epsilon_i \)) distribution, where \( \epsilon_i \) denotes the erasure rate of the channel corresponding to user \( i \).

The problem we consider involves causal feedback that is universally available. That is, at time \( T \), we assume that \( \{Z_1(t), Z_2(t), \ldots, Z_n(t)\}_{t=1,2,\ldots,T-1} \) is available to the transmitter and all receivers. After \( W \) channel uses, user \( i \) utilizes the feedback and his own channel output to reconstruct the source as a length-\( N \) sequence, denoted as \( \hat{S}_i^N \). We will be interested in a fractional recovery requirement so that each symbol in \( \hat{S}_i^N \) either faithfully recovers the corresponding symbol in \( S^N \), or otherwise a failure is indicated with an erasure symbol, i.e., we do not allow for any bit flips.

More precisely, we choose the reconstruction alphabet \( \hat{\mathcal{S}} \) to be an augmented version of the source alphabet so that \( \hat{\mathcal{S}} = \{0, 1, \ast\} \), where the additional \( \ast \) symbol indicates an erasure symbol. Let \( \mathcal{D} = \{0, 1\} \) and \( d_i \in \mathcal{D} \) be the distortion user \( i \) requires. We then express the constraint that an achievable code ensures that each user \( i \in [n] \) achieves a fractional recovery of \( 1 - d_i \) with the following definition.

**Definition 15.** An \((N,W,d_1,d_2,\ldots,d_n)\) code for source \( S \) on the erasure broadcast channel with feedback consists of

1. a sequence of encoding functions \( f_{i,N} : S^N \times \prod_{j=1}^n Z_j^{i-1} \rightarrow \mathcal{X} \) for \( i \in [W] \), such that \( X(i) = f_{i,N}(S^N, Z_1^{i-1}, Z_2^{i-1}, \ldots, Z_n^{i-1}) \), and

2. \( n \) decoding functions \( g_{i,N} : \mathcal{Y}^W \times Z_1^W \rightarrow \hat{\mathcal{S}}^N \) s.t. \( \hat{S}_i^N = g_{i,N}(Y_i^W, Z_1^W, Z_2^W, \ldots, Z_n^W), i \in [n] \),

   (a) \( \hat{S}_i^N \) is such that for \( t \in [N] \), if \( \hat{S}_i(t) \neq S(t) \), then \( \hat{S}_i(t) = \ast \),
(b) \[ \mathbb{E}\left| \{ t \in \mathbb{N} \mid \hat{S}_i(t) = \ast \} \right| \leq Nd_i, \]

We mention that in our problem formulation, we assume that all receivers have causal knowledge of \((Z_1^{t-1}, Z_2^{t-1}, \ldots, Z_n^{t-1})\) at time \(t\). That is, each receiver has causal knowledge of which packets were received. This can be made possible, for example, through the control plane of a network.

We define the **latency** that a given code requires before all users can recover their desired fraction of the source as follows:

**Definition 16.** The latency, \(w\), of an \((N, W, d_1, d_2, \ldots, d_n)\) code is the number of channel uses per source symbol that the code requires to meet all distortion demands, i.e., \(w = W/N\).

Our goal is to characterize the achievable latencies under a prescribed distortion vector, as per the following definition:

**Definition 17.** Latency \(w\) is said to be \((d_1, d_2, \ldots, d_n)\)-achievable over the erasure broadcast channel if for every \(\delta > 0\), there exists for sufficiently large \(N\), an \((N, wN, \hat{d}_1, \hat{d}_2, \ldots, \hat{d}_n)\) code such that for all \(i \in [n]\), \(d_i + \delta \geq \hat{d}_i\).

In this chapter, we consider the case when \(n = 2\). We study the case when \(n = 3\) in the following chapter. We will also make a mention whenever the strategies we use for these cases can be generalized for \(n\) users.

### 5.3 Coding for Two Users

We first show that the case involving only two users can be fully solved. We do this by demonstrating an algorithm that achieves point-to-point optimality for both users at any time during transmission. In the first phase of the algorithm, we successively transmit each source symbol uncoded until at least one user receives it. If \(S(t)\) is received only by user \(i\), then the transmitter places \(S(t)\) into queue \(Q_j\). No action is taken if both users receive \(S(t)\). By assumption, feedback is universally available, and so user \(i\) is also able to maintain a local version of queue \(Q_j\).

Now, after this first phase, the transmitter has built queues \(Q_1\) and \(Q_2\), where for \(i, j \in \{1, 2\}, i \neq j\), user \(i\) has knowledge of packets in \(Q_j\) and is in need of those in \(Q_i\). Thus, the algorithm’s next phase involves successively transmitting a linear combination of the packets at the fronts of \(Q_1\) and \(Q_2\). Let \(q_i\) be the packet at the front of \(Q_i\). Notice that a successfully received channel symbol of the form \(q_1 \oplus q_2\) means that user 1 is able to decode \(q_1 \in Q_1\), since he has access to \(q_2 \in Q_2\). We therefore remove \(q_i\) from \(Q_i\) whenever a linear combination involving \(q_i\) is received by user \(i\). This entire phase continues until the users’ distortion constraints are met. The decoding algorithm for this scheme is also simple. Given that user \(i\) has received source symbol \(S(t)\), user \(i\) sets \(\hat{S}_i(t) = S(t)\).

Our algorithm has two appealing properties. The first is that it involves only transmissions that are *[instantly decodable]*, which is seldom the case when channel codes are used. Secondly, this coding scheme involves only transmissions that are *[distortion-innovative]*. This means that any successfully received channel symbol can be immediately used to reconstruct a single source symbol that was hitherto unknown. In fact, our coding scheme has the property that for any latency \(w \in [0, 1/(1 - \epsilon)]\), after \(wN\) transmissions have been sent over the channel, an expected value of \(\gamma = wN(1 - \epsilon)\) channel symbols were received, which leads to the decoding of precisely \(\gamma\) source symbols. The distortion achieved after \(wN\) transmissions is thus seen to be \(D = 1 - w(1 - \epsilon)\), which we readily recognize as the separation-based
outer bound. Since the transmission of an instantly-decodable, distortion-innovative symbol does not require a channel encoder, we will sometimes refer to such transmissions as analog transmissions.

In the next Section we study a variation of our problem formulation in which only the stronger user has a feedback channel available. For this problem variation, we show that we can still achieve the optimal minmax latency despite the weaker user not having access to a feedback channel.

5.4 Source-broadcast with One-sided Feedback

In this section we consider a one-sided-feedback variation of the problem in Section 5.3 whereby in a broadcast network with two receivers, a feedback channel is available to only the stronger user. In this scenario, we show that given the distortion constraints of both users, there is no overhead in the minmax latency achieved. Specifically, for $i \in \{1, 2\}$, let $w^*_i(d_i)$ be the point-to-point optimal latency for user $i$ and let $w^+(d_1, d_2)$ be the Shannon lower bound for the minmax latency where

$$w^*_i(d_i) = \frac{1 - d_i}{1 - \epsilon_i}, \quad (5.1)$$

and

$$w^+(d_1, d_2) = \max_{i \in \{1, 2\}} \frac{1 - d_i}{1 - \epsilon_i}. \quad (5.2)$$

Section 5.3 showed that for $i \in \{1, 2\}$, user $i$ can achieve distortion $d_i$ at the optimal latency $w^*_i(d_i)$ when a feedback channel is available to both users (c.f. Section 3.5 on individual decoding delays). Clearly, the optimal minmax system latency $w^+(d_1, d_2)$ is also achievable in this case. In contrast, in this section we show that when a feedback channel is available to only the stronger user, while the individual optimal latencies may or may not be achievable, the overall system latency $w^+(d_1, d_2)$ is still achievable.

5.4.1 Problem Formulation

The problem is illustrated in Figure 5.1. We consider the problem of communicating a binary memoryless source $\{S(t)\}_{t=1, 2, \ldots}$ to two users over an erasure broadcast channel with one-sided feedback. Of the two receivers in the broadcast network, only the receiver with the lower erasure rate has a feedback channel available. The source produces equiprobable symbols in $\mathcal{S} = \{0, 1\}$ and is communicated by an encoding function that produces the channel input sequence $X^W = (X(1), \ldots, X(W))$, where $X(t)$ denotes the
Chapter 5. Two-terminal Erasure Source-Broadcast with Feedback

47

of the source as well as the channel outputs of the stronger user prior to time $t$.

Let $Y_i(t)$ be the channel output observed by user $i$ on the $i^{th}$ channel use for $i \in \{1, 2\}$. We let $Y_i(t)$ take on values in the alphabet $\mathcal{Y} = \{0, 1, \star\}$ so that an erasure event is represented by ‘$\star$’. We associate user $i$ with the state sequence $\{Z_i(t)\}_{t=1,2,...,T-1}$, which represents the noise on user $i$’s channel, where $Z_i(t) \in \mathcal{Z} = \{0, 1\}$, and $Y_i(t)$ will be erased if $Z_i(t) = 1$ and $Y_i(t) = X(t)$ if $Z_i(t) = 0$. The channel we consider is memoryless in the sense that $(Z_1(t), Z_2(t))$ is drawn i.i.d. from the probability mass function given by

\[
\Pr(Z_1 = 1, Z_2 = 1) = \epsilon_{12} \\
\Pr(Z_1 = 1, Z_2 = 0) = \epsilon_1 - \epsilon_{12} \\
\Pr(Z_1 = 0, Z_2 = 1) = \epsilon_2 - \epsilon_{12} \\
\Pr(Z_1 = 0, Z_2 = 0) = 1 - \epsilon_1 - \epsilon_2 + \epsilon_{12},
\]

where $\epsilon_{12} \in (0, 1)$ is the probability that an erasure simultaneously occurs on both channels and $\epsilon_i \in (0, 1)$ denotes the erasure rate of the channel corresponding to user $i$, where we assume $\epsilon_1 < \epsilon_2$.

The problem we consider involves causal feedback from the stronger user that is universally available. That is, at time $T$, we assume that $\{Z_i(t)\}_{t=1,2,...,T-1}$ is available to the transmitter and both receivers. After $W$ channel uses, user $i$ utilizes the feedback and his own channel output to reconstruct the source as a length-$N$ sequence, denoted as $\hat{S}^N_i$. We will be interested in a fractional recovery requirement so that each symbol in $\hat{S}^N_i$ either faithfully recovers the corresponding symbol in $S^N_i$, or otherwise a failure is indicated with an erasure symbol, i.e., we do not allow for any bit flips.

More precisely, we choose the reconstruction alphabet $\hat{S}$ to be an augmented version of the source alphabet so that $\hat{S} = \{0, 1, \star\}$, where the additional ‘$\star$’ symbol indicates an erasure symbol. Let $\mathcal{D} = [0,1]$ and $d_i \in \mathcal{D}$ be the distortion user $i$ requires. We then express the constraint that an achievable code ensures that each user $i \in \{1, 2\}$ achieves a fractional recovery of $1 - d_i$ with the following definition.

**Definition 18.** An $(N,W,d_1,d_2)$ code for source $S$ on the erasure broadcast channel with feedback consists of

1. A sequence of encoding functions $f_{i,N} : S^N \times \mathcal{Z}^{i-1} \rightarrow \mathcal{X}$ for $i \in [W]$, s.t. $X(i) = f_{i,N}(S^N, Z_{i-1}^i)$
2. Two decoding functions $g_{i,N} : \mathcal{Y}^W \times Z^W \rightarrow \hat{S}^N_i$ s.t. $\hat{S}^N_i = g_{i,N}(Y_i^W, Z_i^W)$, $i \in \{1,2\}$, and
   (a) $\hat{S}^N_i$ is such that for $t \in [N]$, if $\hat{S}_i(t) \neq S(t)$, then $\hat{S}_i(t) = \star$,
   (b) $\mathbb{E} \left| \{ t \in [N] \mid \hat{S}_i(t) = \star \} \right| \leq N d_i$.

We next define the latency that a given code requires before all users can recover their desired fraction of the source.

**Definition 19.** The latency, $w$, of an $(N,W,d_1,d_2)$ code is the number of channel uses per source symbol that the code requires to meet all distortion demands, i.e., $w = W/N$.

Our goal is to characterize the achievable latencies under a prescribed distortion vector, as per the following definition.

**Definition 20.** Latency $w$ is said to be $(d_1,d_2)$-achieved over the erasure broadcast channel with one-sided feedback if for every $\delta > 0$, there exists for sufficiently large $N$, an $(N,wN,d_1,d_2)$ code such that for all $i \in \{1,2\}$, $d_i + \delta \geq \hat{d}_i$. 
5.4.2 Coding Scheme

We begin by reviewing a repetition coding scheme in the next subsection, whereby we simply ignore the weaker user, and focus on using the stronger user’s feedback to retransmit each of his required source symbols until it is received. A repetition coding scheme is useful insofar as it helps avoid compelling the weaker user to decode additional source symbols that he does not require.

As an example, consider when the stronger user requires $N$ source symbols. We could send random linear combinations of the $N$ symbols, which he could decode after receiving $N$ equations through, say, $W$ transmissions. By the time the stronger user has recovered the $N$ equations, the weaker user, having a weaker channel, would have received less than $N$ equations. At this point, the weaker user could simply ignore the first $W$ transmissions, and have the transmitter encode another random linear combination of symbols for the weaker user to decode. However, such a timesharing scheme is inefficient. On the other hand, the weaker user could prevent the first $W$ transmissions from going to waste by continuing to receive random linear combinations of the group of $N$ source symbols originally intended for the stronger user. However, if the weaker user requires $M < N$ symbols, he would have had to listen to many more transmissions than necessary to recover $M$ symbols thus introducing delay.

We notice the problem is in the random linear combinations used in our coding scheme. In such a scheme, we either receive more than $N$ equations and decode the entirety of the $N$ source symbols, or we receive less than $N$ equations and decode none of the source symbols. This “threshold effect” is detrimental when we have heterogeneous users in a network who require $M < N$ source symbols.

The repetition coding scheme avoids this pitfall by avoiding random linear combinations altogether and instead transmitting uncoded source symbols over the channel. While this avoids compelling the weaker user to decode unnecessary source symbols, it can also be inefficient for the weaker user. Specifically, since the repetition scheme is based solely on the stronger user’s feedback, a source symbol can be retransmitted even after it is received by the weaker user. We show how to circumvent this problem by creating a hybrid coding scheme that consists of both repetitions, and random linear combinations. The coding scheme is controlled by two variables, $\theta$, and $\gamma$. We show how to choose specific values for these parameters to achieve the optimal minmax latency in Section 5.4.3.

Repetition Coding

Consider a coding scheme that simply ignores the weaker user and focuses on using the stronger user’s feedback to retransmit each of his required source symbols until it is received. We wish to calculate the expected value of $N_0$, which we define as the number of unique source symbols received by the weaker user when $k \in \mathbb{N}$ symbols are to be sent to the stronger user via repetition coding.

Let $M_i$ be a random variable representing the number of transmissions needed to be sent for symbol $S(i)$ to be received by the stronger user, user 1, in the repetition scheme. Let $I_i$ be an indicator variable indicating whether symbol $S(i)$ was received by the weaker user, user 2, in any of the $M_i$ transmissions and let $\mathbf{M} = \{M_1, M_2, \ldots, M_k\}$ be the vector of random variables giving the number of repetitions needed to send each source symbol. Given $\mathbf{M}$, we calculate $\mathbb{E}(N_0|\mathbf{M})$ as
Chapter 5. Two-terminal Erasure Source-Broadcast with Feedback

\[ \mathbb{E}(N_0|\mathcal{M}) = \sum_{i=1}^{k} \mathbb{E}(I_i|\mathcal{M}) \]

\[ = \sum_{i=1}^{k} \Pr(\text{user 2 receives } S(i)|\mathcal{M}) \]

\[ \overset{(a)}{=} \sum_{i=1}^{k} \left\{ 1 - \Pr(Z_2 = 1|Z_1 = 1)^{M_i-1} \Pr(Z_2 = 1|Z_1 = 0) \right\} \]

\[ \overset{(b)}{=} \sum_{i=1}^{k} \left\{ 1 - \left( \frac{\epsilon_{12}}{\epsilon_1} \right)^{M_i-1} \left( \frac{\epsilon_2 - \epsilon_{12}}{1 - \epsilon_1} \right) \right\} \]

where

(a) follows by construction of the repetition-based scheme

(b) we have calculated the conditional probabilities from (5.3).

We next use the law of total probability to get that

\[ \mathbb{E}N_0 = \sum_{\mathcal{M}} \mathbb{E}(N_0|\mathcal{M}) \Pr(\mathcal{M}) \]

\[ \overset{(a)}{=} \sum_{\mathcal{M}} \sum_{i=1}^{k} \left\{ 1 - \left( \frac{\epsilon_{12}}{\epsilon_1} \right)^{M_i-1} \left( \frac{\epsilon_2 - \epsilon_{12}}{1 - \epsilon_1} \right) \right\} \Pr(\mathcal{M}) \]

\[ \overset{(b)}{=} \sum_{i=1}^{k} \sum_{j=1}^{\infty} \left\{ 1 - \left( \frac{\epsilon_{12}}{\epsilon_1} \right)^{M_i-1} \left( \frac{\epsilon_2 - \epsilon_{12}}{1 - \epsilon_1} \right) \right\} \Pr(M_i = j) \]

\[ = k - \left( \frac{\epsilon_2 - \epsilon_{12}}{1 - \epsilon_1} \right) \sum_{i=1}^{k} \sum_{j=1}^{\infty} \left( \frac{\epsilon_{12}}{\epsilon_1} \right)^{j-1} \left( \epsilon_1^{j-1} (1 - \epsilon_1) \right) \]

\[ \overset{(c)}{=} k - \left( \frac{\epsilon_2 - \epsilon_{12}}{1 - \epsilon_1} \right) \sum_{i=1}^{k} \sum_{j=1}^{\infty} \left( \frac{\epsilon_{12}}{\epsilon_1} \right)^{j-1} \left( \epsilon_1^{j-1} (1 - \epsilon_1) \right) \]

\[ = k \left( 1 - (\epsilon_2 - \epsilon_{12}) \sum_{j=1}^{\infty} (\epsilon_{12})^{j-1} \right) \]

\[ \overset{(d)}{=} k \left( 1 - (\epsilon_2 - \epsilon_{12}) \left\{ \frac{1}{1 - \epsilon_{12}} \right\} \right) \]

\[ = k \cdot \frac{1 - \epsilon_2}{1 - \epsilon_{12}} \]

where

(a) follows from (5.7)

(b) follows from the fact that the \( M_i \) are i.i.d.

(c) follows from the fact that by construction, \( M_i \) is a geometric random variable with probability of success \( (1 - \epsilon_1) \)

(d) follows from the formula for the geometric series and the fact that \( \epsilon_{12} < 1. \)

**Lemma 8.** Let \( k \) source symbols be sent to the stronger user via a repetition scheme. Then \( \mathbb{E}N_0 \), the expected number of unique source symbols received by the weaker user, is given by (5.15).
Inner Bound

In this section, we formulate a hybrid coding scheme that incorporates both repetition coding and sending random linear combinations of source symbols. The code is tuned via two parameters, \( \theta, \gamma \in [0,1] \). We target point-to-point optimal performance for the stronger user in this section, and show that this is possible for any values of \( \theta \) and \( \gamma \). In the next section however, we show how to optimize \( \theta \) and \( \gamma \) to achieve the optimal minmax latency. As in the coding scheme in Section 5.3, we again split the coding scheme into phases.

In Phase I, we begin by sending each source symbol uncoded over the channel. That is, Phase I consists of \( N \) transmissions and at time \( t \in \{1,2,\ldots,N\} \), we transmit \( X(t) = S(t) \). Let \( A \subseteq \{S(1),S(2),\ldots,S(N)\} \) be the set of symbols received by the stronger user in Phase I. Since the stronger user’s feedback is available to all receivers and the transmitter, \( A \) is known to all parties.

At the conclusion of Phase I, we have that on average, for \( i \in \{1,2\} \), user \( i \) will have received \( N(1-\epsilon_i) \) source symbols and so will require an additional \( N(\epsilon_i-d_i) \) symbols in the remaining phases. Before moving on to Phase II, we first organize the source symbols in \( A \) and \( A^c \) into subsets, where \( A^c \subseteq \{S(1),S(2),\ldots,S(N)\} \) denotes the complement of set \( A \). We first isolate a fraction of \( N(\epsilon_1-d_1) \) source symbols from \( A^c \) into a set denoted as \( B \). That is, we fix the remaining \( N(\epsilon_1-d_1) \) symbols that the stronger user requires in \( B \). We then partition \( B \) into two disjoint sets, one that contains a fraction of \( \theta \in [0,1] \) source symbols from \( B \), denoted as \( B_\theta \), and the other that contains a fraction of \( 1-\theta \) symbols, denoted as \( B_\overline{\theta} \), where \( B = B_\theta \cup B_\overline{\theta} \). Random linear combinations of the symbols in \( B_\theta \) will be sent to the stronger user while the symbols in \( B_\overline{\theta} \) will be sent with repetition coding. We further take a fraction of \( \gamma \in [0,1] \) source symbols from \( A \) and denote this set as \( C \). Finally, we define \( F \) as the union of sets \( C \) and \( B_\theta \). Figure 5.2 illustrates the relationship between all sets and the manner in which they are constructed.

