On Playback Delay in Streaming Communication

by

Gauri Joshi

B. Tech., Indian Institute of Technology Bombay (2009)
M. Tech., Indian Institute of Technology Bombay (2010)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2012

Author . . . .
Department of Electrical Engineering and Computer Science
May 21, 2012

Certified by . . . .
Gregory W. Wornell
Professor of Electrical Engineering at MIT
Thesis Supervisor

Certified by . . . .
Yuval Kochman
Senior Lecturer at Hebrew University of Jerusalem, Israel
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Theses

On Playback Delay in Streaming Communication

by

Gauri Joshi

Submitted to the Department of Electrical Engineering and Computer Science
on May 21, 2012, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering

Abstract

In this thesis, we consider the problem of minimizing playback delay in streaming over a packet erasure channel with fixed bandwidth. In recent years, there has been a rapid increase in live streaming applications where packets have to be played back at the receiver in order. With instantaneous feedback, the automatic-repeat-request (ARQ) protocol is delay optimal. However, with no feedback or delayed feedback, there is a trade-off between transmitting new packets and retransmitting old packets, to reduce the playback delay.

Existing erasure codes such as Reed-Solomon codes and fountain codes that operate without feedback have delay proportional to the length of the stream, and hence are not suitable for streaming applications. Other coding schemes specifically designed for delay-constrained packet transmission aim to minimize the decoding delay. However, playback delay is a more natural metric for applications requiring in-order playback at the receiver.

We aim to find good streaming codes that minimize playback delay for such channels with limited or no feedback. We analyze three cases, namely no-feedback, delayed feedback and broadcast with instantaneous feedback. We find that in all cases the playback delay grows logarithmically with the time elapsed since the start of transmission, and we evaluate the growth constant, i.e. the pre-log term, as a function of the transmission bandwidth (relative to the source bandwidth). The main tool used in the analysis of delay in all cases is to model packet decoding in terms of threshold crossing of a random walk.

We can show that the expected playback delay is asymptotically equal to $1/\lambda \log n$ where λ is referred to as the growth constant. For the no-feedback case, the optimal value is $\lambda = D(1/b||\rho)$ where b is the bandwidth in packets per slot and ρ is the success probability of the erasure channel. We prove that the simple coded repetition scheme where the source transmits combinations all packets generated so far in every slot achieves this optimal growth constant.

With instantaneous feedback, the ARQ scheme is optimal and we can determine the exact expression for λ. For the delayed feedback case we propose a greedy coding scheme and use it to determine a lower bound on λ_d as a function of feedback delay d.

We can prove that the growth constant with feedback is strictly better that the one without, but they have the same asymptotic value in the limit of infinite bandwidth.

We further extend the analysis to a broadcast streaming scenario with instantaneous feedback where the source is transmitting a common packet stream to N users over independent erasure channels. We determine how the growth constant λ_N scales with the number of the users N.

It can be shown that greedy coding is optimal for the without feedback and instantaneous feedback cases, however we have not yet proved its optimality for the delayed feedback and broadcast streaming. This is the major part of ongoing research efforts. Other future research directions include extending the results to packet networks and considering more general channel models.

Thesis Supervisor: Gregory W. Wornell
Title: Professor of Electrical Engineering at MIT

Thesis Supervisor: Yuval Kochman
Title: Senior Lecturer at Hebrew University of Jerusalem, Israel

Acknowledgments

First of all, I would like to express my sincere gratitude to my advisors Gregory Wornell and Yuval Kochman for their guidance and support. Greg taught me to be patient and keep the big picture in mind while defining and solving a problem. Yuval has been a great of source of inspiration and encouragement, especially before the ISIT deadline. He taught me to be rigorous in proofs which helped me to find the flaws and correct them.

I thank everyone else in the Signals, Algorithms and Information Laboratory – Da, James, Maryam, Arya, Ligong, Uri, Ying-zong, Qing, Atulya, Tricia and Lucia – for making it such a great place to work. Special thanks to Qing and Atulya for being amazing office-mates. I also thank Shreeshankar Bodas for helpful discussions on this work.

I would like to thank everyone associated with MIT Sangam, the Indian Students Association. It has been an important part of my life at MIT. Also, I am grateful my close friends both in MIT and India for standing by me. In particular, I thank Prof. Abhay Karandikar, my advisor at IIT Bombay who first taught me research.

Most importantly, I thank my parents for their strife to give me the best possible education. They have always been my stronghold of support and encouragement. I dedicate this thesis to my grandfather, Appa whose passion for learning and teaching has inspired me to pursue academia as my career.

This work was supported in part by MIT Lincoln Laboratory, by AFOSR under Grant No. FA9550-11-1-0183, and by Hewlett-Packard Laboratories.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

In recent years there has been a widespread proliferation of audio/video streaming applications on both wired and wireless media. Unlike transmission of a file of large size where the only the delay until completion of the data transfer matters, streaming imposes delay constraints on each individual packet. Packets have to be decoded and then played back in order to ensure good quality of service experienced by the user.

Guaranteeing quality of streaming is a challenge when the channel is lossy or has a large transmission delay, such as the satellite communications channel. If unlimited transmission bandwidth was available, one could repeat a packet endlessly until it is successfully received. However, with limited available bandwidth one has to make a choice whether to repeat the packets which were already transmitted, or introduce new packets into the stream. Thus, there is a need to design efficient coding schemes which deliver packets with low delay while using a limited transmission bandwidth.

1.2 Previous Work

Traditionally automatic-repeat-request (ARQ) protocols are used for bandwidth limited packet transmission over a lossy channel. The source transmits a packet and waits for an acknowledgment (ACK) or negative acknowledgment (NACK). If a NACK is received, the source retransmits the lost packet, and if an ACK is received it moves forward and transmits the next packet. ARQ protocols are throughput optimal, that is they deliver the packet correctly with the minimum number of retransmissions. However, when feedback is lossy, delayed or completely absent, more efficient erasure-correcting codes are needed. This is because the encoder has inherent uncertainty about the state of the decoder, and it must strike a balance between transmitting new packets and repeating old packets that could have been erased.

Reed-Solomon codes [1] which map K source symbols to N (for $N > K$) channel symbols, can successfully correct up to $N - K$ erasures. However, these codes are practical only for small file size K. Fountain codes proposed in [2, 3, 4] are codes which can transmit files with large size K over an erasure channel without feedback. The source transmits a linear combination of a randomly chosen set of source symbols. The number of symbols included in each combination is determined by a carefully designed degree distribution. Fountain codes achieve channel capacity and have low encoding and decoding complexity.

Fountain codes are also referred to as rateless codes which means that the encoder generates a potentially limitless number of linear combinations of source symbols, such that the file of size K can be recovered when around K combinations are received without erasures. There is no fixed code rate such as the rate K/N of the Reed-Solomon codes described above. The receiver collects linear combinations, but almost no source symbols are decoded until around K combinations are successfully received. At this point, there is an avalanche of decoding and a large number of symbols are decoded from the linear combinations. Thus, the average decoding delay is proportional to the file size K. As a result, fountain codes are not suitable for packet streaming in which we have delay constraints because packets need to be decoded and played as soon as possible.

Only a few papers in literature have analyzed codes for packet streaming. Provably delay-optimal codes without feedback for adversarial and cyclic burst erasure channels have been extensively explored in [5]. The thesis also proposes universal codes for more general erasure models and analyzes their decoding delay. These codes are based upon sending linear combinations of source packets; indeed, it can be shown that there is no loss in restricting the codes to be linear.

This reduces the task of the coding scheme to deciding which packets should be included in every combination. The universal codes proposed in [5] are greedy codes where all packets generated so far are included in a combination. Greedy codes have also been proposed for other applications: in [6] for packet networks, and in [7] for a broadcast scenario with perfect feedback and proposes algorithms to reduce the buffer size at the source encoder. However, the delay performance of greedy codes has not been analyzed.

Many streaming applications involve playback. We thus choose to look at the *playback delay*, which takes this into account and reflects the end-to-end performance, rather than the more common decoding delay metric. In audio and video applications with correlation between packets, some packets can be dropped without affecting the quality of streaming. However, several other applications such as remote desktop have strict order constraints. For example, in remote desktop if the set of instructions moving a window and then closing another window behind it have to be executed in the exact order. Even if the decoding of one instruction is delayed, all the subsequent instructions get delayed. Our definition of playback delay is suitable for these applications. This definition was previously used in [8].

1.3 Our Contributions

The delay performance of greedy codes has not been analyzed and compared to other codes. This work aims to fill that gap, and in particular consider the playback delay. For the no-feedback case, we show that expected playback delay is proportional to $\log n$ for time index n. Thus, the key parameter in understanding the asymptotic behavior of delay is the proportionality constant, or pre-log. We find the optimal constant within a family of schemes that we call *time-invariant*, and conjecture that this is the optimum for any scheme. This optimum is attained by the conceptually simple coded repetition scheme.

With instantaneous feedback a simple ARQ based scheme is optimal. We show that even in this case, the playback delay has similar logarithmic growth, although with a smaller pre-log term. We evaluate that constant, and prove that feedback strictly helps reduce the growth of delay, though the gain vanishes in the limit of infinite bandwidth.The main results on the no-feedback and instantaneous feedback cases are presented in [9].

Unlike instantaneous feedback, the optimal code is not obvious when feedback is delayed. This is because the source has to make assumptions about erasures in the past slots while transmitting new packets. We propose a greedy coding scheme for streaming with delayed feedback and determine how the pre-log term in the growth of playback delay scales with feedback delay.