In Phase II of the coding scheme, we designate the symbols of \( B_\theta \) as the symbols to be transmitted to the stronger user with a repetition scheme. However, we modify the repetition scheme to incorporate random linear combinations of symbols in \( F \). In a conventional repetition scheme, we would retransmit
b ∈ B_2 until it is received by the stronger user. Upon reception by the stronger user, we move on to the next symbol b' ∈ B_2 and continue in this manner until all symbols in B_2 are accounted for. Let b_2(t) ∈ B_2 be the source symbol being repeated at time t. Our modified coding scheme is similar to the conventional repetition scheme except that at any time t, instead of only transmitting b_2(t), we instead send v(t) + b_2(t), where v(t) is a new random linear combination of the source symbols in F generated for every time t. Let b ∈ B_2. If b_2(t) = b is transmitted and subsequently received by the stronger user at time t, the protocol for replacing b at time t + 1 with another source symbol from B_2 is identical to the conventional repetition scheme, however the only difference is that we now combine b_2(t) with a random linear combination of the symbols of F at every transmission. Phase II concludes when all symbols in B_2 have been accounted for by the modified repetition scheme.

At the conclusion of Phase II, since we have transmitted the symbols in B_2 as if we were utilizing a repetition scheme, we have that the stronger user will have received |B_2| equations involving |B_2| + |F| variables. Notice, however, that since F = C ∪ B_2, and C ⊆ A, the stronger user can subtract off all symbols originating from C. Therefore, Phase I actually results in the stronger user receiving |B_2| equations involving |B_2| + |B_2| = |B| unknown variables, where E|B| = N(ε_1 − d_1). The stronger user therefore requires an additional |B_2| equations at the conclusion of Phase II.

In Phase III, we send the remaining equations to the stronger user by continuing to send v(t) at any time t. That is, we continue sending random linear combinations of the symbols in F. Phase III concludes when the feedback of the stronger user indicates that he has received the missing |B_2| equations.

At the conclusion of Phase III, it is not hard to see that the stronger user achieves point-to-point optimal performance, since every channel symbol received has provided an independent equation that can be used to decode a new source symbol. At this point, if w_2^*(d_2) ≤ w_1^*(d_1), we halt any further transmissions, where w_1^*(d_i) is given by (5.1).

In Phase IV, if w_2^*(d_2) > w_1^*(d_1), we continue to transmit v(t), the random linear combinations of the source symbols in F, for an additional N(w_2^*(d_2) − w_1^*(d_1)) transmissions.

### 5.4.3 Minmax Latency Optimality

In this section, we show that it is possible to choose values for θ, γ ∈ [0, 1] from Section 5.4.2 so that the lower bound for the minmax latency in (5.2) is achieved. We first calculate the expected number of unknown variables involved in transmissions to the weaker user from Phase II onwards.

First, since we send random linear combinations of the symbols in F in Phase II, we initially expect this to contribute |F| variables. However, some of the symbols in F have already been received by user 2 in Phase I. Let N_1 be the number of symbols in F not received by user 2 in Phase I. Given a channel noise realization (Z_1^W, Z_2^W) = (z_1^W, z_2^W), we can calculate the expected value of N_1 as
\[ E(N_1(Z_1^W, Z_2^W) = (z_1^W, z_2^W)) = \sum_{s \in F} \Pr(s \text{ not received by user 2 in Phase I}) \]
\[ \equiv \sum_{s \in C} \Pr(s \text{ not received by user 2 in Phase I}) \]
\[ + \sum_{s' \in B_\theta} \Pr(s' \text{ not received by user 2 in Phase I}) \]
\[ \equiv \sum_{s \in C} \Pr(Z_2 = 1|Z_1 = 0) + \sum_{s' \in B_\theta} \Pr(Z_2 = 1|Z_1 = 1) \]
\[ \equiv \sum_{s \in C} \left( \frac{\epsilon_2 - \epsilon_{12}}{1 - \epsilon_1} \right) + \sum_{s' \in B_\theta} \left( \frac{\epsilon_{12}}{\epsilon_1} \right) \]
\[ = |C| \left( \frac{\epsilon_2 - \epsilon_{12}}{1 - \epsilon_1} \right) + |B_\theta| \left( \frac{\epsilon_{12}}{\epsilon_1} \right) \]

where

(a) follows from the fact that \( F = C \cup B_\theta \) and \( C \) and \( B_\theta \) are disjoint by construction

(b) follows from the fact that by construction, all symbols in \( C \) have been received by user 1 and all symbols in \( B_\theta \) were not received by user 1

(c) we have calculated the conditional probabilities from (5.3).

The cardinality of sets \( C \) and \( B_\theta \) depends on the channel noise variables \((Z_1^W, Z_2^W)\). By taking the expectation over the channel noise, we can calculate the unconditional expected value of \( N_1 \) as

\[ E N_1 \overset{(a)}{=} N \gamma (1 - \epsilon_1) \left( \frac{\epsilon_2 - \epsilon_{12}}{1 - \epsilon_1} \right) + N (1 - \theta) (\epsilon_1 - d_1) \left( \frac{\epsilon_{12}}{\epsilon_1} \right) \]

where

(a) follows from (5.20) and by construction of the sets (see Section 5.4.2 and Figure 5.2).

The use of repetition coding for the symbols in \( B_\theta \) in Phase II further adds additional unknown variables to the coding scheme. On average, the expected number of symbols repeated is \( E|B_\theta| = N(1 - \theta)(\epsilon_1 - d_1) \), of which, again, only a fraction of \( \Pr(Z_2 = 1|Z_1 = 1) = \epsilon_{12}/\epsilon_1 \) were not already received by the weaker user in Phase I. By Lemma 8, the number of additional unknown variables introduced to the weaker user as a result of the repetition scheme is therefore given by \( N_2 \), where

\[ E N_2 = N(1 - \theta)(\epsilon_1 - d_1) \left( \frac{\epsilon_{12}}{\epsilon_1} \right) \left( \frac{1 - \epsilon_2}{1 - \epsilon_{12}} \right) \]

Let \( \mathcal{L}(\gamma, \theta) \) be the expected fraction of all source symbols that are involved in transmissions to the weaker user from Phase II onwards that have not yet been decoded prior to Phase II. We have that \( \mathcal{L}(\gamma, \theta) \) is the normalized sum of (5.21) and (5.22), i.e.,

\[ \mathcal{L}(\gamma, \theta) = \frac{E N_1 + E N_2}{N} \]
\[ = k_1 \gamma + k_2 \theta + k_3 \]
where

\[ k_1 = \epsilon_2 - \epsilon_{12}, \]  
\[ k_2 = (\epsilon_1 - d_1) \left( \frac{\epsilon_{12}}{\epsilon_1} \right) \left( \frac{\epsilon_2 - \epsilon_{12}}{1 - \epsilon_{12}} \right), \]  
\[ k_3 = (\epsilon_1 - d_1) \left( \frac{\epsilon_{12}}{\epsilon_1} \right) \left( \frac{1 - \epsilon_2}{1 - \epsilon_{12}} \right). \]

Having calculated the number of unknown variables sent to the weaker user from Phase II onwards, we now consider the number of equations received by the weaker user during Phases II and III. From Section 5.4.2, we know that during these phases, the total number of transmissions was simply equal to the number of trials needed to send \( N(\epsilon_1 - d_1) \) equations to the stronger user with feedback. The number of transmissions in Phases II through III is therefore distributed according to a negative binomial distribution and the mean number of transmissions in this period is \( W_{2,3} = N(\epsilon_1 - d_1)/(1 - \epsilon_1) \). Of these transmissions, the expected number received by the weaker user is equal to \( W_{2,3}(1 - \epsilon_2) \). We rewrite the expression for \( W_{2,3}(1 - \epsilon_2) \), the expected number of transmissions received by user \( 2 \) in Phases II through III, as \( NC_{2,3} \) where

\[ C_{2,3} = \frac{(\epsilon_1 - d_1)(1 - \epsilon_2)}{1 - \epsilon_1}. \]

We next compare \( N\mathcal{L}(\gamma, \theta) \), the amount of source symbols destined for the weaker user, with \( NC_{2,3} \), the expected number of equations received over the weaker user’s channel during Phases II and III.

As mentioned in Section 5.4.2, the weaker user requires an additional \( N(\epsilon_2 - d_2) \) symbols to be sent from Phase II onwards. Therefore, it is necessary that \( \mathcal{L}(\gamma, \theta) \geq \epsilon_2 - d_2 \). However, if \( \mathcal{L}(\gamma, \theta) \) is much greater than \( \epsilon_2 - d_2 \), we encounter the problem explained in the introduction of this section in which the weaker user is forced to decode unnecessary symbols thus introducing delay. Say that we are able to find values of \( \gamma', \theta' \in [0,1] \) such that \( \mathcal{L}(\gamma', \theta') = \epsilon_2 - d_2 \). We consider two cases when this is so – when \( \mathcal{L}(\gamma', \theta') \leq C_{2,3} \) and when \( \mathcal{L}(\gamma', \theta') > C_{2,3} \). We show that in both cases, we can achieve the optimal minmax latency so long as \( \mathcal{L}(\gamma', \theta') = \epsilon_2 - d_2 \).

In the first case, when \( \mathcal{L}(\gamma', \theta') \leq C_{2,3} \), we wish to send less information over the channel than what the channel can support. Therefore, we expect that the weaker user should be able to decode all source symbols before the conclusion of Phase III. However, in general, if the weaker user achieves distortion \( d_2 \) after decoding, it will be at a latency \( w \), where \( w > w_2(d_2) \). That is, in general, the weaker user may not achieve an individual point-to-point optimal latency.

To see why this is so, recall from Section 5.4.2 that in Phase II of our coding scheme, we transmit \( b_\pi(t) + v(t) \) at every time instant \( t \), where \( v(t) \) is a new random linear combination of the source symbols in \( F \) generated at every time \( t \). Since \( \mathcal{L}(\gamma', \theta') \leq C_{2,3} \), there is the possibility that at some point, the weaker user is able to decode all symbols in \( F \) even before Phase II has concluded. Say that this is the case and the stronger user has stalled on receiving a particular symbol \( b \in B_\pi \) being repeated. Let us further assume that the weaker user has already received \( b \). Then while \( b + v(t) \) is being transmitted, all transmissions to the weaker user are redundant. After \( b \) is received by the stronger user and the transmitter moves on to \( b' \), the next symbol in \( B_\pi \) to be sent via the modified repetition scheme, the weaker user can continue to receive innovative information. However, the set of transmissions received while \( b \) is being repeated prevents the weaker user from achieving an optimal individual latency.

However, we show that the optimal minmax latency can still be achieved. Notice that the moment all
Table 5.1: We justify the ordering of the region boundaries for \( d_1 \). In the left column, we have the ordering between two boundary points, and in the right column, we show the necessary and sufficient condition that justifies the ordering.

<table>
<thead>
<tr>
<th>Inequality</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12} &lt; \epsilon_1\epsilon_{12} )</td>
<td>( (1 - \epsilon_{12})^2 &gt; 0 )</td>
</tr>
<tr>
<td>( \epsilon_1\epsilon_{12} &lt; \epsilon_1 )</td>
<td>( \epsilon_{12} &lt; 1, \epsilon_1 &gt; 0 )</td>
</tr>
</tbody>
</table>

symbols in \( F \) can be decoded by the weaker user, the random linear combination \( v(t) \) can be subtracted from any transmission \( b_F(t) + v(t) \) in Phase II. Therefore the remainder of Phase II effectively consists of uncoded transmissions from the weaker user’s perspective, and he is eventually able to decode all \( \mathcal{L}(\gamma', \theta') \) symbols. Thus, so long as \( \mathcal{L}(\gamma', \theta') = \epsilon_2 - d_2 \), the weaker user will decode the necessary amount of symbols before the conclusion of Phase III, while the stronger user decodes at an optimal latency the moment Phase III terminates. In this case, the stronger user is the bottleneck of the system and in fact, the condition \( \mathcal{L}(\gamma', \theta') \leq C_{2,3} \) is equivalent to \( w_1^*(d_1) \geq w_2^*(d_2) \).

On the other hand, if \( \mathcal{L}(\gamma', \theta') > C_{2,3} \), the weaker user has more unknown variables than equations and so he cannot yet decode at the conclusion of Phase III. However, every transmission he has received so far is “innovative” in the sense that it provides an independent equation that can be used to decode the \( N\mathcal{L}(\gamma', \theta') \) source symbols. In order to decode, we simply need to send more equations to the weaker user, and since \( \mathcal{L}(\gamma', \theta') = \epsilon_2 - d_2 \), there will not be any unnecessary source symbols sent. Since the stronger user is point-to-point optimal at the conclusion of Phase III, we have therefore sent a total of \( NW_1^*(d_1) \) transmissions up to that point. Since, from the weaker user’s perspective, we have hitherto been sending random linear combinations of \( N\mathcal{L}(\gamma', \theta') \) variables, we simply need to continue doing so for another \( N(w_2^*(d_2) - w_1^*(d_1)) \) transmissions in Phase IV before he receives the remaining number of equations required and achieves point-to-point optimal performance.

We therefore see that regardless of whether \( \mathcal{L}(\gamma, \theta) \) is greater or less than \( C_{2,3} \), we can achieve an optimal minmax latency so long as we can find \( \gamma, \theta \in [0, 1] \) such that \( \mathcal{L}(\gamma, \theta) = \epsilon_2 - d_2 \). We focus on finding these values of \( \gamma \) and \( \theta \) in the next sections. In doing so, we consider three cases cases for \( d_1 \). The first is when \( 0 \leq d_1 \leq \epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12} \), the second when \( \epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12} < d_1 < \epsilon_1\epsilon_{12} \), and third when \( \epsilon_1\epsilon_{12} \leq d_1 < \epsilon_1 \). We justify the position of these boundary points with Table 5.1. For example, in the first row of Table 5.1, we justify that the boundary point \( \epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12} \) is less than the boundary point \( \epsilon_1\epsilon_{12} \) with the necessary and sufficient condition that \( (1 - \epsilon_{12})^2 > 0 \).

After dividing the values of \( d_1 \) into regions, we then further consider regions of \( d_2/\epsilon_2 \), where each region requires a distinct choice for the values of \( \gamma \) and \( \theta \). Recall from Remark 1 in Section 3.2.1 that for \( i \in \{1, 2\} \), we assume that \( d_i/\epsilon_i < 1 \), otherwise the coding is trivial. Therefore, we consider only values of \( d_2/\epsilon_2 \in [0, 1] \). In the following sections, the regions of \( d_2/\epsilon_2 \) will depend on the boundaries \( a^\dagger, b^\dagger, c^\dagger \) and \( d^\dagger \), which we define as
Table 5.2: We justify the ordering of the region boundaries of Figure 5.3 when \( 0 \leq d_1 \leq \epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12} \).

In the left column, we have the ordering between two boundary points, and in the right column, we show the necessary and sufficient condition that justifies the ordering.

\[
\begin{array}{|c|c|}
\hline
\text{Inequality} & \text{Justification} \\
\hline
a^\dagger < d_1/\epsilon_1 & \epsilon_{12} < \epsilon_2 \\
\frac{d_1}{\epsilon_1} < b^\dagger & d_1 < \epsilon_1 \\
b^\dagger \leq c^\dagger & d_1 \leq \epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12} \\
c^\dagger < d^\dagger & \epsilon_{12} < 1 \\
d^\dagger < 1 & d_1 < \epsilon_1 \\
\hline
\end{array}
\]

Case I: \( 0 \leq d_1 \leq \epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12} \)

We first mention that the upper bound in the assumption \( d_1 \leq \epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12} \) can be either positive or negative depending on the values of \( \epsilon_1 \) and \( \epsilon_{12} \). In the case that the upper bound is positive and \( d_1 \) is in this region, we now go through the process of finding the values of \( \gamma \) and \( \theta \) such that \( \mathcal{L}(\gamma, \theta) = \epsilon_2 - d_2 \).

We begin by dividing the number line for \( d_2/\epsilon_2 \) in Figure 5.3, where we have justified the ordering of each boundary point with Table 5.2. For example, in the third row of the table, we justify the ordering that \( b^\dagger \leq c^\dagger \) with the necessary and sufficient condition that \( d_1 \leq \epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12} \), which is the assumption in this region of \( d_1 \) that we consider. We enumerate all regions of Figure 5.3 and provide the values of \( \gamma \) and \( \theta \).

Region V: In this region, we set \( \gamma = \theta = 0 \) and recover the unmodified repetition coding scheme discussed in the beginning of Section 5.4.2. In fact, we do not require that \( \mathcal{L}(\gamma, \theta) = \epsilon_2 - d_2 \) in this region. Instead, by repeating all \( N(\epsilon_1 - d_1) \) source symbols, by Lemma 8, we can work out that \( Nk_3 \) uncoded source symbols are received by the weaker user, where \( k_3 \) is given by (5.25c). We can confirm that in Region V, in which \( d_2/\epsilon_2 \geq d^\dagger \), the distortion requirement of the weaker user is sufficiently large so that it can be met by the amount of uncoded symbols he receives.

Region IV: In this region, we set \( \theta = 0 \), and \( \gamma = \gamma^* \) where

\[
\gamma^* = \frac{\epsilon_1(\epsilon_2 - d_2)(1 - \epsilon_{12}) - \epsilon_{12}(\epsilon_1 - d_1)(1 - \epsilon_2)}{\epsilon_1(1 - \epsilon_{12})(\epsilon_2 - \epsilon_{12})}.
\]
Within this region, the weaker user’s distortion constraint is sufficiently low so that additional coded symbols must be transmitted. We can confirm that this choice of $\gamma$ and $\theta$ results in $L(\gamma, \theta) = \epsilon_2 - d_2$ in (5.24). Furthermore, the conditions $\gamma^* \geq 0$ and $\gamma^* \leq 1$ are equivalent to $d_2/\epsilon_2 \leq d^\dagger$ and $d_2/\epsilon_2 \geq c^\dagger$ respectively, which are satisfied in this region.

**Region III:** In this region, we set $\gamma = 0$ and $\theta = \theta^*$, where

$$\theta^* = \frac{\epsilon_1 (\epsilon_2 - d_2) (1 - \epsilon_{12}) - \epsilon_{12} (\epsilon_1 - d_1) (1 - \epsilon_2)}{\epsilon_{12} (\epsilon_1 - d_1) (\epsilon_2 - \epsilon_{12})}.$$  \hspace{1cm} (5.32)

We can confirm that this choice of $\gamma$ and $\theta$ results in $L(\gamma, \theta) = \epsilon_2 - d_2$ in (5.24). Furthermore, the conditions $\theta^* \geq 0$ and $\theta^* \leq 1$ are equivalent to $d_2/\epsilon_2 \leq d^\dagger$ and $d_2/\epsilon_2 \geq b^\dagger$ respectively, which are satisfied in this region.

**Region II:** In this region, we set $\theta = 1$, $\gamma = \hat{\gamma}$, where

$$\hat{\gamma} = \frac{\epsilon_1 (\epsilon_2 - d_2) - \epsilon_{12} (\epsilon_1 - d_1)}{\epsilon_1 (\epsilon_2 - \epsilon_{12})}.$$  \hspace{1cm} (5.33)

We can confirm that this choice of $\gamma$ and $\theta$ results in $L(\gamma, \theta) = \epsilon_2 - d_2$ in (5.24). Furthermore, the conditions $\hat{\gamma} \geq 0$ and $\hat{\gamma} \leq 1$ are equivalent to $d_2/\epsilon_2 \leq b^\dagger$ and $d_2/\epsilon_2 \geq a^\dagger$ respectively, which are satisfied in this region.

**Region I:** In this region, we ignore feedback altogether and use the successive segmentation coding scheme of the previous chapter, which showed in Section 3.3.3 that both users can be point-to-point optimal if $d_2/\epsilon_2 \leq d_1/\epsilon_1$.

**Case II:** $\epsilon_1 (2\epsilon_{12} - 1)/\epsilon_{12} < d_1 < \epsilon_1 \epsilon_{12}$

We again divide the number line for $d_2/\epsilon_2$ in Figure 5.4 where we have justified the ordering of each boundary point with Table 5.3. Notice however, that in the ranges of $d_1$ we now consider, the boundary points $c^\dagger$ and $b^\dagger$ have swapped positions compared to Figure 5.3. We again enumerate all regions of Figure 5.4.

**Region IX:** In this region, we set $\gamma = \theta = 0$ and recover the unmodified repetition coding scheme discussed in the beginning of Section 5.4.2 (see the description of Region V in the previous section).

**Region VIII:** In this region, we set $\theta = 0$, and $\gamma = \gamma^*$ where $\gamma^*$ is given by (5.31) (see the description of Region IV in the previous section).

**Region VII:** In this region, we set $\theta = 1$, $\gamma = \hat{\gamma}$, where $\hat{\gamma}$ is given by (5.33) (see the description of Region II in the previous section).

**Region VI:** In this region, we again ignore feedback altogether and use the successive segmentation coding scheme of the previous chapter (see the description of Region I in the previous section).
Table 5.3: We justify the ordering of the region boundaries of Figure 5.4 when $\epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12} < d_1 < \epsilon_1\epsilon_{12}$. In the left column, we have the ordering between two boundary points, and in the right column, we show the necessary and sufficient condition that justifies the ordering.

<table>
<thead>
<tr>
<th>Inequality</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a^\dagger &lt; d_1/\epsilon_1$</td>
<td>$\epsilon_{12} &lt; \epsilon_1$</td>
</tr>
<tr>
<td>$d_1/\epsilon_1 &lt; c^\dagger$</td>
<td>$d_1 &lt; \epsilon_1\epsilon_{12}$</td>
</tr>
<tr>
<td>$c^\dagger &lt; b^\dagger$</td>
<td>$d_1 &gt; \epsilon_1(2\epsilon_{12} - 1)/\epsilon_{12}$</td>
</tr>
<tr>
<td>$b^\dagger &lt; d^\dagger$</td>
<td>$d_1 &lt; \epsilon_1$</td>
</tr>
<tr>
<td>$d^\dagger &lt; 1$</td>
<td>$d_1 &lt; \epsilon_1$</td>
</tr>
</tbody>
</table>

Figure 5.5: The different regions requiring separate coding schemes when $\epsilon_1\epsilon_{12} \leq d_1 < \epsilon_1$.

**Case III: $\epsilon_1\epsilon_{12} \leq d_1 < \epsilon_1$**

For the final case of $d_1$ we consider, we again divide the number line for $d_2/\epsilon_2$ in Figure 5.5, where we have justified the ordering of each boundary point with Table 5.4. Again, the values of $d_1$ we consider have resulted in some of the boundaries in Figure 5.5 moving relative to Figure 5.4, most notably that now $c^\dagger < d_1/\epsilon_1$. We again enumerate all regions of Figure 5.4.

**Region XII**: In this region, we set $\gamma = \theta = 0$ and recover the unmodified repetition coding scheme discussed in the beginning of Section 5.4.2 (see the description of Region V in the previous section).

**Region XI**: In this region, we set $\theta = 0$, and $\gamma = \gamma^*$ where $\gamma^*$ is given by (5.31) (see the description of Region IV in the previous section).

**Region X**: In this region, we again ignore feedback altogether and use the successive segmentation coding scheme of the previous chapter (see the description of Region I in the previous section).

Table 5.4: We justify the ordering of the region boundaries of Figure 5.5 when $\epsilon_1\epsilon_{12} \leq d_1 < \epsilon_1$. In the left column, we have the ordering between two boundary points, and in the right column, we show the necessary and sufficient condition that justifies the ordering.

<table>
<thead>
<tr>
<th>Inequality</th>
<th>Justification</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0 &lt; a^\dagger$</td>
<td>$d_1 &gt; 0$</td>
</tr>
<tr>
<td>$a^\dagger &lt; c^\dagger$</td>
<td>$d_1 &lt; \epsilon_1$</td>
</tr>
<tr>
<td>$c^\dagger &lt; d_1/\epsilon_1$</td>
<td>$d_1 \geq \epsilon_1\epsilon_{12}$</td>
</tr>
<tr>
<td>$d_1/\epsilon_1 &lt; d^\dagger$</td>
<td>$d_1 &lt; \epsilon_1$</td>
</tr>
<tr>
<td>$d^\dagger &lt; 1$</td>
<td>$d_1 &lt; \epsilon_1$</td>
</tr>
</tbody>
</table>
Chapter 6

Three-terminal Erasure Source-Broadcast with Feedback

We study the effects of introducing a feedback channel in the erasure source-broadcast problem for the case of three receivers. In our problem formulation, a feedback channel is available to all three receivers. We again have that each receiver requires a certain fraction of a source sequence, and we are interested in the minimum latency, or transmission time, required to serve them all.

We propose a queue-based hybrid digital-analog coding scheme that achieves optimal performance for the duration of analog transmissions. We propose a method of characterizing the number of analog transmissions that can be sent, which involves solving a linear program, and furthermore give sufficient conditions for which all users can be optimal. In some cases, we find that users can be point-to-point optimal regardless of their distortion constraints.

When the analog transmissions are insufficient in meeting user demands, we propose two subsequent coding schemes. The first uses a queue preprocessing strategy for channel coding. The second method is a novel chaining algorithm, which involves the transmitter targeting point-to-point optimal performance for two users as if they were the only users in the network. Meanwhile, the third user simultaneously builds “chains” of symbols that he may at times be able to decode based on the channel conditions, or with the subsequent reception of additional symbols. Finally, we also provide simulations that highlight the benefits of feedback.

6.1 Introduction

In our work, we utilize uncoded transmissions that are instantly-decodable, and distortion-innovative. The zero latency in decoding uncoded packets has benefits in areas in which packets are instantly useful at their destination such as applications in video streaming and disseminating commands to sensors and robots [36,66]. The availability of sending such uncoded transmissions is random and based on the channel noise. Typically, we set up queues that track which symbols are required by which group of users and subsequently find opportunities for network coding based on these queues.

There have been two primary techniques in deriving a rate region for a queue-based coding scheme. The first was derived from a channel coding problem involving erasure channels with feedback and memory [67]. In [67], the authors use a queue stability criterion to set up a system of inequalities.
involving the achievable rates and flow variables that represent the number of packets that are transferred between queues. When this technique is applied to the chaining algorithm of Section 6.3.2 however, we find that our system of inequalities is over-specified. That is, the large number of queues leaves us with more equations than unknown variables, and we are unable to solve for a rate region.

The technique of using a Markov rewards process to analyze the rate region of a queue-based algorithm was also used in [68]. In their work, the transmitter and receivers were modelled with a Markov rewards process. The authors solved for the steady-state distribution and were able to derive rate region by considering the number of rewards accumulated per timeslot and thus the number of timeslots it would take to send a group of packets. In our analysis in Section 6.3.2 however, we find that our transmitter and receiver must be modelled with an absorbing Markov rewards process. Thus, it is the transient behaviour that is important and we cannot use a steady-state analysis.

In Section 6.3.2, we modify the analysis for that of an absorbing Markov rewards process. The analysis does not involve the optimization of any parameters. Rather, we use the analysis of the absorbing Markov rewards process to state sufficient conditions for optimality. We illustrate the operational significance of this analysis in Section 6.4.3 where we show how Theorem 6 can be used to delineate regions of optimality. Finally, in Section 6.3.1, we also show how to formulate a linear program to derive a rate region by solving for the number of instantly-decodable, distortion innovative transmissions that can be sent.

6.2 System Model

We refer the reader to the system model that was described in Section 5.2. The problem formulation we address in this chapter is identical and specialized for the case when \( n = 3 \).

6.3 Coding for Three Users

Our proposed coding scheme consists of two distinct parts and is described within Sections 6.3.1 and 6.3.2. The first part involves only transmissions that are instantly decodable and distortion-innovative for all users, while the second part is executed if we are no longer able to send additional instantly decodable, distortion-innovative packets.

6.3.1 Instantly Decodable, Distortion-Innovative Transmissions

The first part of the code again involves an initial phase that transmits each source symbol until at least one user receives it. Again, no further processing of source symbol \( S(t) \) is done if it was received by all users. Let \( \mathcal{E}(T) \subset \mathcal{U} \triangleq \{1, 2, 3\} \) represent the set of users whose channel output was an erasure at time \( T \). We will at times drop the time index \( T \) when it is obvious from the context. Then after the transmission at time \( T \), we place \( S(t) \) in queue \( Q_{\mathcal{E}(T)} \) so that, in general, \( Q_U \) maintains a record of which symbols were not received by all users \( i \in U \).

The algorithm contains a subroutine that acts as an event handler. In the event that any user \( i \)'s distortion constraint is met during or after the analog transmissions, the algorithm’s control flow is passed to this subroutine. Within this subroutine, we simply use the two-user algorithm outlined in Section 5.3 to serve the remaining users. We do this by first discarding \( Q_i \) since users \( j \) and \( k \) have
already received all symbols from this queue for $j, k \in \mathcal{U} \setminus \{i\}, j \neq k$. We then merge queues $Q_j$ and $Q_{\{i,j\}}$ and queues $Q_i$ and $Q_{\{i,k\}}$, since user $i$ is no longer relevant, and symbols in $Q_{\{i,j\}}$ can be regarded as symbols that only user $j$ needs at this point. Finally, since $Q_{\{j,k\}}$ contains source symbols that neither user has received so far, we can treat them as source symbols that have yet to be sent uncoded in the algorithm of Section 5.3.

If there are no users with distortion constraints met however, the algorithm continues by sending linear combinations in a similar manner as that in Section 5.3. First, consider $Q_1$ and $Q_{\{2,3\}}$. If they are both non-empty, let $q_1 \in Q_1$ and $q_{2,3} \in Q_{\{2,3\}}$. Notice that if we transmit $q_1 \oplus q_{2,3}$, since user 1 has access to $q_{2,3}$, and user 2 and user 3 have access to $q_1$, every user is able to receive an instantly decodable, distortion-innovative symbol (See Section 5.3). A symbol is removed from a queue if any user for whom the queue is intended receives the linear combination. However, if we have a type of situation where only user 2 receives the linear combination but not user 3, then $q_{2,3}$ is transferred from $Q_{\{2,3\}}$ to $Q_3$, since only user 3 is in need of this symbol now. We continue this procedure with queues $Q_2$ and $Q_{\{1,3\}}$ and queues $Q_3$ and $Q_{\{1,2\}}$ to the extent that it is possible, i.e., for as long as the relevant queues remain non-empty. Upon the emptying of these queues, we determine if we can send linear combinations of the form $q_1 \oplus q_2 \oplus q_3$, which are similarly instantly decodable and distortion-innovative. If this is possible, we again continue to do so until no longer possible.

We mention that in general, for $n$ users, we can similarly maintain queues that manage which symbols are erased by which users. In this case, we can send an instantly decodable, distortion-innovative symbol if there exists non-empty queues $Q_{\Gamma_1}, Q_{\Gamma_2}, \ldots, Q_{\Gamma_m}$ such that $\bigcup_{l=1}^m \Gamma_l = [n]$ and $\sum_{l=1}^m |\Gamma_l| = n$. These two conditions ensure that the index $j$ for user $j$ appears in exactly one $\Gamma_l, l \in [m]$. Thus, user $j$ is in possession of all symbols in these queues except for $Q_{\Gamma_1}$. We then send the linear combination $\sum_{l=1}^m q_{\Gamma_l}$, where $q_{\Gamma_l} \in Q_{\Gamma_l}$. We note however, that this requires an amount of queues that is exponential in the number of users.

### 6.3.2 Non-Instantly-Decodable, Distortion-Innovative Coding

When we have exhausted the queues that allow us to transmit instantly decodable, distortion-innovative symbols, we are left with two possibilities for the types of queues remaining. By assumption, we have that for all $i \in \mathcal{U}$, either $Q_i$ or $Q_{\mathcal{U}\setminus\{i\}}$ is empty, and there exists an $l \in \mathcal{U}$ such that $Q_l$ is empty. Thus, after the stopping condition of the algorithm in Section 6.3.1 has been reached, we are either left with queues $Q_{\{1,2\}}, Q_{\{1,3\}}$ and $Q_{\{2,3\}}$, or queues $Q_i$, $Q_{\{i,j\}}$ and $Q_{\{i,k\}}$ (we will often use the indices $i, j$ and $k$ to refer to unique elements in $\mathcal{U}$). We assume that the latter is the case and propose two methods to address this. Analogous methods (omitted for brevity) can be used for the former case.

The first method involves preprocessing the remaining queues before using a channel coding scheme to satisfy the users’ remaining demands. The second method is a “chaining algorithm,” which involves the transmitter using the algorithm in Section 5.3 to send instantly-decodable, distortion-innovative transmissions to two users as if they were the only users in the network. Meanwhile, the third user simultaneously builds “chains” of symbols that he may at times be able to decode based on the channel conditions, or with the subsequent reception of additional symbols. We begin by describing the queue preprocessing method for channel coding.
Recall that user \( q \) remain, we can send linear combinations of the form point-to-point optimal performance for the remaining users. Specifically, since queues \( Q \) to send instantly-decodable, distortion-innovative symbols which happens when we are left with queues \( Q \) in the previous section. Recall that we invoke this algorithm when we have exhausted all opportunities In this section, we propose an alternative to using queue preprocessing for channel coding as described A Chaining Algorithm for \( n \) well as the one from Section 6.3.1. Finally, we mention that a generalization of this section’s approach which, we have that the total transmission time is the sum of the time required for this algorithm as symbols, the distortion constraints in (6.1) follow.

\[
\min_{\delta} \frac{\delta_i^j + \delta_{i,j}^k + \delta_{i,k}^j}{1 - \epsilon_i} + \frac{\delta_{i,k}^j + \delta_{i,k}^j}{1 - \epsilon_j} + \\
+ \frac{\delta_{i,k}^j}{1 - \max(\epsilon_i, \epsilon_k)} + \frac{\delta_{i,k}^j}{1 - \max(\epsilon_i, \epsilon_j)} \tag{6.1}
\]

subject to \( \delta \geq 0 \)
\[
\delta_i^j \leq |Q_i|, \\
\delta_{i,k}^j + \delta_{i,k}^j + \delta_{i,k}^j \leq |Q_{\{i,k\}}|, \\
\delta_{i,j}^j + \delta_{i,j}^j + \delta_{i,j}^j \leq |Q_{\{i,j\}}|, \\
\delta_i^j + \delta_{i,j}^j + \delta_{i,j}^j + \delta_{i,k}^j \geq 1 - d_i - r_i, \\
\delta_{i,j}^j + \delta_{i,j}^j \geq 1 - d_j - r_j, \\
\delta_{i,k}^j + \delta_{i,k}^j \geq 1 - d_k - r_k,
\]

Queue Preprocessing for Channel Coding

The channel coding scheme we employ is based on [40], which outlines a method of losslessly communicating a set of messages to \( n \) users when feedback is available at the transmitter. As input, it takes a set of queues \( \{Q_U \mid U \subseteq U\} \) where \( U \) is a subset of users each of whom must losslessly reconstruct all symbols in \( Q_U \).

We preprocess queues \( Q_i, Q_{\{i,j\}} \) and \( Q_{\{i,k\}} \) to determine queues \( \hat{Q}_i, \hat{Q}_j, \hat{Q}_k, \hat{Q}_{\{i,j\}}, \hat{Q}_{\{i,k\}} \) that will be passed as input to the channel coding algorithm. Let \( \delta_P^R \) denote the number of symbols taken from queue \( Q_R \) and placed in \( \hat{Q}_T \). Thus, we have, for example, that \( |\hat{Q}_i| = \delta_i^j + \delta_{i,j}^j + \delta_{i,k}^j \). Given that user \( i \) has received \( r_i \) symbols so far, we determine all \( \delta \)'s by solving the linear program in (6.1). Here, each term in the objective function represents the minimum latency required to process its corresponding queue, i.e., the first term represents the latency required to process \( Q_i \), the second is for \( Q_k \), etc. Furthermore, we have that user \( i \) is able to decode \( \delta_P^R \) symbols if \( i \in T \), and so given that this user has already received \( r_i \) symbols, the distortion constraints in (6.1) follow.

After solving (6.1), we run the channel coding algorithm on the newly determined queues, after which, we have that the total transmission time is the sum of the time required for this algorithm as well as the one from Section 6.3.1. Finally, we mention that a generalization of this section’s approach for \( n \) users is an ongoing work.

A Chaining Algorithm

In this section, we propose an alternative to using queue preprocessing for channel coding as described in the previous section. Recall that we invoke this algorithm when we have exhausted all opportunities to send instantly-decodable, distortion-innovative symbols which happens when we are left with queues \( Q_{\{1,2\}}, Q_{\{1,3\}} \) and \( Q_{\{2,3\}} \), or queues \( Q_i, Q_{\{i,j\}} \) and \( Q_{\{i,k\}} \). We again assume that the latter is the case.

We first consider if we were to ignore user \( i \) and simply use the coding scheme of Section 5.3 to target point-to-point optimal performance for the remaining users. Specifically, since queues \( Q_{\{i,j\}} \) and \( Q_{\{i,k\}} \) remain, we can send linear combinations of the form \( q_{i,j} \oplus q_{i,k} \), where \( q_{i,j} \in Q_{\{i,j\}} \) and \( q_{i,k} \in Q_{\{i,k\}} \). Recall that user \( j \) has received all symbols in \( Q_{\{i,k\}} \), and user \( k \) has similarly received all symbols in \( Q_{\{i,j\}} \). In each transmission, each user can therefore subtract off one of the symbols from the linear
Table 6.1: A hypothetical sequence of transmissions and channel noises when targeting point-to-point optimal performance for user $j$ and user $k$.

<table>
<thead>
<tr>
<th>$t$</th>
<th>$X(t)$</th>
<th>$Z_i(t)$</th>
<th>$Z_j(t)$</th>
<th>$Z_k(t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$q_{i,j} \oplus q_{i,k}$</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>$\hat{q}<em>{i,j} \oplus q</em>{i,k}$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>$2\hat{q}<em>{i,j} \oplus q</em>{i,k}$</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
State | $X(t)$ – Transmission at time $t$ | Precondition at $t$
---|---|---
1 | $q_{i,j} \oplus q_{i,k}$ (Initial state) | $|\mathcal{C}(t-1) \cap \mathcal{T}(t)| = 0$
2 | $\hat{q}_{i,j} \oplus q_{i,k}$ | State 1 occupied at time $t-1$
3 | $q_{i,j} \oplus \hat{q}_{i,k}$ | $|\mathcal{C}(t-1) \cap \mathcal{T}(t)| = 1$
4 | $2q_{i,j} \oplus q_{i,k}$ | $|\mathcal{C}(t-1) \cap \mathcal{T}(t)| = 1$
5 | None (Non-decoding absorbing state) | $|\mathcal{C}(t-1) \cap \mathcal{T}(t)| = 0$
6 | None (Decoding absorbing state) | State 4 occupied at time $t-1$

Table 6.2: A description of states in the Markov rewards process. Assume that $q_{i,j} \oplus q_{i,k}$ was sent at time $t = 3$. If the state indicated in the first column is entered, then the transmission indicated by the second column is sent at time $t$. We also show the precondition necessary to enter the state in the third column.

In which we are not be able to decode the chain of symbols if at $t = 3$, we instead had that only users $j$ and $k$ received $X(3)$. In this case, we would have to replace both symbols being sent at $t = 4$ in order to be point-to-point optimal for users $j$ and $k$. Thus, user $i$ would not have enough equations to decode, and we would instead reset the Markov process such that user $i$ would start building another chain when the new symbols are sent.

In addition to the two absorbing states, the Markov rewards process involves four additional transient states. We enumerate and describe all states in Table 6.2. In this table, we assume that $q_{i,j} \oplus q_{i,k}$ was sent at time $t - 1$. Then if state $l \in \Omega \triangleq \{1, 2, \ldots, 6\}$ is entered at time $t$, we show $l$ in the first column and $X(t)$, the transmission at time $t$, in the second column. We also show a necessary precondition for entering state $l$ at time $t$ in the third column.

Before we explain the preconditions, let us first define some notation. Let $\mathcal{C}(t)$ represent the set of symbols appearing in the linear combinations received by user $i$ after having listened to all transmissions from the beginning of the chaining algorithm up to and including time $t$. In the example of Table 6.1, after time $t = 1, 2, 3$, we have $\mathcal{C}(1) = \{q_{i,j}, q_{i,k}\}$, $\mathcal{C}(2) = \{q_{i,j}, q_{i,k}, \hat{q}_{i,j}\}$ and $\mathcal{C}(3) = \{q_{i,j}, q_{i,k}, \hat{q}_{i,j}\}$. Let $\mathcal{T}(t)$ represent the set of symbols appearing in the linear combination in $X(t)$. In the example of Table 6.1, we have that $\mathcal{T}(1) = \{q_{i,j}, q_{i,k}\}$, $\mathcal{T}(2) = \{q_{i,j}, q_{i,k}\}$, and $\mathcal{T}(3) = \{q_{i,j}, q_{i,k}\}$. With this notation, we now describe the preconditions of Table 6.2.

The first state is the initial state that sends the initial linear combination of symbols. After the first outbound transition from the initial state, the initial state is not returned to at any time during the evolution of the Markov reward process unless an absorbing state is reached and we restart the chain-building process. Since user $i$ has not yet received any equations and we begin by transmitting a linear combination of two symbols, we have the precondition that $\mathcal{C}(t-1) = \emptyset$, $\mathcal{T}(t) = \{q_{i,j}, q_{i,k}\}$ and $|\mathcal{C}(t-1) \cap \mathcal{T}(t)| = 0$.

The second state is entered only if the symbol destined for user $j$ needs to be replaced in the next linear combination to be sent and the chain-building process has begun. In order to ensure that we can continue building the chain of symbols, we require that at least one symbol being sent at time $t$ appears in a linear combination already received by user $i$, i.e., $|\mathcal{C}(t-1) \cap \mathcal{T}(t)| = 1$. Similarly the third state is entered only if the symbol destined for user $k$ needs to be replaced in the next linear combination to be sent and the chain-building process has already started. The precondition for this state is analogous to its counterpart in State 2.

The fourth state is entered if only user $i$ received the previous transmission, in which case we need...
to send a new linear combination of the same two symbols in the previous timeslot. In this case, we see that since the transmission is just a different linear combination of the previous two symbols sent, $|C(t-1) \cap T(t)| = 2$.

Finally, the two remaining states are the absorbing states. State 5 is the absorbing state we occupy if the constraint of being point-to-point optimal for users $j$ and $k$ prevent user $i$ from continuing the chain-building process. That is, we are not able transmit the next linear combination such that one of the symbols appearing in the combination also appears in a previously received equation, i.e., $|C(t-1) \cap T(t)| = 0$. Lastly, State 6 is the absorbing state we occupy if we reach the point when we are able to decode all symbols in the chain. It is reached if the transmission sent in State 4 was received by user $i$.

Having described the purpose of each state, we now describe the transitions between states in detail beginning with State 1. Let $Z(t) = (Z_i(t), Z_j(t), Z_k(t))$. For each state, we use a table to describe the outbound transitions based on values of $Z(t)$. In addition, we show the reward accumulated for each transition. For example, in the column with the heading $\rho_j$ in Table 6.4, we show the reward accumulated by user $j$ at each possible outbound transition from State 1 based on the channel noise. We note that although $\rho_j$ depends on both $Z(t)$ and the inbound and outbound states, for notational convenience, we omit explicitly stating this dependence. The reward can be accumulated by a user, or a queue. The notation for rewards is summarized in Table 6.3. In our description of each state, we also define the queue $Q^*$, which is the queue containing prioritized symbols that only user $i$ requires. These symbols are prioritized because the reception of this symbol can decode an entire chain of symbols.