Finally, we extend the analysis of the point-to-point case to a broadcast streaming scenario where the source transmits a common packet stream to multiple users over erasure channels with instantaneous feedback. At any given time, each user has decoded a different subset of the stream based on its channel erasures. We present insights into designing an optimal scheme to transmit combinations such that each user decodes of packets immediately required for playback at each user. We analyze how the pre-log term in the growth of playback delay scales with the number of users served by the source. In particular, we can show that the case of infinite number of users is equivalent to point-to-point streaming without feedback.

1.4 Outline of the thesis

In Chapter 2, we define the system model and the class of coding schemes called full-rank codes that are of interest to us in this thesis. We list various notions of delay and compare their usefulness as a suitable metric to evaluate the performance of coding schemes. We also introduce the important concept of renewals in packet decoding which plays a key role in our analysis of delay in all subsequent chapters. Chapter 3 to Chapter 5 aim to determine the coding scheme which is optimal in terms of playback delay for three different streaming scenarios namely, no-feedback, delayed feedback, and broadcast with instantneous feeback.

In Chapter 3 we consider streaming over an erasure channel without feedback. We propose the coded repetition scheme and show that its expected playback delay is asymptotically equal to $1/\lambda \cdot \log n$ where n is the time slot index. We can determine a closed form expression for the growth constant λ in terms of the bandwidth and the erasure probability of the channel. Further we can show that the coded repetition is optimal among all time-invariant schemes.

In Chapter 4 we consider streaming with feedback about past erasures after a delay of d slots. We analyze the playback delay and determine how λ decays with feedback delay d. In particular, for the instantaneous feedback case we determine the exact value of λ. In Chapter 5 we consider the broadcast streaming setup with instantaneous feedback to the source. Even for this case, we determine how the growth constant decays with the number of users in the system.

Finally Chapter 6 gives a summary of results and discusses future research directions. Appendix A states some standard results that are used in proofs presented in the thesis.

Chapter 2

Preliminaries

In this chapter we describe the system model and define the main performance metrics used in our analysis of delay in packet streaming. In Section 2.1 we present the system model which is considered for the design of coding schemes in this thesis. In Section 2.2 we define some basic coding schemes which will be used for design of optimal codes in the subsequent chapters. In Section 2.3 we introduce the concept of renewals in packet decoding which play a key role in our analysis of delay performance. In Section 2.4 we define the different notions of delay that are used to compare coding schemes and show they can expressed in terms of renewals.

2.1 System Model

We consider a slotted packet transmission scenario as shown in Figure 2-1. A packet is a collection of some number of bits which we assume fixed and suppress in the sequel. The source and receiver are connected by an erasure channel with bandwidth

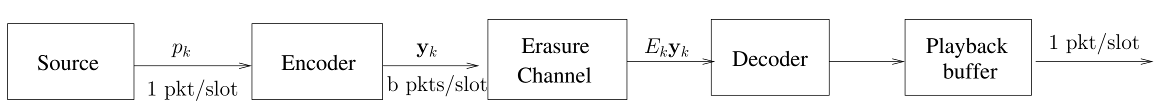


Figure 2-1: System model consists of source and a receiver connected by an erasure channel with bandwidth b packets per slot. $E_k = 1$ if the channel is good in slot k and 0 if that slot is erased. The source generates 1 packet/slot and 1 packet/slot is played in order at the receiver

b packets per slot, where we assume for simplicity that b is an integer and each channel packet is of the same size as a source packet. All the b encoded packets transmitted in that slot are received correctly with probability ρ, otherwise all are erased. In Figure 2-1, $E_k = 1$ if the channel is good in slot k and 0 if that slot is erased.

The source generates one packet per slot. We use p_k to denote the packet generated in slot k. In every slot, the encoder uses all the packets generated so far to create b packets of the same size as the source packet, denoted by the vector $\mathbf{y}_k = [y_{k,1},\ y_{k,2},\ \cdots\ y_{k,b}]$ of size b. Each encoded packet $y_{k,i}$, $1 \leq i \leq b$ is a function of the past packets $f(p_1, p_2, ..p_k)$. For simplicity of notation we drop the subscript i and use y_k when referring to a combination of packets p_1 to p_k.

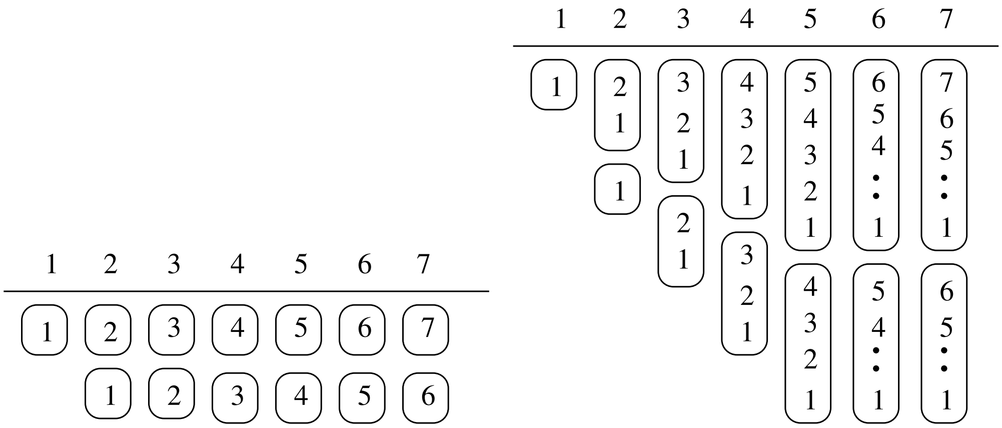
We assume without loss of generality that the transmission delay is zero, that is, a packet transmitted in slot k arrives at the receiver in slot k, or is erased by the channel. All notions of delay defined in this chapter can be modified to account for a non-zero transmission delay by simply adding the transmission delay to the delay metric.

The decoder can recover at most b source packets in every slot. It plays one packet per slot strictly in order. We consider that if a packet is decoded in slot k, it is available for playback from slot $k+1$ onwards. If any packet p_k has not been received, but p_j, $j > k$ is received, it is buffered until p_k is received and played.

In Chapter 3 we use this system model, where we further assume there is no feedback from the receiver to the source. In Chapter 4 we consider the case where delayed feedback about channel erasures is available to the encoder.

2.2 Coding Schemes

For the given system model, a simple strategy is to use a basic repetition scheme to transmit packet p_k, b times in slots $k,\ k+1,\ ..\ k+b-1$, as shown in Figure 2-2a. It is clear that a maximum of b packets will be transmitted per slot thus meeting the fixed bandwidth constraint. If all the b repetitions of a packet are erased, it can never be decoded and the playback of the stream will cease with probability 1. To avoid



(a) Basic repetition scheme (b) Augmented with Coding

Figure 2-2: Example illustrating the advantage of augmenting the basic repetition scheme with coding by transmitting linear combinations of the packets generated so far. Each bubble is a linear combination and the numbers inside it are the indices of packets included in that linear combination.

this, the scheme can be augmented with coding in the following simple manner.

Let y_k denote a linear combination of packets p_j for $1 \leq j \leq k$. We consider an augmented scheme where the source transmits combinations y_k in each slot as shown in Figure 2-2b. The combination $y_k = \sum_{j=1}^{k} c_j p_j$ where $c_j \in \mathbb{F}_q$ are chosen from a field of alphabet size q. Multiplication and addition operations in $\sum_{j=1}^{k} c_j p_j$ are also performed $\mathbb{F}_q$. The alphabet size q is chosen large enough to ensure that with high probability every each y_k is independent of all other y_j. Although we consider y_n as a linear combination here, in general it can be any function $f(p_1, p_2, \cdots, p_n)$ of packets p_1 to p_n.

As pointed out in [8], the delay performance of this coded scheme is at least as good as the uncoded repetition scheme. This is because p_k is played only when all p_j, $1 \leq j \leq n-1$ have been decoded and played. Coding offers the added advantage that if y_k is received when p_k has been decoded already, y_k can be used to decode one of the previous packets.

Now we use this idea to define a general class of codes called full-rank codes. All codes considered in this thesis belong to this class of codes.

Definition 2.1 (Full-rank codes)**.** *A full-rank code is the transmission scheme where in every slot the source transmits combinations* $y_k = \sum_{j=1}^{k} c_j p_j$ *of packets* p_1 *through*

p_k where $c_j \in \mathbb{F}_q$ for all $1 \leq j \leq k$ and $c_k > 0$. For any $n \geq k$ combinations $\mathbf{y}_k = [y_{k,1},\ \ y_{k,2}\ \ \cdots, y_{k,n}]$, the coefficients $c_{i,j}$ $1 \leq i \leq n$, $1 \leq j \leq k$ are such that the matrix

$$\begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,k} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,k} \end{bmatrix}$$

is full-rank.

This means that packets p_1 through p_k can be decoded from any k combinations of the form $y_k = \sum_{j=1}^{k} c_j p_j$. The transmission scheme shown in Figure 2-2b belongs the class of full-rank codes. We refer to it as the coded repetition scheme and formally define it as follows,

Definition 2.2 (Coded repetition scheme)**.** *The coded repetition scheme is the full-rank code in which the source transmits combinations $y_k = \sum_{j=1}^{k} c_j p_j$ of all packets generated until time k where $c_j \neq 0$ for all $1 \leq j \leq k$.*

Since we assume $c_j \neq 0$ for all j, we can simplify the notation in Figure 2-2b by representing each linear combination only by the maximum index among the packets included as shown in Figure 2-5. This convention is used in Chapter 3. In Chapter 4 and Chapter 5 the source may choose to exclude a past packet p_j from the combination to be transmitted by setting $c_j = 0$. In this case, we go back to the convention of denoting a linear combination by a bubble enclosing indices of the packets included.

The coded repetition scheme is a special case of a general class of packet transmission schemes which can be defined as follows,

Definition 2.3 (Time-invariant scheme)**.** *A time-invariant scheme with pattern $\mathbf{a} = [a_1\ a_2\ ..\ a_b]$ is the coding strategy where the source transmits combinations y_{n-a_i} of packets p_1 to p_{n-a_i}, for $1 \leq i \leq b$ in slot n, where $a_i > 0$, and $a_i < a_j$ for all $i < j$.*

The coded repetition scheme corresponds to the case $a_i = i - 1$ for all $1 \leq i \leq b$. Fig. 2-3 shows a typical transmission using a time-invariant scheme. Each number

Time	1	2	3	4	5	6	7	8 ⋯ k	
Packets Transmitted	1	2	3	4	5	6	7	8	$k - a_1$
			1	2	3	4	5	6	$k - a_2$
				1	2	3	4	5	$k - a_3$

$a_2 = 2$

$a_3 = 3$

Figure 2-3: Illustration of packet streaming using a time-invariant scheme with $b = 3$ and $\mathbf{a} = [0\ 2\ 3]$

in the packet transmitted shown in the figure denotes the linear combination $y_k = \sum_{j=1}^{k} c_j p_j$, with $c_j \neq 0$ for all j.

The constraint that $a_i < a_j$ for all $i < j$, is to ensure that each pattern $\mathbf{a}$ corresponds to a unique scheme. If $a_i = a_{i+1}$ for some i, setting $a_{i+1} = a_i + 1$ gives an equivalent scheme. This is because, when $a_i = a_{i+1}$ we transmit two independent linear combinations y_{k-a_i} and y'_{k-a_i} of packets p_1 through p_{k-a_i} in slot k. By eliminating p_k from one of the combinations, we obtain two equivalent combinations y_{k-a_i} and y_{k-a_i-1}, which are exactly the packets transmitted in the scheme with $a_{i+1} = a_i + 1$. Thus, a scheme with a_i's taking any non-negative values can be converted to an equivalent scheme with $a_i < a_j$ for all $i < j$.

In Chapter 3 we compare time-invariant schemes with different patterns $\mathbf{a}$, and determine whether using a particular pattern $\mathbf{a}$ is advantageous in reducing the delay in streaming. In the sequel, we define the concept of renewals which plays a key role in our analysis of delay of transmission schemes.

2.3 Renewals in packet decoding

The receiver is able to decode all packets up to the current time when the number of combinations received exceeds the number of packets generated. After this instant the decoding of future packets is independent of the past, and the system behaves as if it was reset to time zero. This phenomenon gives rise to the following definition of renewals in packet decoding.

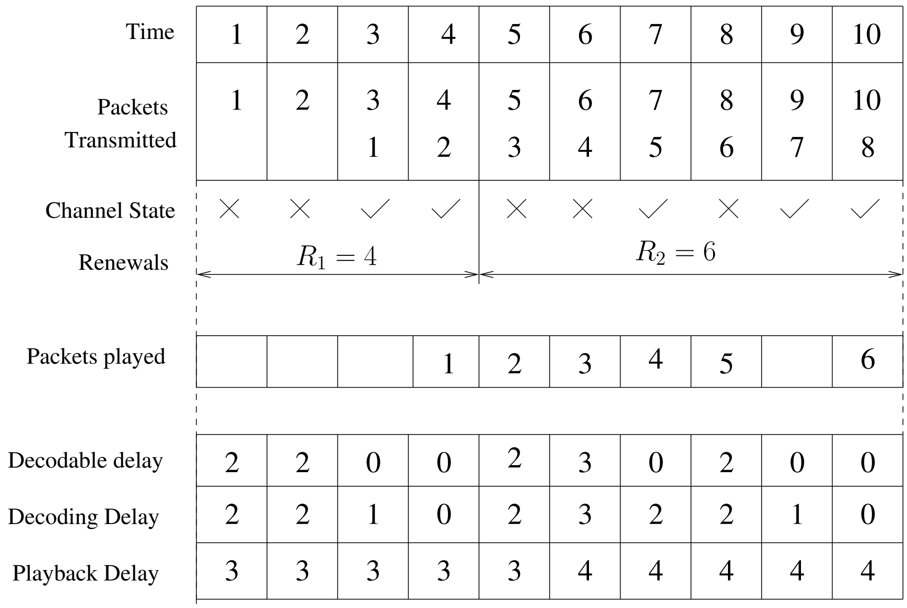


Figure 2-4: Illustration of renewals of the scheme with pattern $\mathbf{a} = [0\ 2]$

Definition 2.4 (Renewal)**.** *A renewal is defined as the earliest time n when all packets p_j, $1 \leq j \leq n$ have been decoded.*

The time between the $(i-1)^{th}$ and i^{th} renewal is defined as the i^{th} inter-renewal time denoted by R_i, where we assume that the 0^{th} renewal occurs at time zero. It is easy to show that inter-renewal times are i.i.d. The concept of a renewal is used extensively in stochastic processes [10]. The term information debt introduced in [5] is also closely related to renewals. Information debt is the amount of more information needed before successful decoding can occur. A renewal occurs when the information debt becomes non-positive.

Note that a renewal at time n does not imply that all packets were decoded exactly at n. A packet p_j may be decoded before time k. For example, in Figure 2-4 shows the scheme with pattern $\mathbf{a} = [0\ 2]$. For the channel realization shown, packet is decoded at time 3 although the first renewal takes place at time 4.

In the special case of the coded repetition scheme, decoding occurs only at renewal instants. Fig. 2-5 illustrates renewals of the coded repetition scheme for $b = 2$. The cross marks denote slots which experience channel blockage and tick marks denote slots in which packets are successfully received. In this example, the first two renewals

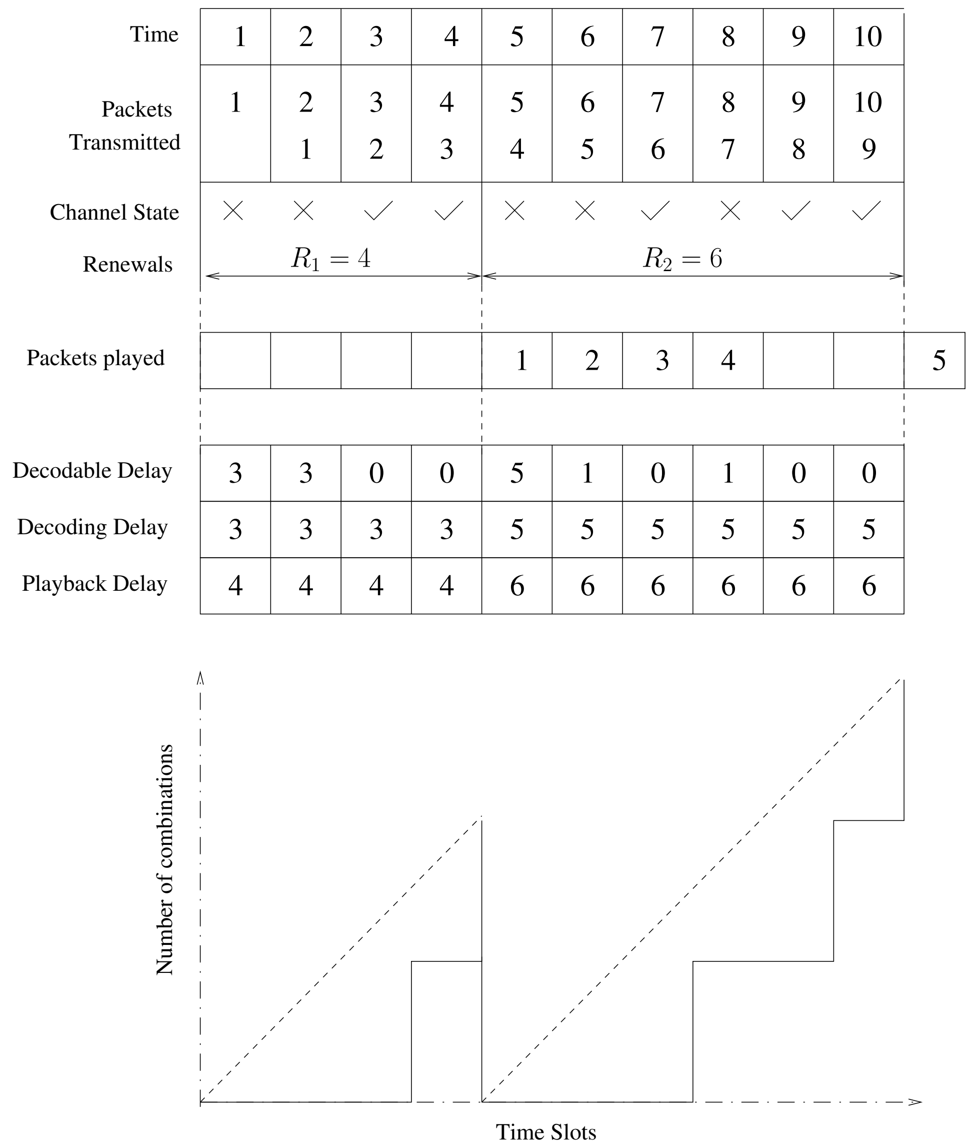


Figure 2-5: Illustration of renewals for the coded repetition scheme. Each number n in the packets transmitted denotes a linear combination of all packets p_1 through p_n. The cross marks denote slots which experience channel blockage and tick marks denote slots in which packets are successfully received.

occur at times 4 and 10. Thus, $R_1 = 4$ and $R_2 = 6$. The plot at the bottom of the figure shows the trajectory of the number of undecoded combinations received with time. A renewal takes place whenever the trajectory hits the slope one line.