In the following descriptions, the reader can confirm that the chaining algorithm is point-to-point optimal for users $j$ and $k$. Each time a symbol is received by one of the users, it is replaced by another instantly-decodable, distortion-innovative symbol.

**State 1:** We enumerate all outgoing transitions from State 1, the initial state, based on all possible channel conditions in Table 6.4.

1. $Z(t) = (0, 0, 0)$. In this case, all users have received the transmission. We therefore see that $\rho_j = \rho_k = 1$, since users $j$ and $k$ can each decode a symbol. Since this is the initial state however, user $i$ has not received any previous equations, and therefore has only one equation in two unknown variables. Since we must replace both symbols in the next transmission, we cannot continue building the current chain. The next state is therefore State 5, the absorbing state in which we are not able to decode any symbols in the chain. However, we do increase the number of equations received by user $i$ by setting $\rho_E = 1$. Furthermore, we arbitrarily place one of the symbols from the linear combination into $Q^*$, the queue containing symbols that can decode an entire chain of symbols. We set $\rho_{Q^*} = 1$ to reflect this.

### Table 6.3: A legend for the reward variables.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_u(l,m)$</td>
<td>Reward representing the number of symbols that can be decoded by user $u$ after a transition from state $l$ to $m$ for $u \in {j,k}$</td>
</tr>
<tr>
<td>$\rho_Q(l,m)$</td>
<td>Reward representing the number of symbols placed in queue $Q$ after a transition from state $l$ to $m$.</td>
</tr>
<tr>
<td>$\rho_E(l,m)$</td>
<td>Reward representing the number of equations user $i$ received after a transition from state $l$ to $m$.</td>
</tr>
</tbody>
</table>
We enumerate all outgoing transitions from State 2 based on all possible channel conditions. The table includes the probability of each transition and the reward accumulated for each transition.

Table 6.4: A detailed table of all outgoing transitions from State 1 based on the channel noise realization. The table includes the probability of each transition and the reward accumulated for each transition.

<table>
<thead>
<tr>
<th>State 1 Outgoing Transitions</th>
<th>Next State</th>
<th>ρj</th>
<th>ρk</th>
<th>ρE</th>
<th>ρQ_i</th>
<th>ρQ_j</th>
<th>ρQ_k</th>
<th>ρQ_κ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. (0, 0, 0) (1 - ϵ_i)(1 - ϵ_j)(1 - ϵ_k)</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2. (0, 0, 1) (1 - ϵ_i)(1 - ϵ_j)ϵ_k</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3. (0, 1, 0) (1 - ϵ_i)ϵ_j(1 - ϵ_k)</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4. (0, 1, 1) (1 - ϵ_i)ϵ_jϵ_k</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5. (1, 0, 0) ϵ_i(1 - ϵ_j)(1 - ϵ_k)</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6. (1, 0, 1) ϵ_i(1 - ϵ_j)ϵ_k</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7. (1, 1, 0) ϵ_iϵ_j(1 - ϵ_k)</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8. (1, 1, 1) ϵ_iϵ_jϵ_k</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

2. Z(t) = (0, 0, 1). In this case, only users i and j have received the transmission. We set ρ_j = ρ_E = 1 to indicate that user j can decode one symbol, and user i has received one more equation. Since the symbol for user j will be replaced in the next transmission, and the chain-building process has just begun, we therefore have that the next state is State 2.

3. Z(t) = (0, 1, 0). Similar to the previous case, however, we now have that only users i and k have received the transmission. We set ρ_k = ρ_E = 1 to indicate that user k can decode one symbol, and user i has received one more equation. Since the symbol for user k will be replaced in the next transmission, and the chain-building process has just begun, we therefore have that the next state is State 3.

4. Z(t) = (0, 1, 1). In this case, only user i has received the transmission so we set only ρ_E = 1. For the subsequent timeslot, we must send a different linear combination of the same two symbols previously sent and so the next state is State 4.

5. Z(t) = (1, 0, 0). In this case, both users j and k can decode a symbol so we set ρ_j = ρ_k = 1. If q_{i,j} ⊕ q_{i,k} was sent in the previous timeslot, we now have that both q_{i,j} and q_{i,k} are required by only user i. We therefore place both these symbols in Q_i, and we set ρ_{Q_i} = 2 to reflect this. Since both q_{i,j} and q_{i,k} will not be sent in the next transmission, we have that the next state is the non-decoding absorbing state, State 5.

6. Z(t) = (1, 0, 1). In this case, only user j received the transmission so we set ρ_j = 1. Notice now that q_{i,j} is no longer needed by user j. Although we must replace this symbol in the next transmission, the chain-building process has not yet begun and user i has still not received any equations involving q_{i,j}. We therefore place this symbol in Q_i, set ρ_{Q_i} = 1 to reflect this, and return to State 1 in the next timeslot.

7. Z(t) = (1, 1, 0). Similar to the previous case, only user k has received the transmission and so we set ρ_k = ρ_{Q_i} = 1, and return to State 1 in the next timeslot.

8. Z(t) = (1, 1, 1). In this case, no users have received the transmission. No rewards are assigned and we return to State 1 to retransmit the same linear combination.

State 2: We enumerate all outgoing transitions from State 2 based on all possible channel conditions in Table 6.5. We remind the reader that State 2 is characterized by user i having already started the chain-building process and the transmitter having just replaced the symbol intended for user j.
For brevity, we omit the details for State 3 and simply state that it is analogous to the discussion we have just given for State 2 with the roles of users $j$ and $k$ reversed.

### State 2 Outgoing Transitions

<table>
<thead>
<tr>
<th>$Z(t)$</th>
<th>$\Pr(Z(t))$</th>
<th>$\text{Next State}$</th>
<th>$p_j$</th>
<th>$p_k$</th>
<th>$p_{E}$</th>
<th>$p_{Q_j}$</th>
<th>$p_{Q_k}$</th>
<th>$p_{Q^*}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0, 0)</td>
<td>$(1 - \epsilon_i)(1 - \epsilon_j)(1 - \epsilon_k)$</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(0, 0, 1)</td>
<td>$(1 - \epsilon_i)(1 - \epsilon_j)\epsilon_k$</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(0, 1, 0)</td>
<td>$(1 - \epsilon_i)\epsilon_j(1 - \epsilon_k)$</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(0, 1, 1)</td>
<td>$(1 - \epsilon_i)\epsilon_j\epsilon_k$</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(1, 0, 0)</td>
<td>$\epsilon_i(1 - \epsilon_j)(1 - \epsilon_k)$</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>(1, 0, 1)</td>
<td>$\epsilon_i(1 - \epsilon_j)\epsilon_k$</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(1, 1, 0)</td>
<td>$\epsilon_i\epsilon_j(1 - \epsilon_k)$</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>(1, 1, 1)</td>
<td>$\epsilon_i\epsilon_j\epsilon_k$</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Table 6.5**: A detailed table of all outgoing transitions from State 2 based on the channel noise realization. The table includes the probability of each transition and the reward accumulated for each transition.

in the linear combination being sent. Therefore, the symbol intended for user $k$ is the common symbol between $C(t - 1)$ and $T(t)$ in the precondition of Table 6.2. We now go over the possible transitions based on whether this new linear combination was received by any of the receivers.

1. $Z(t) = (0, 0, 0)$. This case is analogous to its counterpart in State 1. The subsequent state is again State 5, the non-decoding absorbing state, since user $i$ has has one less equation required to decode the number of unknown variables in the chain. The transition to the subsequent state and the rewards allocated are identical for when $Z(t) = (0, 0, 0)$ in State 1.

2. $Z(t) = (0, 0, 1)$. This case is analogous to its counterpart in State 1.

3. $Z(t) = (0, 1, 0)$. This case is analogous to its counterpart in State 1.

4. $Z(t) = (0, 1, 1)$. This case is analogous to its counterpart in State 1.

5. $Z(t) = (1, 0, 0)$. This case is similar to its counterpart in State 1 with one minor difference due to the fact that the chain-building process has already begun. Recall that in this case, we have that both symbols in the previous linear combination sent will have to be replaced in the next timeslot. However, instead of placing both these symbols in $Q_i$, we should place one of them in $Q^*$, since it can be used to decode an entire chain of symbols. We therefore set $p_{Q^*} = p_{Q^*} = 1$.

6. $Z(t) = (1, 0, 1)$. This case is analogous to its counterpart in State 1, however since the chain-building process has begun, we transition to State 2 again in the next timeslot.

7. $Z(t) = (1, 1, 0)$. This case is in direct contrast from its counterpart in State 1. Recall from the beginning of the description for State 2 that the symbol intended for user $k$ is the symbol in common between $C(t - 1)$ and $T(t)$. Furthermore, given that $Z(t) = (1, 1, 0)$, user $k$ has now received this symbol and so we must replace it in the next timeslot. However, doing so would not allow us to continue building the current chain because there will be no symbols in common between $C(t)$ and $T(t + 1)$. Therefore, we transition to the non-decoding absorbing decoding state in the next timeslot and place the symbol that was intended for user $k$ in $Q^*$.

8. $Z(t) = (1, 1, 1)$. This case is analogous to its counterpart in State 1, except we transition back to State 2 in the next timeslot.
We enumerate all outgoing transitions from State 4 based on all possible channel conditions in Table 6.6: A detailed table of all outgoing transitions from State 4 based on the channel noise realization. The table includes the probability of each transition and the reward accumulated for each transition.

**State 4**: We enumerate all outgoing transitions from State 4 based on all possible channel conditions in Table 6.6. We remind the reader that State 4 is characterized by user \( i \) having just been the only receiver to receive the previous transmission. Thus, the chain-building process has begun and in this timeslot, the transmitter will send a different linear combination of the previous two symbols sent. Finally, we note that this state has outbound transitions in which a source symbol may be placed in \( Q_j \) or \( Q_k \) after time \( t \). In the event that one or both of these queues are nonempty, and depending on whether \( Q_i \) is nonempty, the transmitter may also have the option of temporarily suspending the chaining algorithm to send linear combinations of the form

\[
q_i \oplus q_j \oplus q_k, \quad (6.2a)
\]

\[
q_j \oplus q_{i,k}, \quad (6.2b)
\]

\[
q_k \oplus q_{i,j}, \quad (6.2c)
\]

We now go over the possible state transitions based on whether the linear combination sent at time \( t \) was received by any of the receivers.

1. **\( Z(t) = (0, 0, 0) \)**. In this case, user \( i \) has now received as many equations as there are unknown variables in the chain. Thus the subsequent state is State 6, the decoding absorbing state, and there is no need to set \( \rho_{Q^*} = 1 \), since all variables in the chain have been accounted for.

2. **\( Z(t) = (0, 0, 1) \)**. In this case, user \( i \) can again decode. Furthermore, we place the source symbol intended for user \( k \) into \( Q_k \) and set \( \rho_{Q_k} = 1 \).

3. **\( Z(t) = (0, 1, 0) \)**. Similar to the previous case, user \( i \) can also decode given this channel noise realization, and we place the source symbol intended for user \( j \) into \( Q_j \) and set \( \rho_{Q_j} = 1 \).

4. **\( Z(t) = (0, 1, 1) \)**. In this case, user \( i \) can again decode, and we place the source symbol intended for users \( j \) and \( k \) into \( Q_j \) and \( Q_k \) respectively and set \( \rho_{Q_j} = \rho_{Q_k} = 1 \).

5. **\( Z(t) = (1, 0, 0) \)**. This case is similar to its counterpart in State 2, however, since both symbols appearing in \( X(t) \) have been accounted for, i.e., both symbols appearing in \( X(t) \) are in the set \( C(t-1) \), there is no need to set \( \rho_{Q_i} = 1 \).

6. **\( Z(t) = (1, 0, 1) \)**. This case is analogous to its counterpart in State 2, however, since the symbol just received by user \( j \) has been accounted for, i.e., since it is one of the symbols in \( C(t-1) \), there is no need to set \( \rho_{Q_i} = 1 \).

<table>
<thead>
<tr>
<th>( Z(t) )</th>
<th>( \Pr(Z(t)) )</th>
<th>Next State</th>
<th>( \rho_j )</th>
<th>( \rho_k )</th>
<th>( \rho_{Q_j} )</th>
<th>( \rho_{Q_k} )</th>
<th>( \rho_{Q^*} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ( (0, 0, 0) )</td>
<td>((1 - \epsilon_i)(1 - \epsilon_j)(1 - \epsilon_k))</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2. ( (0, 0, 1) )</td>
<td>((1 - \epsilon_i)(1 - \epsilon_j)\epsilon_k)</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3. ( (0, 1, 0) )</td>
<td>((1 - \epsilon_i)\epsilon_j(1 - \epsilon_k))</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4. ( (0, 1, 1) )</td>
<td>((1 - \epsilon_i)\epsilon_j\epsilon_k)</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>5. ( (1, 0, 0) )</td>
<td>(\epsilon_i(1 - \epsilon_j)(1 - \epsilon_k))</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6. ( (1, 0, 1) )</td>
<td>(\epsilon_i(1 - \epsilon_j)\epsilon_k)</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7. ( (1, 1, 0) )</td>
<td>(\epsilon_i\epsilon_j(1 - \epsilon_k))</td>
<td>3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8. ( (1, 1, 1) )</td>
<td>(\epsilon_i\epsilon_j\epsilon_k)</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
7. \( Z(t) = (1, 1, 0) \). This case is analogous to its counterpart in State 2, however, since the symbol just received by user \( k \) has been accounted for, i.e., since it is one of the symbols in \( C(t-1) \), there is no need to set \( \rho_{Q_i} = 1 \).

8. \( Z(t) = (1, 1, 1) \). This case is analogous to its counterpart in State 1, except we transition back to State 4 in the next timeslot.

Finally, we conclude our discussion of the chaining algorithm by mentioning that if at any time an absorbing state in the Markov rewards process is reached and the distortion constraints of users \( j \) and \( k \) have yet to be met, then we simply restart the Markov rewards process and begin building a new chain. If at any time during this process, one of the users has their distortion constraint met, then that user can leave the network and we are left with two users who we again serve with the algorithm of Section 5.3. If one of the remaining two users is user \( i \), the user building the chain, then we would prioritize the transmission of symbols in \( Q^* \), since the reception of one of these symbols would lead to the decoding of an entire chain of symbols.

6.4 Analysis

In this section, we analyze the coding scheme proposed in Section 6.3.1, and the chaining algorithm of Section 6.3.2. For the analysis of the coding scheme in Section 6.3.1, we propose that the solution of a linear program characterizes the number of instantly decodable, distortion-innovative symbols that can be sent by the algorithm in Section 6.3.1. We furthermore give sufficient conditions for which all users can be point-to-point optimal. In some cases, we find that users can be point-to-point optimal regardless of their distortion constraints. Following this, we analyze the chaining algorithm and give a sufficient condition for point-to-point optimal performance.

6.4.1 Analysis of Coding Scheme of Section 6.3.1

Our analysis begins by first considering the systematic phase of our coding, which transmits the \( N \) source symbols so that each symbol is recovered by at least one user. Let \( T_0 \) be a random variable representing the number of transmissions required for this after being normalized by \( N \). In other words, \( T_0 \) will be said to be the latency required. Then it is clear that \( T_0 \) has expected value \( \bar{T}_0 \) given by

\[
\bar{T}_0 = \frac{1}{1 - \epsilon_1 \epsilon_2 \epsilon_3}.
\] (6.3)

During the systematic transmissions, we direct erased symbols to their appropriate queues as outlined in the algorithm of Section 6.3.1. Then, when the \( NT_0 \) transmissions are completed, we start sending linear combinations of the form \( q_i \oplus q_{j,k} \), where \( q_i \in Q_i, q_{j,k} \in Q_{\{j,k\}} \), and \( i, j, k \in U \) are unique.

Let \( T_1 \) be a random variable representing the normalized number of transmissions we can send of the form \( q_i \oplus q_{j,k} \). We are able to do this so long as \( Q_i \) and \( Q_{\{j,k\}} \) are non-empty. Thus, we must bound the cardinality of these queues. For \( Q_{\{j,k\}} \), we have that a symbol is added during the systematic transmissions whenever user \( i \) receives a symbol that is erased at users \( j \) and \( k \). Symbols are no longer added to this queue after the systematic transmissions, and so we have that \( Q^+_{\{j,k\}} \), the
expected maximum normalized cardinality of $Q_{\{j,k\}}$, is given by

$$Q_{\{j,k\}}^+ = T_0(1 - \epsilon_j)\epsilon_k. \quad (6.4)$$

Now, a symbol is removed from $Q_{\{j,k\}}$ whenever user $j$ or user $k$ receives a linear combination involving one of its elements. Thus, $\bar{T}_i$, the expected value of $T_i$, is bounded as

$$\bar{T}_i \leq \frac{Q_{\{j,k\}}^+}{1 - \epsilon_j\epsilon_k}. \quad (6.5)$$

Inequality (6.5) bounds $\bar{T}_i$ relative to the size of $Q_{\{j,k\}}$. We must also bound $\bar{T}_i$ in terms of the cardinality of $Q_i$, since transmissions must stop if $Q_i$ is empty. This bounding is not as straightforward, however, as symbols are added and removed from $Q_i$ as the algorithm progresses. For example, $Q_1$ may be empty at a certain time, but later replenished when we send linear combinations of the form $q_2 + q_{1,3}$. In particular, when this linear combination is received by user 3 and not user 1, we add $q_{1,3}$ to $Q_1$ as user 1 is now the only user in need of it. If $N\bar{T}_2$ linear combinations of this form are sent, we can expect that $N\bar{T}_2\epsilon_i(1 - \epsilon_k)$ symbols are added to $Q_1$. In general, after $N\bar{T}_j$ and $N\bar{T}_k$ linear combinations are sent from $Q_j$, $Q_{\{i,k\}}$ and $Q_k$, $Q_{\{i,j\}}$, respectively, we can expect that $N\bar{T}_j\epsilon_i(1 - \epsilon_k) + N\bar{T}_k\epsilon_i(1 - \epsilon_j)$ symbols are added to $Q_i$. Given that $Q_i$ initially has $N\bar{T}_0\epsilon_i(1 - \epsilon_j)(1 - \epsilon_k)$ symbols after $N\bar{T}_0$ uncoded transmissions, we have then that the expected maximum normalized cardinality of $Q_i$ is given by $Q_i^+(\bar{T}_j, \bar{T}_k)$ where

$$Q_i^+(\bar{T}_j, \bar{T}_k) = \bar{T}_0\epsilon_i(1 - \epsilon_j)(1 - \epsilon_k) + \bar{T}_j\epsilon_i(1 - \epsilon_k) + \bar{T}_k\epsilon_i(1 - \epsilon_j). \quad (6.6)$$

Since a symbol is removed from $Q_i$ each time user $i$ successfully receives a channel symbol, we can also write that

$$\bar{T}_i \leq \frac{Q_i^+(\bar{T}_j, \bar{T}_k)}{1 - \epsilon_i}. \quad (6.7)$$

Notice that by the definition of our stopping condition, $\bar{T}_i$ must actually meet (6.5) or (6.7) with equality, since otherwise $Q_i$ and $Q_{\{j,k\}}$ would still be non-empty, and we would not have reached the stopping condition. So we in fact must have that for all $i \in U$ with $j, k \in U \setminus \{i\}$ s.t. $j \neq k$,

$$\bar{T}_i = \min \left(\frac{Q_i^+(\bar{T}_j, \bar{T}_k)}{1 - \epsilon_i}, \frac{Q_{\{j,k\}}^+}{1 - \epsilon_j\epsilon_k}\right), \quad (6.8)$$

where $\bar{T}_i > 0$. The following theorem proposes that the solution to (6.8) is unique and can be characterized by solving the linear program

$$\max_{\bar{T}_1, \bar{T}_2, \bar{T}_3} \quad \bar{T}_1 + \bar{T}_2 + \bar{T}_3$$

subject to

$$\bar{T}_i \leq \frac{Q_i^+(\bar{T}_j, \bar{T}_k)}{1 - \epsilon_i}, \quad (6.9)$$

$$\bar{T}_i \leq \frac{Q_{\{j,k\}}^+}{1 - \epsilon_j\epsilon_k} \quad \forall i \in U, j, k \in U \setminus \{i\}, j \neq k.$$

**Theorem 3.** Let $\bar{T}_i$ be the expected value of the normalized number of analog transmissions that can be sent of the form $q_i \oplus q_{j,k}$, where $i \in U, j, k \in U \setminus \{i\}, j \neq k$. Then $\bar{T}_i$ is uniquely given by the solution of (6.9).
We proceed toward this end by establishing several lemmas in the appendix. Lemma 21 first establishes that any optimal solution to (6.9) also satisfies (6.8). As $\bar{T}_1 = \bar{T}_2 = \bar{T}_3 = 0$ is clearly a feasible solution of (6.9), we have that the feasible set of (6.9) is non-empty. Therefore, such a solution of (6.9), and by extension (6.8), indeed exists. Conversely, Lemma 22 establishes that any $(\bar{T}_1, \bar{T}_2, \bar{T}_3)$ that satisfies (6.8) is also an optimal solution to (6.9). Finally, Lemma 23 then shows that the optimal solution for (6.9) is unique. We conclude that any $(\bar{T}_1, \bar{T}_2, \bar{T}_3)$ satisfying (6.8) is itself unique and characterized by (6.9).

Now, let $\bar{T}^* = (\bar{T}_1^*, \bar{T}_2^*, \bar{T}_3^*)$ be the optimal solution of (6.9), and let

$$t^* = \bar{T}_0 + \bar{T}_1^* + \bar{T}_2^* + \bar{T}_3^*,$$

(6.10)

where $\bar{T}_0$ is given by (6.3). It is important to determine $t^*$ since it provides a lower bound for the number of instantly decodable, distortion-innovative transmissions possible. From the discussion at the end of Section 5.3, we know that for any latency $w \leq t^*$, we can meet the point-to-point outer bound at $w$. Say that user $i$’s distortion, $d_i$, necessitates a minimum latency of $w_i(d_i)$ where

$$w_i(d_i) = \frac{1 - d_i}{1 - \epsilon_i}.$$  

(6.11)

Let

$$w^-(d_1, d_2, d_3) = \min_{i \in U} w_i(d_i)$$

(6.12)

and

$$w^+(d_1, d_2, d_3) = \max_{i \in U} w_i(d_i).$$

(6.13)

It is clear then that $w^+$ is an outer bound for our problem, and if $w^+ \leq t^*$, we are optimal for all users. Notice however, that if $w^- \leq t^*$, we can also be optimal for all users. This is because if $w^- \leq t^*$, one user can be fully satisfied at latency $w^-$, and so from thereon, we are left with only two users. From Section 5.3, we know that we can remain optimal for the remaining two users, which leads us to the following theorem.

**Theorem 4.** Given $(d_1, d_2, d_3) \in D^3$, let $t^*, w^-(d_1, d_2, d_3)$ and $w^+(d_1, d_2, d_3)$ be given by (6.10), (6.12) and (6.13) respectively. Then if $w^-(d_1, d_2, d_3) \leq t^*$, the latency $w^+(d_1, d_2, d_3)$ is $(d_1, d_2, d_3)$-achievable.

If $w^- > t^*$, it may still be possible to achieve optimality. In particular, this may happen if after $Nt^*$ transmissions have completed, we are left with non-empty queues $Q_i, i \in U$. We can calculate the expected normalized cardinality of $Q_i$ after $t^*$ transmissions, denoted as $|Q_i(t^*)|$, by noting that in $NT_i^*$ transmissions, a symbol is removed from $Q_i$ with probability $(1 - \epsilon_i)$, and so

$$|Q_i(t^*)| = Q_i^+(\bar{T}_i^*, \bar{T}_j^*) - \bar{T}_i^*(1 - \epsilon_i),$$

(6.14)

where $Q_i^+(\bar{T}_i, \bar{T}_j)$ is given by (6.6). If $|Q_i(t^*)|$ is non-zero for all $i \in U$, we can continue sending linear combinations of the form $q_1 \oplus q_2 \oplus q_3$ until a user’s distortion constraint is met or one of the queues $Q_i$ is exhausted. If the latter were to happen, we have that user $i$ has actually reconstructed every source symbol since all queues $Q_U, U \subset U$ s.t. $i \in U$ are empty. Therefore, we are again left with the situation in Section 5.3 involving only two users. We conclude that we can send instantly decodable, distortion-innovative symbols until all users achieve lossless reconstructions.
Theorem 5. Let $Q^- = \min_{t \in T} |Q_i(t^*)|$, where $|Q_i(t^*)|$ is given by (6.14). If $Q^- > 0$, then for any $(d_1, d_2, d_3) \in D^3$, the latency $w^+(d_1, d_2, d_3)$ is $(d_1, d_2, d_3)$-achievable.

6.4.2 Analysis of Chaining Algorithm of Section 6.3.2

In this section, we give a sufficient condition for all users to simultaneously achieve point-to-point optimal performance when the chaining algorithm is invoked. Our analysis is based on a restricted version of the chaining algorithm previously described. Specifically, we assume that we do not opportunistically send linear combinations of the forms specified in Equations (6.2) when the opportunities present themselves (see the description of State 4 in Section 6.3.2). Thus we can argue a fortiori that the actual unrestricted chaining algorithm could only achieve better performance. Before beginning the analysis, we mention that we will rely on many of the results on Markov rewards processes with impulse rewards and absorbing states derived in Section 6.6. We encourage the reader to become familiarized with this section before reading further.

We begin by deriving the transition matrix for the Markov rewards process. Consider Table 6.4. The table shows all outbound transitions given the channel noise realization, however to construct a transition matrix, we must combine all outbound transitions to the same state. For example, rows 1 and 5 both show transitions from State 1 to State 5, and so to get $p_{15}$, the total probability of transitioning from State 1 to 5, we must add the corresponding probabilities under the Pr$(Z(t))$ columns. This is given by

$$p_{15} = \Pr(Z(t) = (0, 0, 0)) + \Pr(Z(t) = (1, 0, 0))$$

$$= (1 - \epsilon_i)(1 - \epsilon_j)(1 - \epsilon_k) + \epsilon_i(1 - \epsilon_j)(1 - \epsilon_k)$$

$$= (1 - \epsilon_j)(1 - \epsilon_k).$$

We continue in this manner to find $p_{ij}$ for all $i, j \in \Omega$ to populate the transition matrix $P$ where the $(i, j)$th entry of $P$ is given by $p_{ij}$.

Our analysis also requires the derivation of several rewards matrices. Consider deriving the rewards matrix $\tilde{\rho}_E$, whose $(i, j)$th element, $\tilde{\rho}_E(i, j)$, gives the expected number of equations (rewards) received by user $i$ for transitioning from State $i$ to State $j$, where $i, j \in \Omega$. Continuing with our previous example, say we would like to derive $\tilde{\rho}_E(1, 5)$. Of the two possible paths for an outbound transition from State 1 to 5, only one, when $Z(t) = (0, 0, 0)$, is associated with a reward in the column of $\rho_E$ in Table 6.4. We therefore calculate $\tilde{\rho}_E(1, 5)$ by weighting the reward with the conditional probability of the transition resulting from the channel noise $Z(t) = (0, 0, 0)$ given that the transition from State 1 to 5 occurred. Therefore,

$$\tilde{\rho}_E(1, 5) = \Pr(Z(t) = (0, 0, 0)|\text{transition from State 1 to 5}) \cdot 1$$

$$= \frac{(1 - \epsilon_i)(1 - \epsilon_j)(1 - \epsilon_k)}{(1 - \epsilon_i)(1 - \epsilon_j)(1 - \epsilon_k) + \epsilon_i(1 - \epsilon_j)(1 - \epsilon_k)}$$

$$= (1 - \epsilon_i).$$

By performing this calculation for all $i, j \in \Omega$, we are able to populate the entire rewards matrix $\tilde{\rho}_E$. Similarly, we can calculate the $|\Omega| \times |\Omega|$ rewards matrices, $\tilde{\rho}_j, \tilde{\rho}_k, \tilde{\rho}_{Q_j}, \tilde{\rho}_{Q_k},$ and $\tilde{\rho}_{Q^*}$ corresponding to each rewards column in Table 6.4.
Given the transition matrix, for each reward matrix, we can use Corollary 6 from Section 6.6.4 to calculate the expected accumulated reward before absorption each time the Markov rewards process is reset and allowed to run until absorption. To find the total expected reward, we must first find a lower bound for $M^*$, the number of times the Markov process is reset before user $j$ or $k$ has met their distortion constraint. That is, $M^*$ is the number of times the Markov rewards process has reached an absorbing state after having been restarted in the initial state. Therefore, $M^*$ also represents the number of chains built by user $i$ before user $j$ or $k$ have satisfied their distortion constraint.

Say that for $r \in \{i, j, k\}$, user $r$ requires $N(1 - \hat{d}_r)$ symbols at the beginning of the chaining algorithm, where $\hat{d}_r \in (0, \epsilon_r)$. Furthermore, suppose that of the two users for whom we are targeting point-to-point optimal performance, user $u$ is not the bottleneck user, i.e.,

$$u = \arg\min_{r \in \{j, k\}} w_r(\hat{d}_r).$$

(6.21)

where $w_r(\hat{d}_r)$ is given by (6.11). For $l = 1, 2, \ldots$, let $\bar{R}_{u,l}$ be the expected accumulated reward for user $u$ during the $l$th time the Markov rewards process has been reset. Then we can define $M^*$ as

$$M^* = \min \left\{ m : \sum_{l=1}^{m} \bar{R}_{u,l} \geq N(1 - \hat{d}_u) \right\},$$

(6.22)

where the $\bar{R}_{u,l}$ are i.i.d. and $\mathbb{E}\bar{R}_{u,l}$ is given by Corollary 6.

We see that $M^*$ is a stopping rule [69]. In order to calculate $\mathbb{E}M^*$, we could use the discrete version of the renewal equation [70, Chapter 2]. However, to find a lower bound for $\mathbb{E}M^*$, we may simply use Wald’s equation [69]. Let $\sigma_m = \sum_{l=1}^{m} \bar{R}_{u,l}$. Then, by Wald’s equation,

$$\mathbb{E}M^* = \frac{\mathbb{E}\sigma_{M^*}}{\mathbb{E}\bar{R}_{u,l}} \geq \frac{N(1 - \hat{d}_u)}{\mathbb{E}\bar{R}_{u,l}},$$

(6.23)

(6.24)

where again, $\mathbb{E}\bar{R}_{u,l}$ is given by Corollary 6. Now, let

$$M^- = \left\lceil \frac{N(1 - \hat{d}_u)}{\mathbb{E}\bar{R}_{u,l}} \right\rceil$$

(6.25)

be the result of applying the floor function to the right-hand-side of (6.24). We have that $M^-$ gives a lower bound for the expected number of times the Markov chain is reset. If user $i$, the user building the chains, can meet their distortion constraint within the $M^-$ times the Markov rewards process is being run, then all users will be point-to-point optimal. This is because user $i$ is able to decode all his required symbols despite the fact that we are targeting optimal performance for the other two users.

Let $R_{E,l}$ be the expected number of symbols in the chain that can be decoded in the $l$th run of the Markov rewards process given that the decoding absorbing state, State 6, was reached after having started in the initial state, State 1. From the discussion leading to the derivation of Corollary 5, we can similarly argue that $\mathbb{E}R_{E,l}$ can be derived by substituting the transition matrix and rewards matrix $\bar{\rho}_E$ into Theorem 7 and simply taking the $(1, 6)$th element from the resultant matrix. This is because the rewards matrix $\bar{\rho}_E$ is used to calculate the expected number of symbols in the chain built by user $i$, however, user $i$ is able to decode these symbols only if State 6, the decoding absorbing state, is reached.
Let $\sigma$ be the expected normalized number of symbols that can be decoded in $M^-$ iterations of the Markov rewards process. By the linearity of the expectation operator we have

$$\sigma = \frac{1}{N} \sum_{l=1}^{M^-} E \bar{R}_{E,l}$$

$$= \frac{M^-}{N} \times E \bar{R}_{E,l}. \quad (6.26)$$

If the right-hand-side of (6.27) is greater than $1 - \hat{d}_i$, the fraction of symbols user $i$ requires, then we are point-to-point optimal for all users. Combining (6.27) and (6.25), we see that this happens when

$$\left\lfloor \frac{N(1 - \hat{d}_u)}{E R_{u,l}} \right\rfloor \geq \frac{N(1 - \hat{d}_i)}{E R_{E,l}}. \quad (6.28)$$

**Theorem 6.** Let $u$ be the user satisfying (6.21). Then $w^+(d_1, d_2, d_3)$ is $(d_1, d_2, d_3)$-achievable if (6.28) is satisfied where $E R_{u,l}$ is calculated from the transition matrix and rewards matrix $\bar{p}_u$ via Corollary 6, and $E \bar{R}_{E,l}$ is calculated by substituting the transition matrix and rewards matrix $\bar{\rho}_E$ into Theorem 7 and taking the $(1, 6)$th element of the resultant matrix.

Finally, we again remark that the analysis we have just described is only a sufficient condition for a restricted version of the chaining algorithm in which we do not opportunistically send linear combinations of the form in (6.2). Therefore, we expect the unrestricted algorithm to perform better.

### 6.4.3 Operational Significance of Theorem 6

Consider a hypothetical situation in which we have queues $Q_{\{1,2\}}, Q_{\{1,3\}}$ and $Q_{\{2,3\}}$. We again consider quadratic distortions in which for $i \in \{2, 3\}$, $d_i = \epsilon_i^2$, and we fix $\epsilon_1 = 0.1, \epsilon_3 = 0.6$ and vary $\epsilon_2 \in (0.2, 0.6)$.

We illustrate the chaining algorithm when user 1 is the user who builds chains and we send linear combinations of the symbols in $Q_{\{1,2\}}$ and $Q_{\{1,3\}}$ as if users 2 and 3 were the only users in the network. In this case, for $i \in \{2, 3\}$, user $i$’s point-to-point optimal latency is given by $w^*_i = (1 - \epsilon^2_i)/(1 - \epsilon_i)$, and since $w^*_i$ is an increasing function of $\epsilon_i \in [0, 1)$, we have that between users 2 and 3, user 2 is not the bottleneck user. That is, in (6.28), user 2 takes the place of user $u$, and since user 1 is building the chains, user 1 takes the place of user $i$.

We rearrange (6.28) of Theorem 6 to find a lower bound for $\hat{d}_i$, and since $\hat{d}_u = \epsilon_u^2$, we plot this lower bound as a function of $\epsilon_u$ in Figure 6.1. From this figure, we can read which values of $\hat{d}_i$ would yield optimal performance for a given value of $\epsilon_u$ by considering all values of $\hat{d}_u$ above the lower bound. Recall that user $i$ is able to decode symbols in the chain only if there have been two consecutive transmissions for which user $i$ is the only user to have received the transmission. We see that the probability of this event increases as $\epsilon_u$ increases in Figure 6.1. Therefore, user $i$ is able to achieve lower distortions as $\epsilon_u$ increases.

Finally, we again mention that Theorem 6 merely gives a conservative sufficient condition for optimality and ignores other network coding opportunities in its analysis. Therefore, it is possible that point-to-point optimal performance can still be met if user $i$ has a distortion constraint below the lower bound of Figure 6.1.
Figure 6.1: The lower bound from Theorem 6 that delineates the boundary of distortion values for which the minmax optimal latency can be achieved.

Figure 6.2: The total normalized number of instantly decodable, distortion-innovative packets that can be sent.

6.5 Simulations

We demonstrate the performance of the algorithm in Section 6.3.1 with simulations. We consider distortions for which $d_i = \epsilon_i^2$ for all $i \in \mathcal{U}$. In Fig. 6.2, we choose a blocklength of $N = 10^7$, fix $\epsilon_1 = 0.3$, $\epsilon_2 = 0.4$ and vary $\epsilon_3$ on the $x$-axis while plotting the total normalized number of instantly decodable, distortion-innovative transmissions sent on the $y$-axis. Alongside this curve, we plot the number of possible innovative transmissions suggested by the solution of (6.9). We observe a close resemblance in this plot and the empirical simulation curve. Finally, we know that although each user may not have their final distortion constraint met after $Nt^*$ transmissions, the provisional distortion they do achieve after $Nt^*$ transmissions is optimal. Say instead, that we were given these provisional distortion values from the onset and asked what latency would be required to achieve these distortions if feedback were not available. The final plot shows this required latency for the segmentation-based coding scheme of [34], which does not incorporate feedback. The gap between these curves is indicative of the benefit that
feedback provides.

In practice, the values of $\epsilon_i$ are much lower than what we have chosen. The values for $\epsilon_3$, e.g., were deliberately chosen to be high ($\epsilon_3 \geq 0.85$) as we have found that for values even as high as $\epsilon_3 = 0.8$, we achieve point-to-point optimality for all users (see Theorems 4 and 5). If we increase $\epsilon_3$ even higher however, we observe a situation where many symbols destined to user 3 are erased, and so when the stopping condition of the algorithm in Section 6.3.1 is reached, we are left with queues $Q_3$, $Q_{\{1,3\}}$ and $Q_{\{2,3\}}$. When this occurs, we have not yet satisfied all users, and so we resort to the coding schemes proposed in Section 6.3.2. Fig. 6.3 shows a plot where each point required the invocation of the queue preprocessing scheme for channel coding in Section 6.3.2. It plots the overall latency required to achieve distortions $d_i = \epsilon_i^2$ for the values of $\epsilon_i$ given earlier. Alternatively, if the chaining algorithm of Section 6.3.2 is used, we find that the latency coincides with the outer bound. Again, the segmentation-based scheme is provided as a benchmark along with the outer bound $w^+(d_1, d_2, d_3)$.

6.6 Markov Rewards Processes with Impulse Rewards and Absorbing States

We study the expected accumulated reward for a discrete-time Markov reward model with absorbing states. The rewards are impulse rewards, where a reward $\rho_{ij}$ is accumulated when transitioning from state $i$ to state $j$. We derive an explicit, single-letter expression for the expected accumulated reward as the number of time steps $n \to \infty$.

6.6.1 Related Work

Markov reward models have been a well-studied area of research for decades [71] particularly in the literature for performance and dependability [72–76]. Variations of the problem formulation have primarily been based on whether the Markov chain is discrete-time or continuous-time, whether there are any absorbing states in the state space, and whether rewards are assigned for the occupancy of a state (rate-based Markov reward models) or for the transition to the state (impulse-based Markov reward models). Within any problem formulation, there have also been variations on the quantity of interest, with
some authors calculating the expected instantaneous reward rate [74], while others find the steady-state expected reward rate [75], the expected accumulated reward [74], the distribution of the accumulated reward until absorption [76], etc. (see [72] for a review of the literature for different reward-based measures). In terms of numerical computation, the topic of model checking for Markov chains has been used to verify whether certain properties hold such as [77], “after a request for service there is at least a 98% probability that the service will be carried out within 2 seconds.” Such a framework has also been extended for Markov reward models [78].

Surprisingly, one formulation that has gone unstudied is that of finding the expected accumulated reward for a discrete-time Markov reward model with absorbing states and impulse rewards. The continuous-time counterpart of this problem has been studied [72]. For a discrete-time model, to the author’s knowledge, the results have either involved a steady-state analysis that excludes absorbing states [71], or a transient analysis for rate-based models that include absorbing states but exclude impulse rewards [71].

Granted, an impulse-based Markov reward model can be translated into a rate-based model by introducing an intermediary state between the transitioning states. Specifically, suppose that in an impulse-based model, state $i$ transitions to state $j$ with probability $p_{ij}$ and a reward of $\rho_{ij}$ is assigned for such a transition. Then in the rate-based counterpart to this model, for every such transition, we create an auxiliary state $k$ such that state $i$ transitions to state $k$ with probability $p_{ij}$ and state $k$ transitions to state $j$ with probability one. In this rate-based model, we assign the same reward $\rho_{ij}$ for occupying state $k$ and solve for the expected accumulated reward for rate-based models as in [71].

While this approach is hypothetically possible, for a state space of size $|\Omega|$, such an approach could potentially add an additional $|\Omega|^2$ intermediate states if the states in the Markov model form a complete digraph. As the solution in [71] involves the inverting of a $|\Omega| \times |\Omega|$ matrix, the computational cost of this approach could be prohibitive. In our work, we instead derive a closed-form solution for the expected accumulated reward without having to resort to introducing intermediary states.

### 6.6.2 Problem Formulation

A discrete-time Markov chain is a sequence of random variables $S_0, S_1, \ldots$, where for every $i = 0, 1, \ldots, S_i$ takes values from the state space $\Omega$, i.e., $S_i \in \Omega$, and the probability of transitioning from state $i$ at time $n-1$ to state $j$ at time $n$ given previous states $S_0, S_1, \ldots, S_{n-1}$ satisfies the Markov property

$$\Pr(S_n = j | S_{n-1} = i, S_{n-2} = i_{n-2}, \ldots, S_1 = i_1) = \Pr(S_n = j | S_{n-1} = i).$$  \hspace{1cm} (6.29)$$

Without loss of generality, we assume the state space, $\Omega$, is indexed by a set of integers so that $\Omega = \{1, 2, \ldots, |\Omega|\}$.

We study time-homogenous Markov chains where the transition probabilities do not depend on $n$. Specifically, at any time $n = 0, 1, \ldots$, the probability of transitioning from state $i$ to state $j$ does not depend on $n$, i.e., $\Pr(S_n = j | S_{n-1} = i) = p_{ij}$. At any time $n$, the transitions between states can therefore be described by a $|\Omega| \times |\Omega|$ transition matrix $P$ whose $(i, j)$th entry is given by $p_{ij}$ for $i, j \in \Omega$.

In addition to being time-homogenous, the Markov chains we study are also absorbing.

**Definition 21.** A state $i \in \Omega$ is said to be absorbing if for all $j \in \Omega \setminus \{i\}$, $p_{ij} = 0$ and $p_{ii} = 1$.

**Definition 22.** A state is said to be transient if it is not an absorbing state.
Definition 23. A discrete-time Markov chain is said to be an absorbing Markov chain if it has at least one absorbing state, and it is possible to reach an absorbing state from any transient state within a finite number of transitions.

A Markov reward process is a Markov chain that incorporates rewards that are accumulated during the evolution of the Markov chain. The reward process we consider is additionally characterized by an impulse-reward matrix.

Definition 24. The impulse-reward matrix, $\Theta$, is a $|\Omega| \times |\Omega|$ matrix whose $(i, j)$th entry, $\rho_{ij}$, represents the reward accumulated when transitioning from state $i$ to state $j$.

For any realization of a sequence of $n+1$ states from time step zero to time step $n$, i.e., given $S_0 = i_0$, $S_1 = i_1, \ldots, S_n = i_n$, we define the accumulated reward $R_n(S_0 = i_0, S_1 = i_1, \ldots, S_n = i_n)$ as

$$R_n(S_0 = i_0, S_1 = i_1, \ldots, S_n = i_n) = \sum_{k=1}^{n} \rho_{i_k-1i_k}.$$

Definition 25. Let $R_n$ be a random variable representing the accumulated reward at time step $n$ when the state sequence $S_0, S_1, \ldots, S_n$ is taken to be random.

Definition 26. Let $\bar{R}_n(i, j)$ be the expected value of $R_n$ given initial state $S_0 = i$ and final state $S_n = j$, i.e., $\bar{R}_n(i, j) = E(R_n|S_0 = i, S_n = j)$.