2.4 Notions of Packet Delay

In this section we define different notions of delay and discuss their usefulness as a suitable performance metric in the packet streaming scenario. We show how each metric can be expressed in terms of renewals in packet decoding which simplifies its analysis.

1. **Decoding Delay:** A common delay metric is decoding delay, the time between the generation of a packet and until it is decoded at the receiver. It has been used in previous work [5, 6, 7, 11]. However, due to the constraint that the packets have to be played in order, a decoded packet cannot be used until all the previous packets have been decoded.

2. **Playback Delay:** The playback delay P_k of packet p_k is time between the generation of a packet at until it is played at the receiver. It is a natural delay metric for in-order playback, and hence is the main metric of interest in this thesis. For the coded repetition scheme, with pattern $a_i = i - 1$ for $1 \leq i \leq b$, when a renewal takes place at time n, all the packets p_1 through p_n are decoded exactly at time n. Thus

$$P_n = \max(R_1, R_2, ..R_k),$$

   where k is the smallest index such that $\sum_{i=1}^{k} R_i \geq n$. This is illustrated in Figure 2-5. Playback delay P_k is also equal to the total interruption time, or the number of slots in which no packet was played until time k. Since packets are played in strict order, P_k is monotonically increasing with k.

3. **Ordered Decoding Delay:** We define a new delay metric called ordered

decoding delay C_k of packet p_k which is the time between its generation and the earliest instant when all packets p_1 to p_k have been decoded. Unlike playback delay, the ordered decoding delay is stationary, i.e. the expected value $\mathbb{E}[C_k]$ does not change with time k. Ordered decoding delay is the right metric in applications where the packets have to be in order, but are not necessarily played back at the receiver. An example is remote desktop where a set of instructions are executed immediately after in-order decoding instead playing them back one by one.

4. **Decodable Delay:** We introduce another delay metric called decodable delay. A packet p_n is said to be decodable when the receiver can form a linear combination only of packets p_j, $1 \leq j \leq n$. In other words, when p_k is decodable, the decoder has sufficient information to decode it, given packets $p_1, p_2, ..p_{k-1}$. We define decodable delay D_k as the time between generation of p_k and when it first becomes decodable. Decoding implies decodability, but the converse is not true. Thus, the decoding delay $C_k \geq D_k$ for all k.

   Decodability plays a key role when the source gets feedback about past erasures. Once the source knows that p_k is decodable, it does not need to include that packet in any future combinations transmitted. Thus, decodability helps control the number of packets buffered at the source, and also reduces encoding and decoding complexity. We use the concept of decodability in designing coding schemes in Chapter 4 and Chapter 5. Another application of decodability is in designing codes for packet networks. A relay node can forward packets over to the next hop only after they become decodable. Thus, codes that minimize expected decodable delay are optimal for the source-relay link.

   Decodability is similar to the notion of packet being 'seen' defined in [7]. The difference is just a matter of convention. When a linear combination is received, the lowest index packet in that combination is marked 'seen'. Instead, we mark the maximum-index packet in that combination as decodable.

The concept of renewals in packet decoding and the various notions of delay introduce in this chapter play a key role throughout this thesis in the design and analysis of schemes that minimize delay.

Chapter 3

Streaming without feedback

In this chapter we consider the problem of packet transmission without feedback over a point-to-point i.i.d. erasure channel with success probability ρ, and fixed transmission bandwidth b packets per slot. We use the notions of packet delay defined in Chapter 2 to find the optimal scheme among the class of time-invariant schemes.

In Section 3.1 we determine the distribution of inter-renewal time for time-invariant schemes. We use this analysis in Section 3.2 to prove our main result that the expected playback delay $\mathbb{E}[P_n]$ grows asymptotically as ${}^{1}/{}_{\lambda} \log n$, where λ is referred to as the growth constant. The coded repetition scheme is optimal since it achieves the largest value of λ, among all time-invariant schemes.

In Section 3.3 we determine the expected ordered decoding delay $\mathbb{E}[C_k]$ of the coded repetition scheme. In Section 3.4 we analyze decodable delay and show that the coded repetition scheme gives minimum $\mathbb{E}[D_k]$ among all time-invariant schemes. In Section 3.5 we optimize the coded repetition scheme to allow lossy playback and reduce computational complexity. Finally, Section 3.6 shows how the results derived for an i.i.d erasure channel can be extended to more general channel models.

3.1 Properties of inter-renewal time

In this section we analyze the coded repetition scheme, which belongs to the class of time-invariant schemes defined in Chapter 2, and has pattern $a_i = i - 1$, $1 \leq i \leq b$.

We determine the closed form expression for probability mass function (PMF) of inter-renewal time R. This analysis is used in Section 3.2 to study the evolution of expected playback delay.

3.1.1 PMF of inter-renewal time

Lemma 3.1 (Distribution of inter-renewal time)**.** *For an i.i.d. erasure channel with success probability ρ and bandwidth b packets/slot, the PMF of inter-renewal time R for the coded repetition scheme is*

$$\Pr(R = n) = \left(1 - \frac{b(k-1)}{n-1}\right)\binom{n-1}{k-1}\rho^k(1-\rho)^{n-k}, \tag{3.1}$$

where $k = \lceil n/b \rceil$.

Proof. In each slot the decoder receives b equations with probability ρ and 0 with probability $1-\rho$. Let S_n be the number of equations received up to time n. Define the event $G_{n-1} = \{S_j < j \text{ for } 1 \leq j \leq n-1\}$, which means that there is no renewal up to slot $n-1$. The Generalized Ballot theorem from [12] given in Appendix A states that

$$\Pr(G_{n-1}|S_{n-1}) = \left(1 - \frac{S_{n-1}}{n-1}\right)^+, \tag{3.2}$$

where the operator $(x)^+ = \max(x, 0)$. For a renewal to occur at time n, $b\,(k-1)$ equations, where $k = \lceil n/b \rceil$, should be received in $n-1$ slots and the channel should be good in the n^{th} slot. Thus,

$$\Pr(R = n) = \rho \cdot \Pr(G_{n-1}|S_{n-1}) \Pr\left(S_{n-1} = b(k-1)\right). \tag{3.3}$$

Substituting (3.2) and the PMF of binomial distribution for $\Pr\left(S_{n-1} = b(k-1)\right)$, we get (3.1). □

3.1.2 Asymptotic behavior of the PMF

Since we are interested in the long term evolution of playback delay P_n, it is useful to look at the behavior of the distribution $\Pr(R = n)$ for large n. We show that it decays exponentially at rate $\lambda_c \triangleq D(1/b||\rho)$. In this definition, $D(p||q)$ is the binary information divergence function which is defined for probabilities $0 < p, q < 1$ as,

$$D(p||q) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}.$$

Lemma 3.2 (Asymptotic behavior of the PMF)**.** *For the coded repetition scheme, the tail distribution of inter-renewal time R decays exponentially with rate*

$$-\lim_{n\to\infty} \frac{\log \Pr(R > n)}{n} = D\left(\frac{1}{b}||\rho\right) = \lambda_c \tag{3.4}$$

Proof. The tail distribution of R is given by,

$$\begin{aligned}
Pr(R > n) &= \sum_{k=1}^{\lceil \frac{n}{b} \rceil - 1} Pr(G_n|S_n).Pr(S_n = bk), &(3.5)\\
&= \sum_{k=1}^{\lceil \frac{n}{b} \rceil - 1} \binom{n}{k} \rho^k (1-\rho)^{n-k} \left(1 - \frac{bk}{n}\right), &(3.6)\\
&= \sum_{k=1}^{\lceil \frac{n}{b} \rceil - 1} \sqrt{\frac{n}{2\pi k(n-k)}} \cdot \frac{e^{\mu_n}}{e^{\mu_k + \mu_{n-k}}} \left(1 - \frac{bk}{n}\right) \cdot e^{-nD(\frac{k}{n}||\rho)}, &(3.7)\\
&= e^{-f_1(n) - nD(\frac{1}{b}||\rho)}, &(3.8)\\
&\doteq e^{-nD(\frac{1}{b}||\rho)}. &(3.9)
\end{aligned}$$

We apply the Stirling's approximation for factorials in the form

$$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n e^{\mu_n}$$

where $\mu_n = O(1/n)$ to the binomial in (3.6) and obtain (3.7). For large n, the $k = \lceil n/b \rceil - 1$ term in the summation dominates. Thus we get (3.8) where function $f_1(n)$ is such that $\lim_{n\to\infty} \frac{f_1(n)}{n} = 0$. In (3.9), the '$\doteq$' sign stands for asymptotic equality where the relation $f(n) \doteq g(n)$ between some functions f and g means that,

$$\lim_{n\to\infty} \frac{\log f(n)}{n} = \lim_{n\to\infty} \frac{\log g(n)}{n}.$$

□

3.2 Playback delay

In this section we analyze the expected playback delay $\mathbb{E}[P_n]$ for large n and prove that the coded repetition scheme is the optimal transmission scheme that gives minimum rate of growth of playback delay. We prove the following main theorem,

Theorem 3.1 (Expected Playback Delay)**.** *For the optimal time-invariant scheme, the expected playback delay $\mathbb{E}[P_n]$ satisfies*

$$\mathbb{E}[P_n] = \frac{1}{\lambda_c} \log n + O(\log \log n). \tag{3.10}$$

The achievability and converse parts of Theorem 3.1 are proved in the following Section 3.2.1 and Section 3.2.2 respectively.

3.2.1 Achievability proof

The achievability part of Theorem 3.1 is an immediate corollary of the following lemma. It shows that coded repetition scheme achieves the optimal $\lambda_c = D(1/b||\rho)$ in (3.10).

Lemma 3.3 (Performance of the coded repetition scheme)**.** *For the coded repetition scheme, the expected playback delay $\mathbb{E}[P_n]$ satisfies*

$$\mathbb{E}[P_n] \leq \frac{1}{\lambda_c} \log n + O(1), \tag{3.11}$$
$$\mathbb{E}[P_n] \geq \frac{1}{\lambda_c} \log n - \log \log n + O(1). \tag{3.12}$$

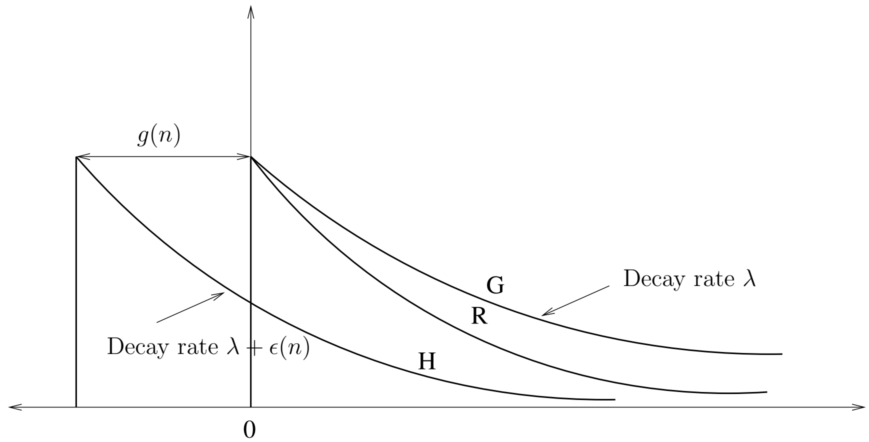


Figure 3-1: The tail distribution of inter-renewal R is upper and lower bounded by geometric distributions G and H respectively.

Proof. For the coded repetition scheme, $P_n = \max(R_1,\ R_2,\ ..,\ R_J)$ where J is the smallest integer such that $\sum_{k=1}^{J} R_k \geq n$, and R_k's are i.i.d. with distribution of the inter-renewal time in Lemma 3.1. Thus,

$$\mathbb{E}[P_n] = \mathbb{E}_J \left[\mathbb{E} \left[\max(R_1, R_2, ..R_J) \middle| \sum_{i=1}^{J-1} R_i < n, \sum_{i=1}^{J} R_i \geq n \right] \right]. \tag{3.13}$$

We use this to prove the upper and lower bounds (3.11) and (3.12) on $\mathbb{E}[P_n]$ as follows. From Lemma 3.2 we know that the tail distribution

$$\Pr(R > m) = e^{-f_1(m) - m\lambda_c}, \tag{3.14}$$

where the function $f_1(m)$ is such that $\lim_{m\to\infty} {}^{f_1(m)}/_{m} = 0$.

To get an upper bound, we define a geometric random variable G with decay rate λ_c. By its definition, we know that the tail distribution of G, $\Pr(G > m) = e^{-m\lambda_c} \geq \Pr(R > m)$ for all m. Thus,

$$\mathbb{E}[\max(R_1, R_2, ..R_J)] \leq \mathbb{E}[\max(G_1, G_2, ..G_J)], \tag{3.15}$$

$$\leq \frac{1}{\lambda_c} \log J + O(1), \tag{3.16}$$

where in (3.16) we use the result given in [13] that the expectation of the maximum of J geometric random variables with decay rate λ_c is ${}^1\!/\!_{\lambda_c} \cdot \sum_{i=1}^{J} {}^1\!/\!_i$, which is asymptotically $\log J$. By the strong law of renewal processes stated in Appendix A with the detailed proof given in [10], we know that J grows linearly with n. Thus, the expectation of (3.16) with respect to J, replaces $\log J$ by $\log n$ and adds a constant, or an $O(1)$ term to give the upper bound (3.11).

Similarly, we derive the lower bound (3.12) by defining another geometric random variable H with decay rate $\lambda_c+\epsilon(n)$ and shifted $g(n)$ units to the left of 0 as illustrated in Fig. 3-1. The functions $g(n)$ and $\epsilon(n)$ are chosen such that for all m,

$$\begin{aligned} \Pr(H > m) &\leq \Pr(R < m), \\ e^{-(m+g(n))(\lambda_c+\epsilon(n))} &\leq e^{-f_1(m)-m\lambda_c}, \\ \epsilon(n) &\geq \frac{f_1(m) - \lambda_c g(n)}{m + g(n)}. \end{aligned} \tag{3.17}$$

We choose function $g(n) = \log\log n$. For large enough n, the right-hand side of (3.17) will be negative and hence we can choose $\epsilon(n) = 0$. Thus for large n we have,

$$\begin{aligned} \mathbb{E}[\max(R_1,\, R_2,\, ..\, R_J)] &\geq \mathbb{E}[\max(H_1,\, H_2,\, ..\, H_J)], \\ &\geq \frac{1}{\lambda_c + \epsilon(n)} \log J - g(n) + O(1), \\ &= \frac{1}{\lambda_c} \log J - \log\log n + O(1). \end{aligned} \tag{3.18}$$

Again, using the strong law of renewal processes and taking the expectation over J of (3.18) gives the lower bound (3.12). $\square$

Thus, we have shown that the $\mathbb{E}[P_n]$ of the coded repetition scheme is asymptotically ${}^1\!/\!_{\lambda_c} \cdot \log n$ where $\lambda_c = D({}^1\!/\!_b||\rho)$.

3.2.2 Converse proof

The converse part of Theorem 3.1 is a corollary of the following lemma where we prove that no other time-invariant scheme can achieve a larger growth constant λ than λ_c achieved by the coded repetition scheme.

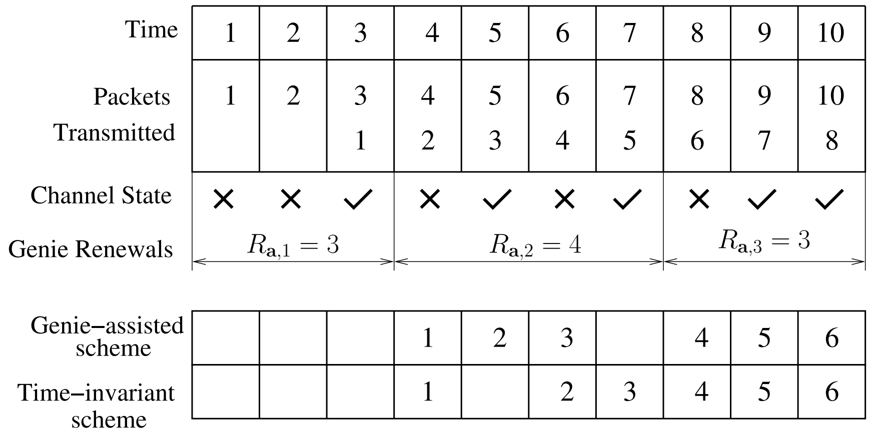


Figure 3-2: Illustration of the difference between the time-invariant scheme with pattern $\mathbf{a} = [0\ 2]$ and its genie-assisted form for $b = 2$. The two bottom rows show the packets played in every slot for the two schemes. Empty boxes indicate interrupted slots where no packet is played.

Lemma 3.4 (Performance of any time-invariant scheme)**.** *For any time-invariant scheme with pattern* $\mathbf{a}$ *as defined in Definition 2.3, the expected playback delay* $\mathbb{E}[P_n]$ *satisfies*

$$\mathbb{E}[P_n] \geq \frac{1}{\lambda_{\mathbf{a}}} \log n + O(\log \log n), \tag{3.19}$$

where $\lambda_{\mathbf{a}} \leq \lambda_c$ *for all* $\mathbf{a}$*.*

To simplify the analysis of playback delay, we define a genie-assisted lower bound for every time-invariant scheme. Whenever the actual scheme decodes the first packet in a renewal interval at time n, we consider that a genie at the source results in decoding of all packets up to p_n in the genie-assisted scheme. A renewal with the actual scheme implies a renewal with the genie-assisted scheme at that time instant. However, the converse is not true. Fig. 3-2 illustrates the difference between the time-invariant scheme with pattern $\mathbf{a} = [0\ 2]$ and its genie bound.