From Definition 26, the expression for $\bar{R}_n(i, j)$ should be clear when $n = 1$.

Corollary 2. $\bar{R}_1(i, j) = \rho_{ij}$

Definition 27. Let $\tilde{R}_n$ be the $|\Omega| \times |\Omega|$ matrix whose $(i, j)$th entry is given by $\bar{R}_n(i, j)$.

Definition 28. Let $\hat{R}_n(i, j)$ be a scaled version of $\tilde{R}_n(i, j)$ where $\hat{R}_n(i, j) = \tilde{R}_n(i, j) \times \Pr(S_n = j|S_0 = i)$.

Definition 29. Let $\hat{R}_n$ be the $|\Omega| \times |\Omega|$ matrix whose $(i, j)$th entry is given by $\hat{R}_n(i, j)$.

We are often also interested in the expected accumulated reward at time $n$ given only initial state $S_0 = i$ irrespective of the state at time $n$.

Definition 30. Let $\bar{R}_n(i)$ be the expected accumulated reward at time $n$, given initial state $S_0 = i$.

As we will see in the next section, the probability that the Markov reward process eventually reaches an absorbing state is unity, and so the long-term accumulated reward, also known as the reward until absorption, is of interest.

Definition 31. Let $\bar{R}_\infty(i, j) = \lim_{n \to \infty} \bar{R}_n(i, j)$ be the long-term value of $\bar{R}_n(i, j)$.

Definition 32. Let $\tilde{R}_\infty$ be the $|\Omega| \times |\Omega|$ matrix whose $(i, j)$th entry is given by $\tilde{R}_\infty(i, j)$.

Definition 33. Let $\hat{R}_\infty(i) = \lim_{n \to \infty} \hat{R}_n(i)$ be the long-term value of $\hat{R}_n(i)$.

In practice, $\hat{R}_\infty(i)$ is often the most relevant quantity of interest, since we start the Markov reward process in an initial state and wish to know the accumulated reward before absorption. We therefore define our problem as that of finding an expression for $\hat{R}_\infty(i)$. 

We do this by first spending the majority of our efforts in Section 6.6.4 to derive an expression for the scaled accumulated reward variables, which culminates in the derivation of the transient scaled accumulated reward, $\hat{R}_n$, in Lemma 18 and the long-term scaled accumulated reward, $\hat{R}_\infty$, in Theorem 7. In Section 6.6.4, we then show how to use these expressions to derive expressions for the unscaled accumulated reward variables. These variables are conditioned on initial state $S_0 = i$, and consist of the transient accumulated reward, $\bar{R}_n(i)$, in Theorem 8, and the accumulated reward before absorption, $\bar{R}_\infty(i)$, in Corollary 5. Finally, if a prior distribution over the initial states is known, we show how to calculate the unconditional transient accumulated reward, $\bar{R}_n$, in Theorem 9 and the unconditional accumulated reward before absorption, $\bar{R}_\infty$, in Corollary 6.

6.6.3 Background for Absorbing Markov Chains

A discrete-time absorbing Markov chain has a state space $\Omega$ that can be partitioned into a set of absorbing states, $\Omega_A$, and a set of transient states, $\Omega_T$, such that $\Omega = \Omega_A \cup \Omega_T$. Recall our assumption that the state space is indexed by the integers $\{1, 2, \ldots, |\Omega|\}$. We further assume that transient states have lower index values than absorbing states. If this is the case, we can write out the transition matrix in its canonical form

$$P = \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix}, \quad (6.31)$$

where

1. $Q$ is a $|\Omega_T| \times |\Omega_T|$ matrix whose elements represent the probability of transitioning from one transient state to another transient state
2. $R$ is a $|\Omega_T| \times |\Omega_A|$ matrix whose elements represent the probability of transitioning from a transient state to an absorbing state
3. The matrix $0$ is the $|\Omega_A| \times |\Omega_T|$ zero matrix whose elements represent the impossibility of transitioning from an absorbing state to a transient state
4. $I$ is the $|\Omega_A| \times |\Omega_A|$ identity matrix whose elements represent the probability of transitioning from one absorbing state to another absorbing state.

Remark 5. Given the assumption that transient states have lower index values than absorbing states, we may also assume without loss of generality that the impulse-reward matrix, $\Theta$, has the form

$$\Theta = \begin{bmatrix} \Theta_1 & \Theta_2 \\ 0 & 0 \end{bmatrix}, \quad (6.32)$$

where

1. $\Theta_1$ is a $|\Omega_T| \times |\Omega_T|$ matrix whose elements represent the reward accumulated for transitioning from one transient state to another transient state
2. $\Theta_2$ is a $|\Omega_T| \times |\Omega_A|$ matrix whose elements represent the reward accumulated for transitioning from a transient state to an absorbing state
3. the zero matrices have the appropriate dimensions for $\Theta$ to be a $|\Omega| \times |\Omega|$ matrix.
Definition 34. Let $H$ be the Hadamard (element-wise) product of the reward matrix $\Theta$ and transition matrix $P$, i.e., $H = \Theta \odot P$ so that $H(i,j) = \rho_{ij}p_{ij}$

Remark 6. Similar to Remark 5, we may assume without loss of generality that $H$ has the form

\[ H = \begin{bmatrix} H_1 & H_2 \\ 0 & 0 \end{bmatrix}. \] (6.33)

From Definition 28, and Corollary 2, the expression for $\hat{R}_n(i,j)$ should be clear when $n = 1$.

Corollary 3. $\hat{R}_1(i,j) = H(i,j)$

At time step $n$, the probability of being in state $j$ given initial state $i$ is given by the $(i,j)$th entry of $P^n$, which is given by the following lemma.

Lemma 9. For $n = 1, 2, \ldots$, the transition matrix taken to the $n$th power is given by

\[ P^n = \begin{bmatrix} Q^n & \sum_{i=0}^{n-1} Q^i R \\ 0 & I \end{bmatrix}. \] (6.34)

Proof. We proceed by induction.

Base case: It is readily verified by substituting $n = 1$ into (6.34) that we may recover the canonical form of the transition matrix in (6.31).

Inductive Hypothesis: We assume that for $n - 1 \geq 1$,

\[ P^{n-1} = \begin{bmatrix} Q^{n-1} & \sum_{i=0}^{n-2} Q^i R \\ 0 & I \end{bmatrix}. \] (6.35)

Induction Step: We have that

\[ P^n = P^{n-1} P \]

\[ \overset{(a)}{=} \begin{bmatrix} Q^{n-1} & \sum_{i=0}^{n-2} Q^i R \\ 0 & I \end{bmatrix} \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix}, \] (6.37)

where

(a) follows from (6.31) and the inductive hypothesis.

After performing the block matrix multiplication in (6.37), it is straightforward to see that the right-hand sides of (6.34) and (6.37) are equal.

Lemma 10 ([79, Theorem 11.3]). In an absorbing Markov chain, the probability that the process will be absorbed is 1 (i.e., $Q^n \to 0$ as $n \to \infty$).

Lemma 11 ([79, Theorem 11.4]). For an absorbing Markov chain, the matrix $I - Q$ has an inverse, $N$, termed the fundamental matrix, and $N = I + Q + Q^2 + \ldots$. 
Lemma 12. The steady state probability $P^\infty = \lim_{n \to \infty} P^n$ is given by

\[ P^\infty = \begin{bmatrix} 0 & NR \\ 0 & I \end{bmatrix}. \] (6.38)

Proof. This follows from substituting the results of Lemmas 10 and 11 into the expression for $P^n$ in 6.34.

Definition 35 ([80, Definition 5.6.8]). The spectral radius $\rho(A)$ of a matrix $A \in \mathbb{R}^{n \times n}$ is

\[ \rho(A) \triangleq \max\{ |\lambda| : \lambda \text{ is an eigenvalue of } A \}. \] (6.39)

The spectral radius is itself not a matrix norm, however the following corollary states that there exists a norm that is arbitrarily close to the spectral radius.

Lemma 13 ([80, Lemma 5.6.10]). Let $A \in \mathbb{R}^{n \times n}$ and $\epsilon > 0$ be given. There is at least one matrix norm $\| \cdot \|$ such that $\rho(A) \leq \| A \| \leq \rho(A) + \epsilon$.

Lemma 14 ([80, Lemma 5.6.12]). Let $A \in \mathbb{R}^{n \times n}$. Then $\lim_{n \to \infty} A^n = 0$ if and only if $\rho(A) < 1$.

By Lemmas 10 and 14, we have that $\rho(Q) < 1$. By appropriately defining $\epsilon$ in Lemma 13, it then follows that there is a matrix norm for which $\| Q \| < 1$.

Corollary 4. For an absorbing markov chain, there exists a matrix norm for which $\| Q \| < 1$.

6.6.4 Expected Rewards

Scaled Rewards

We first derive Lemma 15 to express $\bar{R}_n(i,j)$ in terms of the elements in both the reward matrix and the transition matrix. We then use this lemma to derive a recurrence relation for the scaled transient accumulated reward, $\hat{R}_n$, in Lemma 17. We use the recurrence relation to derive an actual expression for the transient scaled reward in Lemma 18. Finally, we use properties of absorbing Markov chains to derive a single letter expression for the long-term scaled reward, $\hat{R}_\infty$, in Theorem 7.

Lemma 15. For $n = 2, 3, \ldots$

\[ \bar{R}_n(i,j) = \sum_{k_1=1}^{[\Omega]} \sum_{k_2=1}^{[\Omega]} \cdots \sum_{k_{n-1}=1}^{[\Omega]} \left( \rho_{k_1} + \rho_{k_1 k_2} + \cdots + \rho_{k_{n-1}} \right) \times \frac{p_{i k_1} p_{k_1 k_2} \cdots p_{k_{n-1} j}}{\Pr(S_n = j | S_0 = i)} \] (6.40)
Proof. We calculate

\[ \hat{R}_n(i,j) \]

\[ \begin{align*}
&= \sum_{k_1=1}^{[\Omega]} \sum_{k_2=1}^{[\Omega]} \cdots \sum_{k_{n-1}=1}^{[\Omega]} \{ \mathbb{E}(R_n|S_0 = i, S_1 = k_1, S_2 = k_2, \ldots, S_{n-1} = k_{n-1}, S_n = j) \\
&\quad \times \Pr(S_1 = k_1, S_2 = k_2, \ldots, S_{n-1} = k_{n-1}|S_0 = i, S_n = j) \} \\
&\overset{(b)}{=} \sum_{k_1=1}^{[\Omega]} \sum_{k_2=1}^{[\Omega]} \cdots \sum_{k_{n-1}=1}^{[\Omega]} \{ \mathbb{E}(R_n|S_0 = i, S_1 = k_1, S_2 = k_2, \ldots, S_{n-1} = k_{n-1}, S_n = j) \\
&\quad \times \frac{\Pr(S_1 = k_1, S_2 = k_2, \ldots, S_{n-1} = k_{n-1}, S_n = j|S_0 = i)}{\Pr(S_n = j|S_0 = i)} \} \tag{6.42} \\
&\overset{(c)}{=} \sum_{k_1=1}^{[\Omega]} \sum_{k_2=1}^{[\Omega]} \cdots \sum_{k_{n-1}=1}^{[\Omega]} (\rho_{ik_1} + \rho_{k_1,k_2} + \ldots + \rho_{k_{n-1}j}) \times \frac{\Pr(S_1 = k_1, S_2 = k_2, \ldots, S_{n-1} = k_{n-1}, S_n = j|S_0 = i)}{\Pr(S_n = j|S_0 = i)} \tag{6.44} \\
&\overset{(d)}{=} \sum_{k_1=1}^{[\Omega]} \sum_{k_2=1}^{[\Omega]} \cdots \sum_{k_{n-1}=1}^{[\Omega]} (\rho_{ik_1} + \rho_{k_1,k_2} + \ldots + \rho_{k_{n-1}j}) \times \frac{\rho_{ik_1} \rho_{k_1,k_2} \cdots \rho_{k_{n-1}j}}{\Pr(S_n = j|S_0 = i)} \tag{6.45}
\end{align*} \]

where

(a) follows from Definition 26 and the law of total expectation

(b) follows from Bayes’ Theorem

(c) follows from the fact that we have conditioned on each state from time \( t = 0, 1, 2, \ldots, n \), and so the additive rewards for each transition is known (see (6.30))

(d) follows from the Markov property

\[ \square \]

Lemma 16.

\[ \hat{R}_2(i,j) = HP + PH \] \tag{6.46}

Proof. We use Definition 28 and substitute \( n = 2 \) into Lemma 15 to get

\[ \begin{align*}
\hat{R}_2(i,j) &= \sum_{k_1=1}^{[\Omega]} (\rho_{ik_1} + \rho_{k_1,j}) p_{ik_1} p_{k_1,j} \tag{6.47} \\
&= \sum_{k_1=1}^{[\Omega]} (\rho_{ik_1} p_{k_1}) p_{k_1,j} + p_{ik_1} (\rho_{k_1,j} p_{k_1,j}) \tag{6.48} \\
&\overset{(a)}{=} \sum_{k_1=1}^{[\Omega]} H(i,k_1) p_{k_1,j} + p_{ik_1} H(k_1,j) \tag{6.49} \\
&\overset{(b)}{=} [HP]_{i,j} + [PH]_{i,j} \tag{6.50}
\end{align*} \]

where
(a) follows from Definition 34

(b) follows from the definition of matrix multiplication

We now use Lemma 15 to derive a recurrence relation for \( \hat{R}_n(i,j) \).

**Lemma 17.** For \( n = 2, 3, \ldots \)

\[
\hat{R}_n = \hat{R}_{n-1}P + P^{n-1}H
\]  

(6.51)

**Proof.** We first prove the lemma for \( n = 3, 4, \ldots \)

\[
\hat{R}_n(i,j) \equiv \sum_{k_{n-1}=1}^{\Omega} \left( \sum_{k_{1}=1}^{\Omega} \sum_{k_{2}=1}^{\Omega} \cdots \sum_{k_{n-2}=1}^{\Omega} \left( \rho_k + \rho_{k_1}k_2 + \cdots + \rho_{k_{n-2}}k_{n-1} \right) \times p_{ik_1}p_{k_1k_2} \cdots p_{k_{n-2}k_{n-1}} \right) \times p_{k_{n-1}j} + \sum_{k_{1}=1}^{\Omega} \sum_{k_{2}=1}^{\Omega} \cdots \sum_{k_{n-2}=1}^{\Omega} \rho_{k_{n-1}j} \times p_{ik_1}p_{k_1k_2} \cdots p_{k_{n-1}j} \right) \} (6.52)
\]

\[
\hat{R}_n(i,j) \equiv \sum_{k_{n-1}=1}^{\Omega} \left( \hat{R}_{n-1}(i,k_{n-1}) \right) p_{k_{n-1}j} + \sum_{k_{1}=1}^{\Omega} \sum_{k_{2}=1}^{\Omega} \cdots \sum_{k_{n-2}=1}^{\Omega} \rho_{k_{n-1}j} \times p_{ik_1}p_{k_1k_2} \cdots p_{k_{n-1}j} + \sum_{k_{n-1}=1}^{\Omega} \hat{R}_{n-1}(i,k_{n-1}) p_{k_{n-1}j} + \rho_{k_{n-1}j} \times p_{ik_1}p_{k_1k_2} \cdots p_{k_{n-1}j} \} (6.53)
\]

\[
\hat{R}_n(i,j) \equiv \sum_{k_{n-1}=1}^{\Omega} \hat{R}_{n-1}(i,k_{n-1}) p_{k_{n-1}j} + \rho_{k_{n-1}j} \sum_{k_{1}=1}^{\Omega} \sum_{k_{2}=1}^{\Omega} \cdots \sum_{k_{n-2}=1}^{\Omega} p_{ik_1}p_{k_1k_2} \cdots p_{k_{n-1}j} \} (6.54)
\]

\[
\hat{R}_n(i,j) \equiv \sum_{k_{n-1}=1}^{\Omega} \hat{R}_{n-1}(i,k_{n-1}) p_{k_{n-1}j} + \rho_{k_{n-1}j} \sum_{k_{1}=1}^{\Omega} \sum_{k_{2}=1}^{\Omega} \cdots \sum_{k_{n-2}=1}^{\Omega} p_{ik_1}p_{k_1k_2} \cdots p_{k_{n-1}j} \} (6.55)
\]

\[
\hat{R}_n(i,j) \equiv \sum_{k_{n-1}=1}^{\Omega} \hat{R}_{n-1}(i,k_{n-1}) p_{k_{n-1}j} + \rho_{k_{n-1}j} \times p_{ik_1}p_{k_1k_2} \cdots p_{k_{n-1}j} P^{n-1}(i,k_{n-1}) \} (6.56)
\]

\[
\hat{R}_n(i,j) \equiv \sum_{k_{n-1}=1}^{\Omega} \hat{R}_{n-1}(i,k_{n-1}) p_{k_{n-1}j} + \rho_{k_{n-1}j} \times p_{ik_1}p_{k_1k_2} \cdots p_{k_{n-1}j} P^{n-1}(i,k_{n-1}) \} (6.57)
\]

\[
\hat{R}_n(i,j) = |\hat{R}_{n-1}P|_{i,j} + |P^{n-1}H|_{i,j} (6.58)
\]

where

(a) follows from Definition 28 and rearranging Lemma 15

(b) follows from Definition 28 and the application of Lemma 15 for \( \hat{R}_{n-1}(i,k_{n-1}) \)

(c) follows from Lemma 26 in Appendix C.2, where we have used the fact that \( n \geq 3 \)

(d) follows from Definition 34

(e) follows from the definition of matrix multiplication

We mention that although we derived the lemma assuming \( n \in \{3, 4, \ldots \} \), the lemma also holds if \( n = 2 \). We can see this by using Corollary 3 to compare the right-hand-sides of (6.46) and (6.51) when \( n = 2 \).
Lemma 18. For \( n = 1, 2, \ldots \)

\[
\hat{R}_n = \begin{bmatrix} A_n & B_n \\ 0 & 0 \end{bmatrix},
\]  
(6.59)

where

\[
A_n = \sum_{i=0}^{n-1} Q^i H_1 Q^{n-i-1},
\]  
(6.60)

\[
B_n = N(I - Q^n)H_2 + N(I - Q^{n-1})H_1 NR - \sum_{i=0}^{n-2} Q^i H_1 NQ^{n-i-1} R.
\]  
(6.61)

Proof. We proceed by induction.

**Base case:** We use Corollary 3 and Remark 6 to verify (6.59) for the base case after substituting \( n = 1 \) into (6.60) and (6.61) to get that

\[
A_1 = H_1,
\]  
(6.62)

\[
B_1 = N(I - Q)H_2
\]  
(6.63)

\[
\overset{(a)}{=} H_2,
\]  
(6.64)

where

\( (a) \) follows from the Definition of the fundamental matrix in Lemma 11.

**Inductive Hypothesis:** We assume that for \( n - 1 \geq 1 \),

\[
A_{n-1} = \sum_{i=0}^{n-2} Q^i H_1 Q^{n-i-2};
\]  
(6.65)

\[
B_{n-1} = N(I - Q^{n-1})H_2 + N(I - Q^{n-2})H_1 NR - \sum_{i=0}^{n-3} Q^i H_1 NQ^{n-i-2} R.
\]  
(6.66)

**Induction Step:** We begin with the recurrence relation in Lemma 17 from which we get

\[
\hat{R}_n = \hat{R}_{n-1}P + P^{n-1}H
\]  
(6.67)

\[
\overset{(a)}{=} \begin{bmatrix} A_{n-1} & B_{n-1} \\ 0 & 0 \end{bmatrix} P + P^{n-1}H
\]  
(6.68)

\[
\overset{(b)}{=} \begin{bmatrix} A_{n-1} & B_{n-1} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix} + \begin{bmatrix} Q^{n-1} & \sum_{i=0}^{n-2} Q^i R \\ 0 & I \end{bmatrix} \begin{bmatrix} H_1 & H_2 \end{bmatrix}
\]  
(6.69)

\[
\overset{(c)}{=} \begin{bmatrix} A_{n-1} & B_{n-1} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q & R \\ 0 & I \end{bmatrix} + \begin{bmatrix} Q^{n-1} & \sum_{i=0}^{n-2} Q^i R \\ 0 & I \end{bmatrix} \begin{bmatrix} H_1 & H_2 \end{bmatrix}
\]  
(6.70)

where

\( (a) \) follows from the inductive hypothesis

\( (b) \) follows from Lemma 9
(c) follows from Remark 6.

We first consider $A_n$. We perform the block matrix multiplication in (6.70) to get that

$$A_n = A_{n-1}Q + Q^{n-1}H_1$$ \hspace{1cm} (6.71)

$$= \left( \sum_{i=0}^{n-2} Q^i H_1 Q^{n-i-2} \right) Q + Q^{n-1}H_1$$ \hspace{1cm} (6.72)

$$= \sum_{i=0}^{n-2} Q^i H_1 Q^{n-i-1} + Q^{n-1}H_1$$ \hspace{1cm} (6.73)

$$= \sum_{i=0}^{n-1} Q^i H_1 Q^{n-i-1}$$ \hspace{1cm} (6.74)

where

(a) follows from the inductive hypothesis.