Let $R_{\mathbf{a}}$ be inter-renewal time of the genie-assisted scheme with pattern $\mathbf{a}$. Then, the playback delay after n slots $P_n^* = \max(R_{\mathbf{a},1}, R_{\mathbf{a},2}\ ..R_{\mathbf{a},K})$ where K is the smallest integer such that $\sum_{i=1}^{K} R_{\mathbf{a},i} \geq n$. Let $\lambda_{\mathbf{a}}$ be the decay rate of its tail distribution as defined in Lemma 3.2. We can prove the following result,

Lemma 3.5 (Asymptotic decay rate for time-invariant schemes)**.** *The decay rate $\lambda_{\mathbf{a}}$ of the genie-assisted time-invariant scheme with pattern $\mathbf{a}$ is such that, $\lambda_{\mathbf{a}} \leq \lambda_c$ for all $\mathbf{a}$.*

Proof. We lower bound the tail distribution of $R_{\mathbf{a}}$ by

$$\begin{aligned}
\Pr(R_{\mathbf{a}} > n) &\geq (1-\rho)^{a_b+1} \Pr(R_{\mathbf{a}} > n | \text{first } a_b + 1 \text{ slots erased}), \\
&\geq (1-\rho)^{a_b+1} . \Pr(R > n - a_b - 1), \qquad (3.20) \\
&\doteq e^{-n\lambda_c}.
\end{aligned}$$

where in (3.20), R is the inter-renewal time of the coded repetition scheme and its tail distribution is as derived in Lemma 3.2. □

Proof of Lemma 3.4. Since the genie-assisted version gives a lower bound on the playback delay of the actual time-invariant scheme we have,

$$\mathbb{E}[P_n] \geq \mathbb{E}[\max(R_{\mathbf{a},1}, R_{\mathbf{a},2} \;..R_{\mathbf{a},K})], \qquad (3.21)$$

where in K is the smallest integer such that $\sum_{i=1}^{K} R_{\mathbf{a},i} \geq n$. We then obtain the results (3.19) by using analysis similar to the proof of Lemma 3.3, but applied to renewals of the genie-assisted scheme. □

Thus, we have shown that the coded repetition scheme gives the largest growth constant λ among all time-invariant schemes. We have the following conjecture about time-variant schemes.

Conjecture 3.1. *No scheme can achieve a larger value of growth constant λ than λ_c for the coded repetition scheme.*

We believe this is true because in absence of feedback, the statistics of undecoded packets are asymptotically stationary. Although the playback delay is not stationary, it is a function of the undecoded packets. Thus, using a time-varying scheme cannot improve the playback delay performance.

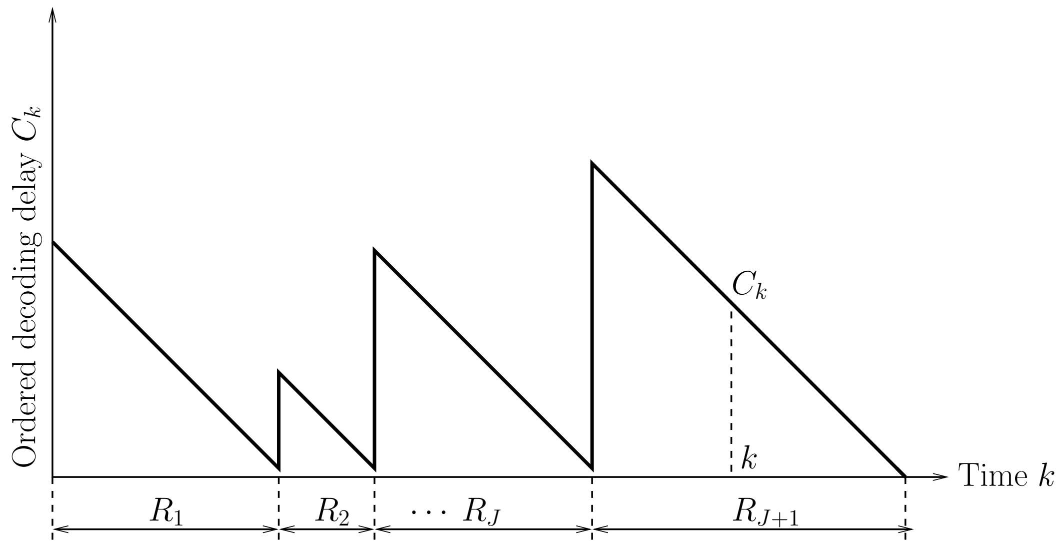


Figure 3-3: Ordered decoding delay of packet p_k is the time duration from k to the next renewal instant. It is same as the residual life of a renewal process

3.3 Ordered Decoding Delay

In Section 2.4 we defined the ordered decoding delay C_k. In this section we determine $\bar{C}$, the time-averaged ordered decoding delay for the coded repetition scheme in terms of moments of the inter-renewal time.

Lemma 3.6 (Time-average Ordered Decoding Delay)**.** *For the coded repetition scheme with inter-renewal time R, the time-averaged ordered decoding delay is*

$$\bar{C} = \frac{\mathbb{E}[R^2]}{2\mathbb{E}[R]} - \frac{1}{2} \quad \textit{with probability 1.} \tag{3.22}$$

Proof. The ordered decoding delay corresponds to the residual life of a renewal process [10]. Suppose J renewals have occurred till time n. If the length of the j^{th} renewal interval is R_j, then the ordered decoding delays of the R_j packets in that interval are $R_j - 1$, $R_j - 2$, and 1 respectively as shown in Figure 3-3. Thus, the average C_k over this time window can be bounded above and below as follows,

$$\sum_{j=1}^{J} \frac{R_j(R_j-1)}{2n} \leq \frac{\sum_{k=1}^{n} C_k}{n} \leq \sum_{j=1}^{J+1} \frac{R_j(R_j-1)}{2n} \tag{3.23}$$

Taking the limit $n \to \infty$, the upper and lower bounds converge to the same value with probability one. We now evaluate the lower bound.

$$\lim_{n\to\infty} \sum_{j=1}^{J} \frac{R_j(R_j-1)}{2n} = \lim_{J\to\infty} \frac{\sum_{j=1}^{J} R_j(R_j-1)}{2J} \cdot \lim_{n\to\infty} \frac{J}{n} \tag{3.24}$$

$$= \frac{\mathbb{E}[R^2]-\mathbb{E}[R]}{2} \frac{1}{\mathbb{E}[R]} \quad \text{with probability } 1 \tag{3.25}$$

By the strong law of large numbers, the first limit in (3.24) converges to $(\mathbb{E}[R^2] - \mathbb{E}[R])/2$ with probability 1. The second limit converges to $1/\mathbb{E}[R]$ with probability 1 by the strong law of renewal processes.

□

Since C_k is an ergodic process, the time averaged ordered decoding delay $\bar{C} = \mathbb{E}[C_k]$. In Section 3.1 we derived the distribution of R and showed that it is asymptotically exponential. If R is purely geometric distributed, we get $\bar{C} = \mathbb{E}[R] - 0.5$. Thus, $\mathbb{E}[C_k] = O(\mathbb{E}[R])$ for the coded repetition scheme. We conjecture that the coded repetition scheme gives the minimum $\mathbb{E}[C_k]$ among all time-invariant schemes.

3.4 Decodable Delay

In Chapter 2 we defined the decodable delay D_k of p_k as the time from packet generation, until when the receiver can form a linear combination of p_i and all past packets p_j, $1 \leq j < i$. In this section we present the following main result.

Theorem 3.2. *The coded repetition scheme, with $a_i = i-1$ for all $1 \leq i \leq b$ gives the minimum time-averaged decodable delay among all time-invariant schemes for every channel realization.*

The proof of this theorem given in Appendix B. Since D_k is an ergodic process, the time-average equals ensemble average and we obtain the following corollary of Theorem 3.2.

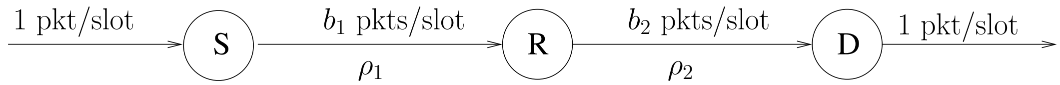


Figure 3-4: System model of streaming over a line network. The coded repetition scheme is optimal for the source-relay link because it gives minimum expected decodable delay among time-invariant schemes as proved in Theorem 3.2.

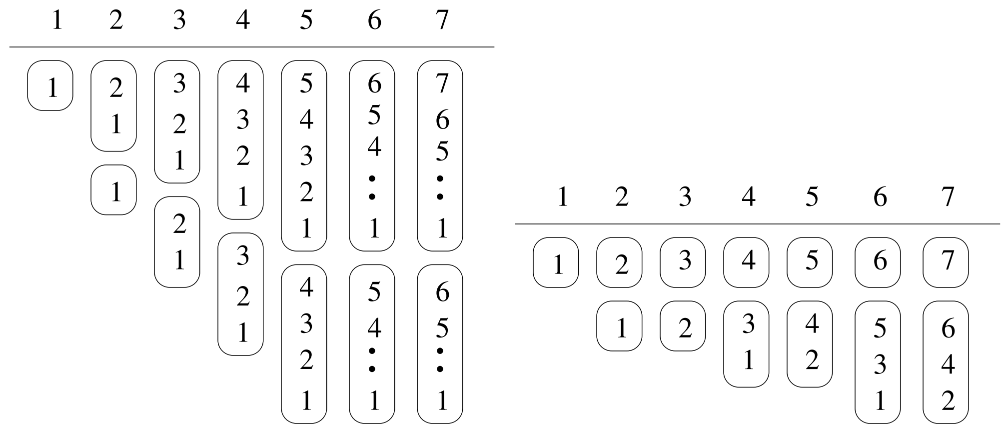
Corollary 3.1. *The coded repetition scheme gives the minimum expected decodable delay* $\mathbb{E}[D_k]$ *among all time-invariant schemes. The distribution of* D_k *is same as inter-renewal time* R *given in (3.1).*

3.4.1 Application to relay network

An interesting application of Corollary 3.1 is in finding the optimal code for streaming over a line network without feedback as shown in Figure 3-4. The line network has a source node and a destination node connected via a relay by two erasure channels with bandwidth b_1 and b_2 packets per slot respectively. The success probabilities of the channels are ρ_1 and ρ_2 respectively. One packet per slot is generated at the source, and decoded packets are played one packet per slot in exact order at the destination. The concept of decodability is applicable here because a relay node can transmit a combination $y_k = \sum_{i=1}^{k} c_i p_i$ to the next hop only after p_i becomes decodable. Since the coded repetition scheme gives the minimum expected decodable delay it is the optimal transmission scheme for the source-relay link. However, we cannot argue this for the relay destination link because playback of packets at the destination is involved.

3.5 Reduced coded repetition scheme

The coded repetition scheme was shown to be optimal in terms of playback delay in Section 3.2. In this section we show that it is possible to reduce the number of packets included in each combination without affecting playback delay if packets are played strictly in order at the receiver. In applications such as audio or video streaming, the receiver can drop some packets or play their interpolated versions without affecting he



(a) Coded repetition scheme (b) Reduced scheme

Figure 3-5: Reduced coded repetition scheme for $b = 2$. Each bubble is a linear combination and the numbers inside it are the indices of packets included in that linear combination.

quality of streaming. Unlike the coded repetition scheme, the reduced version allows packet dropping in applications where lossy playback is acceptable. In addition, minimizing the number of packets in every combination reduces the encoding and decoding complexity.

3.5.1 Elimination packets from a combination

Our objective is to eliminate as many packets as possible from each transmitted combination, while ensuring that the received combinations are innovative for every possible erasure pattern. In other works, we keep packet p_i in a combination transmitted in slot n only if it leads to decoding of p_i for at least one erasure pattern of slots 1 to n. The formal algorithm to perform this elimination of unnecessary packets in each combination is Algorithm 3.1.

The reduced version of the coded repetition scheme is shown in Figure 3-5. We observe that the reduced scheme divides the packet stream is divided into b parallel sub-streams such that the i^{th} sub-stream consists of combinations of packets p_n where $i = n \mod b$. With strict in-order playback, the reduced scheme is equivalent to the original scheme, that is, it gives the same playback delay as the coded repetition scheme for every packet in the stream.

Algorithm 3.1 Minimizing packets in a combination

for $i = n$ to 1 **do**
    $set\_of\_patterns \leftarrow$ all erasure patterns up to time $n-1$ for which p_i is not decodable.
    $keep\_in\_combn[i] \leftarrow false$
    while $set\_of\_patterns$ is not empty **and** $keep\_in\_combn[i] = 0$ **do**
        if Adding p_i to this combination leads to decoding of p_i **then**
            $keep\_in\_combn[i] \leftarrow 1$
        else
            Remove current pattern from $set\_of\_patterns$
        end if
    end while
end for
Form a combination of all p_i for which $keep\_in\_combn[i] = 1$

3.6 Generalizing the channel model

So far we considered an i.i.d erasure channel with channel success probability ρ. We now show how the results on analysis of delay for streaming without feedback can be extended to more general channel models. The analysis of playback delay for streaming with feedback and broadcast streaming considered in Chapter 4 and Chapter 5 respectively can also be extended in a similar manner.

3.6.1 Markov Erasure Channel

Consider the two-state Markov channel model shown Figure 3-6. ρ_1 and ρ_2 are the bad-to-good and good-to-bad state transition probabilities respectively, where $0 < \rho_1, \rho_2 < 1$. We require the condition $\frac{\rho_1 b}{(\rho_1+\rho_2)} > 1$ to ensure that the rate of packet generation at the source is less than channel capacity.

For every visit to state 1, b combinations are received and a renewal takes place when the number of combinations exceeds the number of time slots. Clearly, the channel has to be in the good state when a renewal takes place and by the Markov property, successive inter-renewal times R_i are i.i.d. Let S_k be the time of the k^{th} visit to state 1. Then, $S_k = Z_1 + Z_2 + ..Z_k$ where Z_i is the time between the $(i-1)^{th}$ and i^{th} visit to state 1.

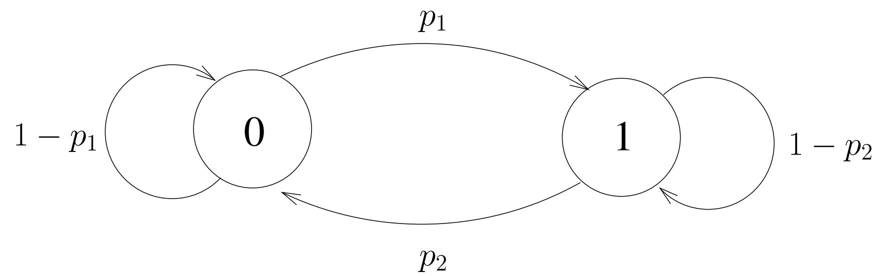


Figure 3-6: Two state Markov channel model. '0' denotes bad state or erasure and '1' denotes good channel state. ρ_1 and ρ_2 are the bad-to-good and good-to-bad transition probabilities respectively.

Asymptotic Behavior of PMF of R

The decoder receives b equations every time the system visits state 1. A renewal does not take place after k visits to state 1 if, $S_i > ib$ for all $1 \leq i \leq k$. Thus,

$$Pr(R > n) = Pr\{S_1 > b, S_2 > 2b, ..S_{\lceil \frac{n}{b} \rceil} > n\}, \tag{3.26}$$
$$= Pr\{S'_1 > 0, S'_2 > 0, ..S'_{\lceil \frac{n}{b} \rceil} > 0\}, \tag{3.27}$$
$$= Pr\left(K > \lceil \frac{n}{b} \rceil\right), \tag{3.28}$$

where in (3.27), S'_k corresponds to the shifted random walk with $Z'_i = Z_i - b$. The new random walk has a negative drift since $\mathbb{E}(Z') = \mathbb{E}(Z) - b < 0$. In (3.28), K is the smallest integer for which $S'_K \leq 0$. We now apply Wald's identity given in Appendix A to determine the tail distribution $\Pr(K > k)$ of stopping time.

$$1 = \mathbb{E}[\exp(rS'_K - K\gamma(r))], \tag{3.29}$$
$$\geq \Pr(K > k) \cdot \mathbb{E}[\exp(rS'_K - J\gamma(r))|K > k], \tag{3.30}$$
$$\geq \Pr(K > k) \cdot \mathbb{E}[\exp(rZ'_K - k\gamma(r))], \tag{3.31}$$
$$\geq \Pr(K > k) \cdot \exp(-\inf_{r>0} \gamma(r)(k-1)), \tag{3.32}$$

where $\gamma(r)$ is the log of the moment generating function of Z'. Since $\gamma'(0) = \mathbb{E}(Z') < 0$, and $\gamma(r)$ is a convex function, there exists $\gamma(r) < 0$ for some $r > 0$. Hence, in (3.28) $Pr(K > \lceil \frac{n}{b} \rceil)$ decays exponentially with rate $\lambda_m \triangleq -\inf_{r>0} \gamma(r)$.

Playback Delay

We can prove Theorem 3.1 in Chapter 3 for the Markov erasure channel. The achievability and converse parts can be proved in the same way as Lemma 3.3 and Lemma 3.4 with λ_c replaced by λ_m.

Achievability proof: For the coded repetition scheme, the playback delay P_n after n slots is given by $P_n = \max(R_1, R_2, \cdots R_J)$ where J is the smallest integer such that $\sum_{j=1}^{J} R_j \geq n$. Since $Pr(R > k) \doteq \exp(-\lambda_m k)$, the expected playback delay satisfies,

$$\mathbb{E}[P_n] \leq \frac{1}{\lambda_m} + O(1) \tag{3.33}$$

$$\mathbb{E}[P_n] \geq \frac{1}{\lambda_m} - \log\log n + O(1). \tag{3.34}$$

We prove this by considering geometric random variables G and H and the upper and lower bounding $\mathbb{E}[\max(R_1, R_2, \cdots R_J)]$ as done in the proof of Lemma 3.3.

Converse proof: We can show that no other time-invariant scheme can achieve a growth constant $\lambda \geq \lambda_m$ than the coded repetition scheme. We consider a genie-assisted version of a time-invariant scheme with pattern $\mathbf{a}$ where we assume that a renewal occurs when the first packet is decoded. Then, we can lower bound the tail distribution of inter-renewal time as follows.

$$\begin{aligned}
\Pr(R_{\mathbf{a}} > n) &= \Pr(p_1 \text{ not decoded until time } n) && (3.35)\\
&\geq \Pr(\text{first } a_b + 1 \text{ slots erased}).p_1. \Pr(R > n - a_b - 2) && (3.36)\\
&= \rho_2(1-\rho_1)^{a_b} p_1 \Pr\{S_1 > b, S_2 > 2b, ..S_{\lceil \frac{n-a_b-2}{b} \rceil} > n - a_b - 2\} && (3.37)\\
&\doteq \Pr\{S_1 > b, S_2 > 2b, ..S_{\lceil \frac{n}{b} \rceil} > n\} && (3.38)\\
&\doteq \exp(-\lambda_m n) && (3.39)
\end{aligned}$$

Thus, we have proved that for all patterns $\mathbf{a}$, the growth constant of playback delay for any time-invariant scheme is worse than that achieved by the coded repetition scheme.