We conclude that (6.60) holds after comparing with (6.74) in the induction step. Next, for $B_n$, we again perform the block matrix multiplication in (6.70) to get that

$$B_n = A_{n-1}R + B_{n-1} + Q^{n-1}H_2$$ \hspace{1cm} (6.75)

$$= \left( \sum_{i=0}^{n-2} Q^i H_1 N(1-Q)Q^{n-i-2} R \right) R + \left( N(I-Q^{n-1})H_2 + N(I-Q^{n-2})H_1 NR \right.$$

$$\hspace{1cm} \left. - \sum_{i=0}^{n-3} Q^i H_1 N Q^{n-i-2} R \right) + Q^{n-1}H_2$$ \hspace{1cm} (6.76)

$$= \left( \sum_{i=0}^{n-2} Q^i H_1 N(1-Q)Q^{n-i-2} R - \sum_{i=0}^{n-3} Q^i H_1 N Q^{n-i-2} R \right)$$

$$\hspace{1cm} + N(I-Q^{n-1})H_2 + N(I-Q^{n-2})H_1 NR + Q^{n-1}H_2$$ \hspace{1cm} (6.77)

$$= \left( \sum_{i=0}^{n-2} Q^i H_1 N Q^{n-i-2} R - Q^i H_1 N Q^{n-i-1} R \right) - \sum_{i=0}^{n-3} Q^i H_1 N Q^{n-i-2} R$$

$$\hspace{1cm} + N(I-Q^{n-1})H_2 + N(I-Q^{n-2})H_1 NR + Q^{n-1}H_2$$ \hspace{1cm} (6.78)

$$= \left( Q^{n-2} H_1 NR - \sum_{i=0}^{n-2} Q^i H_1 N Q^{n-i-1} R \right)$$

$$\hspace{1cm} + N(I-Q^{n-1})H_2 + N(I-Q^{n-2})H_1 NR + Q^{n-1}H_2$$ \hspace{1cm} (6.79)

$$= N(I-Q^{n-1} + N^{-1} Q^{n-1})H_2 + N(I-Q^{n-2} + N^{-1} Q^{n-2})H_1 NR$$

$$\hspace{1cm} - \sum_{i=0}^{n-2} Q^i H_1 N Q^{n-i-1} R$$ \hspace{1cm} (6.80)

$$= N(I-Q^n)H_2 + N(I-Q^{n-1})H_1 NR - \sum_{i=0}^{n-2} Q^i H_1 N Q^{n-i-1} R$$ \hspace{1cm} (6.81)

where

(a) follows from the inductive hypothesis
(b) and (c) follow from the definition of the fundamental matrix, i.e., \( N^{-1} = (I - Q) \) in Lemma 11.

We conclude the proof after comparing (6.61) with (6.81) in the induction step.

\[ \hat{R}_\infty = \begin{bmatrix} 0 & B_\infty \\ 0 & 0 \end{bmatrix} \]

(6.82)

where

\[ B_\infty = N(H_2 + H_1R). \]

(6.83)

**Theorem 7.** Let \( A_\infty = \lim_{n \to \infty} A_n \), and \( B_\infty = \lim_{n \to \infty} B_n \). Then \( A_\infty = 0 \) and

\[ \hat{R}_\infty = \begin{bmatrix} 0 & B_\infty \\ 0 & 0 \end{bmatrix} \]

where

\[ B_\infty = N(H_2 + H_1R). \]

Proof. We begin by writing

\[ B_\infty \overset{(a)}{=} \lim_{n \to \infty} \left( N(I - Q^n)H_2 + N(I - Q^{n-1})H_1NR - \sum_{i=0}^{n-2} Q^iH_1NQ^{n-i-1}R \right) \]

(6.84)

\[ \overset{(b)}{=} NH_2 + NH_1R - \lim_{n \to \infty} \sum_{i=0}^{n-2} Q^iH_1NQ^{n-i-1}R \]

(6.85)

where

(a) follows from Lemma 18

(b) follows from Lemma 10

In order to prove (6.83), we must now show the last term in (6.85) converges to the zero matrix. We use Corollary 4 and Lemma 27 in Appendix C.2 to conclude that this is indeed the case.

For \( A_\infty \), we have that

\[ A_\infty = \lim_{n \to \infty} \sum_{i=0}^{n-1} Q^iH_1Q^{n-i-1} \]

(6.86)

\[ = \lim_{n \to \infty} \left( Q^{n-1}H_1 + \sum_{i=0}^{n-2} Q^iH_1Q^{n-i-1} \right) \]

(6.87)

We conclude from Lemma 10 that the first term in (6.87) converges to the zero matrix. Finally we again use Corollary 4 and Lemma 27 in Appendix C.2 to conclude that the second term in (6.87) also converges to the zero matrix so that \( A_\infty = 0 \).

**Unscaled Rewards**

Having found an expression for \( \hat{R}_n \) in the previous section, we now show that given initial state \( S_0 = i \), the expected accumulated reward at time \( n \) is given by the sum over all columns of the \( i \)th row of the matrix \( \hat{R}_n \).
Theorem 8.

\[ \bar{R}_n(i) = \sum_{j=1}^{|\Omega|} \bar{R}_n(i, j) \]  

(6.88)

Proof. By the law of total probability we have that

\[ \bar{R}_n(i) = \sum_{j=1}^{|\Omega|} \bar{R}_n(i, j) \, \Pr(S_n = j | S_0 = i). \]  

(6.89)

We conclude the result in Theorem 8 holds by Definition 28.

\[ \square \]

Corollary 5.

\[ \bar{R}_\infty(i) = \sum_{j=1}^{|\Omega|} \bar{R}_\infty(i, j). \]  

(6.90)

Similarly, given \( \bar{R}_n(i) \) and a prior distribution over initial states, we can use the law of total probability to calculate the unconditional expected value of \( \bar{R}_n \).

Theorem 9. Let \( \bar{R}_n \) be as defined in Definition 25. If a prior distribution \( \Pr(S_0) \) over the initial state \( S_0 \) is known, then

\[ \bar{R}_n = \sum_{i=1}^{|\Omega|} \bar{R}_n(i) \, \Pr(S_0 = i). \]  

(6.91)

Corollary 6.

\[ \bar{R}_\infty = \sum_{i=1}^{|\Omega|} \bar{R}_\infty(i) \, \Pr(S_0 = i). \]  

(6.92)

Finally, it may be of interest to know the expected accumulated reward after absorption given initial state \( i \) and absorbing state \( j \).

Theorem 10. Let \( i \in \Omega_T \) and \( j \in \Omega_A \). Let \( \bar{R}_\infty(i, j) \) represent the expected accumulated reward after absorption given initial state \( i \) and absorbing state \( j \). Then

\[ \bar{R}_\infty(i, j) = \frac{1}{P_\infty(i, j)} \bar{R}_\infty(i, j), \]  

(6.93)

where \( P_\infty(i, j) \) is the \((i,j)\)th entry of \( P_\infty \) given in Lemma 12.
Chapter 7

Conclusions

In this dissertation, we have studied the information-theoretical limits of the erasure source-broadcast problem. In doing so, our goal was to facilitate robust communications and also the rising demand for video content. Our main contributions were as follows.

- For the erasure source-broadcast problem with \( n \) users and no feedback channel available, in Chapter 3, we propose an achievable coding scheme, i.e., an inner bound. Furthermore, we give sufficient conditions for which the successive segmentation-based coding scheme is optimal.

- For the erasure source-broadcast problem with \( n = 2 \) users where each user has a feedback channel available, we propose an optimal coding scheme in Section 5.3.

- For the erasure source-broadcast problem with \( n = 2 \) users where only the stronger user has a feedback channel available, we similarly propose an optimal coding scheme in Section 5.4.

- For the erasure source-broadcast problem with \( n = 3 \) users where each user has a feedback channel available, we propose an achievable coding scheme in Section 6.3. Furthermore, we give sufficient conditions for which the coding scheme is optimal.

- In analyzing the coding scheme of Section 6.3, we propose two new techniques for analyzing queue-based opportunistic network coding algorithms. The first is in deriving a linear program in Section 6.3.1 to solve for the number of instantly-decodable, distortion-innovative symbols that can be sent. The second technique is in using a Markov rewards process with impulse rewards and absorbing states to analyze queue-based algorithms in Section 6.4.2.

- In our analysis of the chaining algorithm of Section 6.4.2, we derive a more general result for explicitly finding the expected accumulated reward for a discrete-time Markov rewards process with impulse rewards and absorbing states in Section 6.6.

- For the erasure source-broadcast problem with \( n = 2 \) users and no feedback channel available, we propose an outer bound (converse) for a class of non-erasure randomized codes.
7.1 Future Work

For future work, there are a variety of extensions and open problems to pursue. In the area of erasure source-broadcast without feedback, in light of the results of the one-sided feedback problem of Section 5.4, we would like to extend the segmentation-based coding scheme of Section 3.3.2 to include repetition coding.

In order to see how repetition coding could be beneficial even when there is no feedback channel available, consider again the case when all users have a quadratic distortion constraint so that for $i \in [n]$, $d_i = \epsilon_i^2$. In this case, we have that $\hat{w} = 2$ is an upper bound for the achievable latency. This upper bound is achieved by simply transmitting each source symbol uncoded over the channel twice. Then $w$, the number of channel symbols sent per source symbol, is clearly given by $w = \hat{w}$. However, note also that each user’s distortion constraint is also satisfied with this uncoded repetition scheme. This is because the probability that each symbol is not received by user $i$ is simply $\epsilon_i^2$. Therefore, an average distortion of $d_i = \epsilon_i^2$ is trivially satisfied by user $i$ for all $i \in [n]$. There are values of $\epsilon_i$, however, in which the segmentation-based coding scheme achieves a latency greater than $\hat{w}$ when $d_i = \epsilon_i^2$.

For the segmentation-based coding scheme, it is our interest to also conduct a thorough analysis of individual latencies achieved by users. We would also like to analyze the segmentation-based scheme for finite block-lengths and extend the scheme for multiple-description-coded Gaussian sources. Finally, the last topic of future work we propose for the segmentation-based coding scheme is to study the latency achieved with a practical implementation of a multiple-description-coded video source being sent over an erasure broadcast channel. As mentioned in Section 1.3, multiple description codes usually sacrifice compression efficiency for flexibility in being able to reconstruct the source in a way such that refinement layers do not have to be received in order. That is, in terms of source coding, a progressive code is typically more efficient. However, if the output of a progressive code is to be sent over an erasure broadcast channel, the requirement that refinement layers must be received in order would force the receiver to decode specific packets, and discard others that are intended for other users and not useful. Thus, the gain in source-coding efficiency in using a progressive code could be offset by the channel coding required. Fundamentally, the use of a progressive code cascaded with a channel code is an architecture based on source-channel separation, and it would be interesting to see how such an architecture performs compared to the joint source-channel coding architecture we propose.

In terms of future work for the erasure source-broadcast problem with feedback, for the case of $n = 3$ users, it would be of interest to study whether the chaining algorithm of Section 6.3.2 combined with the uncoded transmissions of Section 6.3 is optimal. Furthermore, the problem of generalizing the coding schemes to $n$ users is of interest. In particular, although the queue-based coding schemes we discussed could partially be generalized to $n$ users as mentioned in the end of Section 6.3.1, the number of queues involved in such a scheme would exponentially increase with $n$. In such a case, it may become computationally infeasible to find instantly-decodable, distortion-innovative packets to transmit. In addition, it is not clear how the analysis we conducted for determining regions of optimality could be generalized when $n > 3$. Finally, when it is no longer possible to transmit instantly-decodable, distortion-innovative packets, it would also be interesting to study if the chaining algorithm of Section 6.3.2 could be extended when $n > 3$.

An initial crude approach to generalizing the coding scheme for $n$ users could build upon the results of Section 5.4 and 6.3. In Section 6.3, we studied the problem when $n = 3$, and all three users have access to a feedback channel. For $n$ users, we could divide all users into three groups, and a group can
offer an acknowledgement feedback to the transmitter when all users in that group have received a source symbol. Alternatively, we can find the best decision rule for determining what feedback to send from each group and also optimize to determine how to allocate users to groups. Alternatively, a future topic could also be to find a hybrid coding scheme that progressively ignores feedback as \( n \) increases. For such a coding scheme, the results of Section 5.4 could be useful, since in that section, we studied the case when only one user has a feedback channel available. If such a hybrid coding scheme were to be designed, a future topic of study would be to design it in a way in which it reduces to the segmentation-based coding scheme as \( n \to \infty \).

In future work for deriving an outer bound for the erasure source-broadcast problem without feedback, we would like to further study whether the outer bound we derived applies to not only the class of non-erasure-randomized codes. Alternatively, we could study whether an optimal erasure source-broadcast code is necessarily in the class of non-erasure-randomized codes. As mentioned in Section 4.2.2, the argument for our outer bound derives from a similar argument in [61]. In [81], the authors improved upon the outer bound of [61] by leveraging a subset entropy inequality, and so the topic of how this technique could improve the outer bound studied in Chapter 4 is of interest. Lastly, the generalization of the outer bound for more than two users is also an area to pursue.

There is also potential for future work in deriving an outer bound for the source-broadcast problem with feedback. To the author’s best knowledge, there has been little research in this area thus far. An initial approach to this problem would be to follow the line of work in deriving an outer bound for the problem of channel coding over the erasure broadcast channel with feedback (see Section 2.5). For this related problem, an outer bound was first derived for a physically degraded broadcast channel in [39] where it was shown that feedback does not increase the capacity in this case. This outer bound was then leveraged to derive an outer bound for broadcast channels that are not physically degraded. By hypothetically allowing some receivers to coordinate and have access to each other’s channel outputs, the authors of [40] were able to construct a physically degraded channel and therefore derive a genie-aided bound for broadcast channels that are not physically degraded.

The problem we have studied thus far has assumed the model of a discrete memoryless source that is distributed according to a \( \text{Bern}(1/2) \) distribution. However, in practical situations, it is often the case that the source has redundancy [16]. In general, the source sequence could be non-uniformly distributed, have a Markov structure [82], etc. In such a case, the inner bounds we derived would still hold since the fraction of symbols recovered by our coding schemes would remain the same regardless of the source statistics. On the other hand, the outer bounds would, in general, differ. This is because the outer bounds have been derived based on the separation theorem and the rate-distortion function, which would change for different source statistics (see Chapter 2). We also mention that we may similarly study the erasure source-broadcast problem for erasure channels with memory.

Finally, we mention that the analysis conducted in this thesis was primarily concerned with the expected values of variables, e.g., the expected distortions and expected latencies. In contrast, the analysis of the coding schemes described for finite blocklengths is also an area to pursue.
Appendix A

Segmentation-based Coding Appendix

A.1 Proof of Claim 2

By way of contradiction, suppose that the optimal rates do not belong to the set $R = \{1\} \cup \{1-\epsilon_i, i \in [n]\}$. Then in the optimal solution $(K^*, a^*, r^*)$, there exists some $j, l \in [K^*]$, $j \leq l$, and $i' \in \{0\} \cup [n]$, such that $1-\epsilon_{i'} > r^*_j > r^*_{j+1} > \cdots > r^*_l > 1-\epsilon_{i'+1}$, where we have defined $\epsilon_0 = 0$. Let $j' = \min\{j : 1-\epsilon_{i'} \geq r^*_j\}$.

Then, consider $(K', a', r')$, where $K' = K^* - (l - j')$, $a_k' = \begin{cases} a_k^*, & k = 0, 1, \ldots, j' - 1, \\ \sum_{k=j'}^l a_k^*, & k = j', \\ a_{k+1-j'}^*, & k = j' + 1, \ldots, K', \end{cases}$ and $r_k' = \begin{cases} r_k^*, & k = 0, 1, \ldots, j' - 1, \\ 1-\epsilon_{i'}, & k = j', \\ r_{k+1-j'}^*, & k = j' + 1, \ldots, K'. \end{cases}$

It is not hard to verify that $(K', a', r')$ satisfies all the distortion constraints while the latency (3.3) is strictly reduced. This contradicts the optimality assumption.

A.2 Proof of Theorem 1

We first reformulate the optimization problem in (3.5) by introducing a change of variables. If we let $b_i = \sum_{j=0}^i a_j$ for $i = 0, 1, \ldots, n$, (and hence $a_0 = b_0$ and $a_i = b_i - b_{i-1}$ for $i = 1, 2, \ldots, n$), we can rearrange terms so that (3.5) becomes
\[
\begin{aligned}
\min_{b_0, \ldots, b_n} & \quad \frac{b_n}{1 - \epsilon_n} - b_0 \left( \frac{1}{1 - \epsilon_1} - 1 \right) \\
& \quad - \sum_{i=1}^{n-1} b_i \left( \frac{1}{1 - \epsilon_{i+1}} - \frac{1}{1 - \epsilon_i} \right) \\
\text{subject to} & \quad 0 \leq b_0 \leq b_1 \leq \cdots \leq b_n \leq 1 \\
& \quad (1 - \epsilon_{i+1}) b_i + (b_n - b_i) \geq 1 - d_{i+1} \\
& \quad \text{for } i = 0, 1, 2, \ldots, n-1
\end{aligned}
\] (A.1a)

Our problem is therefore reduced to finding the optimal solution for Problem (A.1), and it is not hard to see that this will in turn allow us to construct the optimal solution for Problem (3.5). We proceed along these lines by first giving a lemma that states that in our search for a segmentation-based code that minimizes latency, we do not sacrifice any optimality by restricting our search to those codes whose segments partition the entire source sequence, i.e., those with \( b_n = 1 \).

**Lemma 19.** Let \( b^* = (b_0^*, b_1^*, \ldots, b_n^*) \) be an optimal solution to (A.1) where \( b_n^* < 1 \). Then \( \beta^* = (b_0^* + \Delta, b_1^* + \Delta, \ldots, b_{n-1}^* + \Delta, 1) \) is also an optimal solution where \( \Delta = (1 - b_n^*)/\epsilon_n \).

**Proof.** It is readily verified that in addition to being feasible, \( \beta^* \) also does not change the objective function in comparison to \( b^* \). The verification relies on the fact that \( d_n \leq \epsilon_n \), which is assumed in our setup. \( \square \)

We now use Lemma 19 in order to show that Theorem 1 gives the optimal segmentation-based scheme.

**Theorem 11.** For the optimization problem in (A.1), there is an optimal solution with \( b_n = 1 \) and \( b_i = \min_{j=i+1}^{n} \left\{ \frac{d_j}{\epsilon_j} \right\} \) for \( i = 0, 1, \ldots, n-1 \).

**Proof.** From Lemma 19, it is sufficient to consider segmentation-based codes with \( b_n = 1 \). From the feasibility constraints of (A.1b) and (A.1c) evaluated with \( b_n = 1 \), we have

\[
\begin{aligned}
b_i - 1 \leq \min \left\{ b_i, \frac{d_i}{\epsilon_i} \right\} \quad \text{for } i \in [n],
\end{aligned}
\] (A.2)

Upon inspection of (A.1a), we see that in order to minimize the objective function, we would like to maximize \( b_{i-1} \) for \( i \in [n] \). Consider first, \( b_{n-1} \), which is upper-bounded as \( b_{n-1} \leq d_n/\epsilon_n \). Continuing, we have that \( b_{n-2} \leq \min\{d_{n-1}/\epsilon_{n-1}, d_n/\epsilon_n\} \) and in general

\[
\begin{aligned}
b_i \leq \min_{j=i+1, \ldots, n} \left\{ \frac{d_j}{\epsilon_j} \right\} \quad \text{for } i = 0, 1, \ldots, n-1.
\end{aligned}
\] (A.3)

We can therefore individually maximize each \( b_i \) by choosing equality in (A.3). This completes the claim. \( \square \)

Finally, to complete the justification of Theorem 1, we note that the expression for \( b_i \) in (A.3) is simply an alternative representation for the variables \((a_0, a_1, \ldots, a_n)\) stated in Theorem 1.
Lemma 20. Let \( X_1, X_2, \ldots, X_n \) be random variables. Then

\[
\sum_{i=1}^{n} H(X_i) = H(X_1, X_2, \ldots, X_n) + \sum_{i=1}^{n} I(X_i; X_{i-1}^i) \quad \text{(B.1)}
\]

Proof.

\[
H(X_1, X_2, \ldots, X_n) = \sum_{i=1}^{n} H(X_i|X_{i-1}^i) \quad \text{(B.2)}
\]

\[
= \sum_{i=1}^{n} H(X_i) - I(X_i; X_{i-1}^i) \quad \text{(B.3)}
\]

The lemma follows from rearranging the last line. \( \square \)
Appendix C

Erasure Source-Broadcast with Feedback Appendix

C.1 Proof of Supporting Lemmas of Theorem 3

Lemma 21. Let \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) be the optimal solution of the linear program in (6.9). Then \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) satisfies (6.8).

Proof. To show that the optimal solution of (6.9) must satisfy (6.8), we note that in order to maximize the objective function in (6.9), one of the inequality constraints for \(\overline{T}_i\) must be met with equality. Otherwise, if \(\overline{T}_i, \overline{T}_j, \overline{T}_k\) is a purported optimal solution where \(\overline{T}_i\) does not meet an inequality constraint with equality, we can find some \(\delta > 0\) such that \(\overline{T}_i + \delta, \overline{T}_j, \overline{T}_k\) is also feasible and gives a strictly larger objective function value. \(\square\)

Lemma 22. Let \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) satisfy (6.8), where \(\overline{T}_i > 0\) for all \(i \in \mathcal{U}\). Then \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) is an optimal solution of (6.9).

Proof. We show that \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) is a local maximum of the linear program in (6.9). Consider a neighbouring element in the feasible region of (6.9) of the form \((\overline{T}_1 + \Delta_1, \overline{T}_2 + \Delta_2, \overline{T}_3 + \Delta_3)\). We show that the objective function evaluated at this new element is not greater than the objective function evaluated at \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\). We do this by showing that the linear program

\[
\max_{\Delta_1, \Delta_2, \Delta_3} \Delta_1 + \Delta_2 + \Delta_3
\]

subject to \(\overline{T}_1 + \Delta_i \leq \frac{Q^+_{i} (\overline{T}_j + \Delta_j, \overline{T}_k + \Delta_k)}{1 - \epsilon_i}\),

\(\overline{T}_i + \Delta_i \leq \frac{Q^+_{\{i,j,k\}}}{1 - \epsilon_j \epsilon_k} \quad \forall i \in \mathcal{U}, j, k \in \mathcal{U} \setminus \{i\}, j \neq k.\) (C.1)

has an optimal value no greater than zero.

By assumption, \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) satisfies (6.8), and so for all \(i \in \mathcal{U}\), \(\overline{T}_i\) meets one of the inequality constraints in (6.9) with equality. Let \(\mathcal{T}(\overline{T}_1, \overline{T}_2, \overline{T}_3)\) be the set of \(i\) for which \(\overline{T}_i\) meets the first inequality constraints in (6.9) with equality. Then

\[
\max_{\Delta_1, \Delta_2, \Delta_3} \Delta_1 + \Delta_2 + \Delta_3
\]

subject to \(\overline{T}_i + \Delta_i \leq \frac{Q^+_{i} (\overline{T}_j + \Delta_j, \overline{T}_k + \Delta_k)}{1 - \epsilon_i}\),

\(\overline{T}_i + \Delta_i \leq \frac{Q^+_{\{i,j,k\}}}{1 - \epsilon_j \epsilon_k} \quad \forall i \in \mathcal{T}(\overline{T}_1, \overline{T}_2, \overline{T}_3), j, k \in \mathcal{U} \setminus \{i\}, j \neq k.\) (C.1)

has an optimal value no greater than zero.

By assumption, \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) satisfies (6.8), and so for all \(i \in \mathcal{U}\), \(\overline{T}_i\) meets one of the inequality constraints in (6.9) with equality. Let \(\mathcal{T}(\overline{T}_1, \overline{T}_2, \overline{T}_3)\) be the set of \(i\) for which \(\overline{T}_i\) meets the first inequality constraints in (6.9) with equality.

By assumption, \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) satisfies (6.8), and so for all \(i \in \mathcal{U}\), \(\overline{T}_i\) meets one of the inequality constraints in (6.9) with equality. Let \(\mathcal{T}(\overline{T}_1, \overline{T}_2, \overline{T}_3)\) be the set of \(i\) for which \(\overline{T}_i\) meets the first inequality constraints in (6.9) with equality.

By assumption, \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) satisfies (6.8), and so for all \(i \in \mathcal{U}\), \(\overline{T}_i\) meets one of the inequality constraints in (6.9) with equality. Let \(\mathcal{T}(\overline{T}_1, \overline{T}_2, \overline{T}_3)\) be the set of \(i\) for which \(\overline{T}_i\) meets the first inequality constraints in (6.9) with equality.

By assumption, \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) satisfies (6.8), and so for all \(i \in \mathcal{U}\), \(\overline{T}_i\) meets one of the inequality constraints in (6.9) with equality. Let \(\mathcal{T}(\overline{T}_1, \overline{T}_2, \overline{T}_3)\) be the set of \(i\) for which \(\overline{T}_i\) meets the first inequality constraints in (6.9) with equality.

By assumption, \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) satisfies (6.8), and so for all \(i \in \mathcal{U}\), \(\overline{T}_i\) meets one of the inequality constraints in (6.9) with equality. Let \(\mathcal{T}(\overline{T}_1, \overline{T}_2, \overline{T}_3)\) be the set of \(i\) for which \(\overline{T}_i\) meets the first inequality constraints in (6.9) with equality.

By assumption, \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) satisfies (6.8), and so for all \(i \in \mathcal{U}\), \(\overline{T}_i\) meets one of the inequality constraints in (6.9) with equality. Let \(\mathcal{T}(\overline{T}_1, \overline{T}_2, \overline{T}_3)\) be the set of \(i\) for which \(\overline{T}_i\) meets the first inequality constraints in (6.9) with equality.

By assumption, \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) satisfies (6.8), and so for all \(i \in \mathcal{U}\), \(\overline{T}_i\) meets one of the inequality constraints in (6.9) with equality. Let \(\mathcal{T}(\overline{T}_1, \overline{T}_2, \overline{T}_3)\) be the set of \(i\) for which \(\overline{T}_i\) meets the first inequality constraints in (6.9) with equality.

By assumption, \((\overline{T}_1, \overline{T}_2, \overline{T}_3)\) satisfies (6.8), and so for all \(i \in \mathcal{U}\), \(\overline{T}_i\) meets one of the inequality constraints in (6.9) with equality. Let \(\mathcal{T}(\overline{T}_1, \overline{T}_2, \overline{T}_3)\) be the set of \(i\) for which \(\overline{T}_i\) meets the first inequality constraints in (6.9) with equality.
Appendix C. Erasure Source-Broadcast with Feedback Appendix

constraint with equality, i.e.,

\[
\mathcal{T}(\vec{T}_1, \vec{T}_2, \vec{T}_3) = \left\{ i \in \mathcal{U} \mid \vec{T}_i = \frac{Q^+_i(\vec{T}_j, \vec{T}_k)}{1 - \epsilon_i}, \vec{T}_i > 0, j, k \in \mathcal{U} \setminus \{i\}, j \neq k \right\}.
\]  

(C.2)

When the context is clear, we will at times use \(\mathcal{T}\) to refer to \(\mathcal{T}(\vec{T}_1, \vec{T}_2, \vec{T}_3)\). Furthermore, for \(i \in \mathcal{T}\), we will at times find it convenient to emphasize the linear dependence of \(\vec{T}_i\) on \(\vec{T}_j\) and \(\vec{T}_k\) by writing

\[
Q^+_i(\vec{T}_j, \vec{T}_k) = k_i + a_{ik}\vec{T}_j + a_{ij}\vec{T}_k,
\]  

(C.3)

where

\[
a_{ik} = \frac{\epsilon_i (1 - \epsilon_j)(1 - \epsilon_k)}{(1 - \epsilon_i)} > 0,
\]  

(C.4)

\[
k_i = \frac{T_0\epsilon_i(1 - \epsilon_j)(1 - \epsilon_k)}{1 - \epsilon_i} > 0,
\]  

(C.5)

and \(T_0\) is given by (6.3).

Now, we have that if \(i \notin \mathcal{T}\), then \(\vec{T}_i = Q^+_i(\vec{T}_j) / (1 - \epsilon_j \epsilon_k)\). From the second inequality of (C.1), we have then that for these values of \(i\), \(\Delta_i \leq 0\). Notice however, that an optimal solution of (C.1) must have all of its components non-negative. Otherwise, a larger-valued objective function could be obtained by setting any negative component to zero. We can therefore upper bound the optimal value for the linear program in (C.1) with the negated optimal value for the relaxed linear program

\[
\min_{\Delta_1, \Delta_2, \Delta_3} - (\Delta_1 + \Delta_2 + \Delta_3)
\]  

subject to \(\Delta_i \leq a_{ik}\Delta_j + a_{ij}\Delta_k\)

\[
\forall i \in \mathcal{T}, j, k \in \mathcal{U} \setminus \{i\}, j \neq k
\]

\(\Delta_j = 0, \ \forall j \notin \mathcal{T}\).

(C.6)

where we have used (C.2) and (C.4) to simplify the constraints of (C.1).

We now consider several cases. Notice first that if \(0 \leq |\mathcal{T}| \leq 1\), the constraints of (C.6) clearly show that the optimal value of (C.6) is zero. Alternatively, if \(2 \leq |\mathcal{T}| \leq 3\), we consider the Lagrange dual function, \(g(\lambda)\), given by

\[
g(\lambda) = \begin{cases} 
0 & \text{if } A^T \lambda = -c, \\
-\infty & \text{else},
\end{cases}
\]  

(C.7)

where we have assumed that the objective function and inequality constraints of (C.6) have been written as \(c^T \Delta\) and \(A\Delta \leq 0\) respectively for some \(\Delta\), \(A\) and \(c\) that will be subsequently defined depending on \(|\mathcal{T}|\). For any \(\lambda \geq 0\), \(g(\lambda)\) gives a lower bound on the optimal value of (C.6). If we can therefore find a \(\lambda \geq 0\) such that

\[
A^T \lambda = -c,
\]  

(C.8)

we will have thus shown that the optimal value of (C.1) is no greater than zero, as was our original goal.

1. Consider now, if \(|\mathcal{T}| = 2\). Let \(i, j\) and \(k\) be distinct elements in \(\mathcal{U}\), where we assume without loss
Appendix C. Erasure Source-Broadcast with Feedback Appendix

of generality that \( k \) is the only element not in \( T \). In this case, we have

\[
A = A_2 \triangleq \begin{bmatrix} 1 & -a_{ik} \\ -a_{jk} & 1 \end{bmatrix},
\]

(C.9)

\[
c = c_2 \triangleq \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \quad \Delta = \hat{\Delta} \triangleq \begin{bmatrix} \Delta_i \\ \Delta_j \end{bmatrix},
\]

(C.10)

where we remind the reader that we have assumed that the constraints in (C.6) are written as \( A\Delta \leq 0 \). We explicitly invert the system of equations for \( \lambda \) in (C.8) to get that

\[
\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}^T = -\left(A_2^T\right)^{-1}c_2
\]

where

\[
\left(A_2^T\right)^{-1} = \frac{1}{1-a_{ik}a_{jk}} \begin{bmatrix} 1 & a_{jk} \\ a_{ik} & 1 \end{bmatrix}.
\]

(C.11)

Lemma 24 shows that if \( i \) and \( j \) are in \( T \), the denominator in (C.11) is positive. We use this along with (C.4) and (C.10) to conclude that indeed \( \lambda \succeq 0 \).

2. If \( |T| = 3 \), we have that

\[
A = A_3 \triangleq \begin{bmatrix} 1 & -a_{ik} & -a_{ij} \\ -a_{jk} & 1 & -a_{ji} \\ -a_{ki} & -a_{kj} & 1 \end{bmatrix},
\]

(C.12)

\[
c = c_3 \triangleq \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}, \quad \Delta = \hat{\Delta} \triangleq \begin{bmatrix} \Delta_i \\ \Delta_j \\ \Delta_k \end{bmatrix}.
\]

(C.13)

In Lemma 25, we establish that if the system of linear equations in the definition of \( T \) are satisfied when \( |T| = 3 \), then \( \det(A_3) > 0 \). We can therefore again invert the system of linear equations in (C.8) to get that \( [\lambda_1, \lambda_2, \lambda_3]^T = -(A_3^T)^{-1}c_3 \) where

\[
\left(A_3^T\right)^{-1} = \frac{1}{\det(A_3)} \begin{bmatrix} 1 - a_{ji}a_{ki} & a_{jk} + a_{ji}a_{kj} & a_{kj} + a_{jk}a_{ki} \\ a_{ik} + a_{ij}a_{ki} & 1 - a_{ij}a_{kj} & a_{ki} + a_{ik}a_{kj} \\ a_{ij} + a_{ik}a_{ji} & a_{ji} + a_{ij}a_{jk} & 1 - a_{ik}a_{jk} \end{bmatrix}.
\]

(C.14)

By Lemmas 24 and 25 and Equations (C.4) and (C.13), we again conclude that \( \lambda \succeq 0 \).

\[\square\]

Lemma 23. Let \((\bar{T}_1, \bar{T}_2, \bar{T}_3)\) be the optimal solution of the linear program in (6.9). Then \((\bar{T}_1, \bar{T}_2, \bar{T}_3)\) is unique.

Proof. We show that the solution to (6.9) is unique via Theorem 1 of [83]. Intuitively, its reasoning is that to maximize the objective function of a linear program, we follow the objective function’s gradient vector until we reach the boundary of the feasible region. If this stopping occurs at a face of the region rather than a single point, we have a non-unique solution. In this case, following the gradient vector after it has undergone a small perturbation in its direction will lead us to a different solution on the face of the boundary region. On the other hand, if any small perturbation in the gradient vector’s direction leads us to the same boundary point as the unperturbed gradient vector, we know we have a unique solution. This is stated in the following fact.
Fact 12. A solution \( \tilde{x} \) to the linear programming problem

\[
\max_x \quad c^T x \\
\text{subject to} \quad Ax \leq b
\]  

(C.15)
is unique iff for all \( q \), there exists a \( \delta > 0 \) s.t. \( \tilde{x} \) is still a solution when the objective function is replaced by \((c + \delta q)^T x\).

For our purposes, we have that \( c = [1, 1, 1]^T \) in Fact 12. Furthermore, for any given \( q \), we choose \( \delta \) large enough so that all coefficients of the newly-perturbed objective function remain positive.

Let \( \bar{T}^* = (\bar{T}_1^*, \bar{T}_2^*, \bar{T}_3^*) \) be the optimal solution of (6.9), and let \( T = (T_1, T_2, T_3) \) be another element in the feasible region of (6.9). We will show that \( \bar{T}^* \) remains the optimal solution of (6.9) when the objective function is perturbed in a way that maintains the positivity of its coefficients by showing that

\[
\bar{T}^*_i - T_i \geq 0, \quad \text{for all } i \in U. \tag{C.16}
\]

Assume, by way of contradiction, that (C.16) does not hold, i.e., there exists a non-empty set \( V \subseteq U \) such that for all \( j \in V \), \( \bar{T}^*_j - T_j < 0 \). We will show that the existence of such a set violates the optimality assumption of \( \bar{T}^* \). In particular, we construct a feasible solution of (6.9) with a strictly larger objective function than \( \bar{T}^* \). This new solution retains all \( \bar{T}^*_i \) for \( i \in U \setminus V \) and replaces all \( \bar{T}^*_j \) with \( T_j \) for all \( j \in V \).

For all \( i \in U \setminus V \), we now show the feasibility of \( \bar{T}^*_i \) within the newly-constructed solution by writing

\[
\bar{T}^*_i \leq \frac{Q^+_{\{j,k\}}}{1 - \epsilon_j \epsilon_k}, \tag{C.17}
\]

and

\[
\bar{T}^*_i \overset{(a)}{=} k_i + a_{ik} \bar{T}^*_j + a_{ij} \bar{T}^*_k \tag{C.18}
\]

\[
= k_i + \sum_{u \in U \setminus V, v \in U \setminus \{i,u\}} a_{iu} \bar{T}^*_u + \sum_{u \in U \setminus \{i\}} a_{iu} \bar{T}^*_u \tag{C.19}
\]

\[
\overset{(b)}{=} k_i + \sum_{u \in U \setminus V, v \in U \setminus \{i,u\}} a_{iv} \bar{T}^*_u + \sum_{u \in U \setminus \{i\}} a_{iu} \bar{T}^*_v, \tag{C.20}
\]

where

\( (a) \) and (C.17) follow from (C.3) and the feasibility of \( \bar{T}^* \)

(b) follows from the definition of \( V \).

Hence, (C.20) shows that \( \bar{T}^*_i \) remains feasible within the newly-constructed solution. Similarly, we can show that \( \bar{T}^*_j \) remains feasible within the newly-constructed solution for all \( j \in V \).

\[\square\]

Lemma 24. Let \( i, j \) and \( k \) be distinct elements in \( U \), and let \((T_1, T_2, T_3)\) satisfy (6.8). If \( i, j \in T(\bar{T}_1, \bar{T}_2, \bar{T}_3) \), then \( 1 - a_{ik} a_{jk} > 0 \), where \( T \) is given by (C.2) and \( a_{ik} \) is given by (C.4).
Proof. We eliminate $T_j$ from the two linear equations

$$\bar{T}_i = Q^+_i (\bar{T}_j, \bar{T}_k)/(1-\epsilon_i) \text{ and } \bar{T}_j = Q^+_j (\bar{T}_i, \bar{T}_k)/(1-\epsilon_j)$$

to get that

$$\left( a_{jk} - \frac{1}{a_{ik}} \right) \bar{T}_i + a_{ji} \bar{T}_k + k_j + \frac{1}{a_{ik}} (\bar{T}_k + k_i) = 0. \quad (C.21)$$

We observe that the coefficient of $\bar{T}_i$ in (C.21) must be negative, as all the other terms in (C.21) are strictly positive.

Lemma 25. Let $i, j$ and $k$ be distinct elements in $U$. If $\bar{T} = [\bar{T}_i, \bar{T}_j, \bar{T}_k]^T > 0$ satisfies $A_3 \bar{T} = k$, where $A_3$ is given by (C.12), and $k = [k_i, k_j, k_k]$ is given by (C.5), then $\det(A_3) > 0$.

Proof. We first show that $\det(A_3) \neq 0$ by assuming, by way of contradiction, that $\det(A_3) = 0$. A trivial determinant can result in a system of equations having either no solution, or infinitely many solutions. For our specific situation however, we will show that the system of linear equations $A_3 \bar{T} = k$ is inconsistent if $\det(A_3) = 0$.

For $i \in U$, let $\alpha_i$ be the $i$th row of $A_3$. Now, a trivial determinant implies that for some $\gamma_2$ and $\gamma_3$,

$$\alpha_3 = \gamma_1 \alpha_1 + \gamma_2 \alpha_2. \quad (C.22)$$

Equation (C.22) is an overspecified system of linear equations, since we have three equations in the two unknown variables $\gamma_2$ and $\gamma_3$. This in turn imposes an additional constraint for the elements of $A_3$ if (C.22) is to be true. We assume that these additional constraints hold, and notwithstanding this issue, we solve for $\gamma_2$ and $\gamma_3$ to get that

$$\begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} = (A_2^T)^{-1} \begin{bmatrix} -a_{kj} \\ -a_{ki} \end{bmatrix}, \quad (C.23)$$

where $(A_2^T)^{-1}$ is given by (C.11). Since each element of $(A_2^T)^{-1}$ is positive, we conclude from (C.4) and (C.23) that $\gamma_2$ and $\gamma_3$ are both negative (c.f. Lemma 22 for the case when $|T| = 2$).

We now proceed towards a contradiction by writing

$$k_k = \alpha_3 \bar{T} \quad (C.24)$$

$$= (\gamma_1 \alpha_1 + \gamma_2 \alpha_2) \bar{T} \quad (C.25)$$

$$= \gamma_1 k_i + \gamma_2 k_j. \quad (C.26)$$

The contradiction results from the fact that by (C.5) and (C.23), the left-hand-side of (C.26) is positive, while the right-hand-side of (C.26) is negative.

Having shown that $\det(A_3)$ is non-zero, we now show that it is, furthermore, positive. We do this by first inverting the system of linear equations $A_3 \bar{T} = k$ to get that

$$\bar{T} = A_3^{-1} k, \quad (C.27)$$

where the transpose of $A_3^{-1}$ is given by (C.14). From (C.5), (C.14), (C.27), and Lemma 24 we see that each element of $\bar{T}$ has the same sign, and so if $\bar{T} > 0$, we must have that $\det(A_3) > 0$. 

\qed
C.2 Matrix Properties

Lemma 26. Let $A$ be an $m \times m$ matrix, and let $A^n$ denote the $n$th power of $A$ for some $n \in \{2, 3, \ldots \}$. Then the $(i, j)$th entry of $A^n$, denoted by $A^n(i, j)$, is given by

$$A^n(i, j) = \sum_{k_1=1}^{m} \sum_{k_2=1}^{m} \cdots \sum_{k_{n-1}=1}^{m} A(i, k_1)A(k_1, k_2) \cdots A(k_{n-1}, j) \quad (C.28)$$

Proof. We proceed by induction.

Base case: We substitute $n = 1$ into (C.28) to get that

$$A^2(i, j) = \sum_{k_1=1}^{m} A(i, k_1)A(k_1, j), \quad (C.29)$$

which is the familiar definition for matrix multiplication.

Inductive Hypothesis: We assume that for $n - 1 \geq 1$,

$$A^{n-1}(i, j) = \sum_{k_1=1}^{m} \sum_{k_2=1}^{m} \cdots \sum_{k_{n-2}=1}^{m} A(i, k_1)A(k_1, k_2) \cdots A(k_{n-2}, j). \quad (C.30)$$

Induction Step: We have that

$$A^n(i, j) \overset{(a)}{=} \sum_{k_{n-1}=1}^{m} A^{n-1}(i, k_{n-1})A(k_{n-1}, j) \quad (C.31)$$

$$\overset{(b)}{=} \sum_{k_{n-1}=1}^{m} \left\{ \sum_{k_1=1}^{m} \sum_{k_2=1}^{m} \cdots \sum_{k_{n-2}=1}^{m} A(i, k_1)A(k_1, k_2) \cdots A(k_{n-2}, k_{n-1}) \right\}A(k_{n-1}, j) \quad (C.32)$$

where

(a) follows from the definition of matrix multiplication and the fact that $A^n = A^{n-1}A$

(b) follows from the inductive hypothesis

After rearranging the right-hand-side of (C.32), we conclude that (C.28) indeed holds.

Lemma 27. Let $g : \mathbb{N} \times \mathbb{R}^{m \times m} \times \mathbb{R}^{m \times m} \rightarrow \mathbb{R}^{m \times m}$ be the function given by

$$g(n, A, Q) = \sum_{i=0}^{n-2} Q^i A Q^{n-i-1}. \quad (C.33)$$

where $Q$ is a matrix such that there exists a norm for which $\|Q\| < 1$. Then

$$\lim_{n \rightarrow \infty} g(n, A, Q) = 0. \quad (C.34)$$
Proof. We show that the norm of $g(n, A, Q)$ approaches zero as $n \to \infty$. For any $n$, we have that

$$
\|g(n, A, Q)\| = \left\| \sum_{i=0}^{n-2} Q^i A Q^{n-i-1} \right\| \quad (C.35)
$$

(a) follows from sub-additive property of the matrix norm

(b) follows from sub-multiplicative property of the matrix norm

Finally, we use L’Hospital’s Rule and the assumption that $\|Q\| < 1$ to conclude that the right-hand-side of (C.39) approaches zero as $n \to \infty$. 

$\Box$
Bibliography