3.6.2 Partial erasures of packets in a slot

So far we assumed that the b packets transmitted a slot are received with probability ρ, otherwise all are erased. We can generalize this model so that a subset of the b packets could be erased. Consider that i packets in a slot are received successfully, with probability ρ_i. Thus, $\rho_0 = 1 - (\rho_1 + ..\rho_b)$ is the probability that all the b packets transmitted in a slot are erased.

PMF of inter-renewal time

For a given window of n slots we define the empirical distribution $\mathbf{q} = [q_0 \; q_1 \; \cdots \; q_b]$, where q_i is the fraction of slots in which i out the b transmitted packets were erased. For a renewal to take place at time n, we need $\sum_{i=0}^{b} iq_i \geq n$. Let ξ_n be i.i.d random variables corresponding to the number of equations received in each slot. The distribution of ξ is,

$$\Pr(\xi = i) = \rho_i \quad \text{for } 0 \leq i \leq b \tag{3.40}$$

Let $S_n = \sum_{i=1}^{n} \xi_i$, the total number of combinations received up to time n. It has the multinomial distribution,

$$\Pr(S_n = m) = n! \sum_{\mathbf{q} \in \mathcal{Q}_m} \left(\frac{\prod_{i=0}^{b} \rho_i^{q_i n}}{\prod_{i=1}^{b} (q_i n)!} \right) \tag{3.41}$$

where $\mathcal{Q}_m$ is the set of distributions $\mathbf{q}$ such that $\sum_{i=0}^{b} iq_i = m$. Define G as the event there is no renewal until time $n-1$. We use the generalized Ballot theorem stated in Appendix A to determine the PMF of inter-renewal time as follows.

$$\Pr(R = n) = \sum_{i=0}^{b} \Pr(G|S_{n-1} = n-i) \cdot \Pr(S_{n-1} = n-i) \cdot \Pr(\xi_n = i), \tag{3.42}$$

$$= \sum_{i=2}^{b} \rho_i \left(1 - \frac{n-i}{n-1}\right) \Pr(S_{n-1} = n-i), \tag{3.43}$$

$$= \sum_{i=2}^{b} \rho_i \left(\frac{i-1}{n-1}\right) n! \sum_{\mathbf{q} \in \mathcal{Q}_{n-i}} \left(\frac{\prod_{i=0}^{b} \rho_i^{q_i n}}{\prod_{i=1}^{b} (q_i n)!} \right), \tag{3.44}$$

where in (3.42), $\Pr(G|S_{n-1})$ is given by the generalized Ballot theorem. In (3.43) we can remove the $i = 0$ and $i = 1$ from the summation because for $\xi_n = 0$ and $\xi_n = 1$, there would already be a renewal in some slot before n.

Asymptotic Behavior of the PMF

We now determine the asymptotic behavior of the distribution of inter-renewal time. We simplify the expression of the tail distribution as follows,

$$
\begin{aligned}
Pr(R > n) &= \sum_{m=0}^{n-1} \Pr(G|S_n)\Pr(S_n = m) && (3.45)\\
&= \sum_{m=0}^{n-1} \left(1 - \frac{m}{n}\right) n! \sum_{\mathbf{q}\in\mathcal{Q}_m} \left(\frac{\prod_{i=0}^{b} \rho_i^{q_i n}}{\prod_{i=1}^{b} (q_i n)!} \right) && (3.46)\\
&\doteq \sum_{m=0}^{n-1} \exp(-n \min_{\mathbf{q}\in\mathcal{Q}_m} D(\mathbf{q}||\boldsymbol{\rho})) && (3.47)\\
&\doteq \exp(-n \min_{\mathbf{q}\in\mathcal{Q}_{n-1}} D(\mathbf{q}||\boldsymbol{\rho})). && (3.48)
\end{aligned}
$$

Thus, the tail distribution of inter-renewal time decays with $\lambda = \min_{\mathbf{q}\in\mathcal{Q}_{n-1}} D(\mathbf{q}||\boldsymbol{\rho}))$. By the Pythagoras theorem for distributions stated in Appendix A, the optimal $\mathbf{q}^*$ lies on the exponential family starting from $\mathbf{p}$ and parametrized by some r. Thus, it is the solution to the following system of equations,

$$
\begin{aligned}
q_i = \rho_i \cdot \exp(ri - \psi(r)) \text{ for all } 0 \leq i \leq b && (3.49)\\
\sum_{i=0}^{b} i \cdot q_i = n - 1 && (3.50)
\end{aligned}
$$

where $\psi(r) = \log \mathbb{E}[\exp(r\xi)]$, the log MGF of ξ. The resulting minimum $\lambda_p \triangleq D(\mathbf{q}^*||\boldsymbol{\rho})$ is the rate of decay of the tail distribution of inter-renewal time.

Playback Delay

We can prove that for this erasure model, the expected playback delay of the coded repetition scheme satisfies Theorem 3.1 with λ_c replaced by λ_p and no other time-invariant scheme can achieve a larger value of growth constant. The proof of achievability and converse is same as shown in Lemma 3.3 and Lemma 3.4 respectively.

3.7 Conclusions

In this chapter we analyzed the delay in streaming over an erasure channel with bandwidth b and success probability ρ, without any feedback to the source. Our main metric of interest was playback delay. We proposed a simple greedy coding strategy called the coded repetition scheme and showed that the expected playback delay is asymptotically ${}^{1}\!/\!{}_{\lambda} \cdot \log n$ where λ is referred to as the growth constant. Further, we proved that the coded repetition scheme is optimal, that is it achieves the largest growth constant $\lambda = D({}^{1}\!/\!{}_{b}||\rho)$ among all time-invariant schemes.

We also analyzed coding schemes in terms of other delay metrics defined in Chapter 2. In Section 3.5 we modified the coded repetition scheme to allow lossy playback and reduce the encoding and decoding complexity. Finally, in Section 3.6 we showed that the analysis presented for an i.i.d erasure channel can be extended to more general channel models.

From this chapter we conclude that when there is no feedback to the source, the coded repetition scheme is optimal in terms λ, the pre-log term in the growth of playback delay. However, if the source receives feedback about the locations of erasures in previous slots, it can adapt its transmission strategy to increase λ and hence improve the quality of streaming. This idea is explored in the next chapter where we analyze streaming with feedback.

Chapter 4

Streaming with feedback

In Chapter 3 we analyzed streaming codes without feedback and showed that the quality of streaming can be measured in terms of the rate of growth of playback delay. We proved that in the no-feedback case, the expected playback delay is asymptotically equal to ${}^{1}\!/\!_{\lambda} \cdot \log n$ with $\lambda = D({}^{1}\!/\!_{b}||\rho)$, where b is the fixed available channel bandwidth and ρ is the channel success probability. No other transmission scheme can achieve a higher value of λ.

If the source receives delayed feedback about past erasures, it can use the feedback to alter its future transmission strategy. In this chapter we analyze playback delay for streaming with feedback and show that it also logarithmic growth ${}^{1}\!/\!_{\lambda} \cdot \log n$. Feedback strictly increases λ and thus helps reduce the growth of delay. However the gain vanishes in the limit of infinite bandwidth where both with feedback and no-feedback cases converge to the same value of $\lambda = -\log(1-\rho)$.

First we consider streaming with instantaneous feedback in Section 4.2. We show that a simple ARQ scheme is optimal and we determine the corresponding growth constant λ. In Section 4.3 we consider a model where the source receives error-free feedback, but after a delay of d slots. We present a greedy algorithm for the source to use the feedback to adapt its transmission strategy, and analyze the variation of λ with d. We conjecture that the proposed scheme is optimal, but the proof is a part of ongoing work.

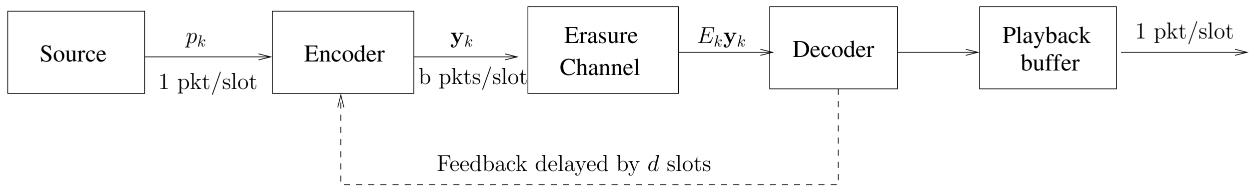


Figure 4-1: The system model for streaming with delayed feedback. In slot k, source can transmit packets based on feedback about channel erasures up to slot $k-d$. If the channel is erased $E_k = 0$, otherwise $E_k = 1$.

4.1 System Model

We use the same system model as described in Chapter 2, with one packet generated per slot at source, and one packet per slot played at the receiver strictly in order. The channel has bandwidth b packets/slot. The b packets are received successfully with probability ρ, otherwise all are erased.

In this chapter, we add a feedback path from the decoder to the encoder. The feedback is error-free, but with a delay of d slots; that is, the source has complete knowledge of erasures up to slot $k-d$ before transmitting in slot k. It can use this information to adapt its transmission strategy in slot k.

A special case of this model is $d = \infty$, the no-feedback scenario analyzed in Chapter 3. In Section 4.2 we analyze the other extreme case, $d = 1$ which corresponds to instantaneous feedback.

4.2 Streaming with instantaneous feedback

Consider a model where the source receives instantaneous feedback about past erasures. For this model, we present the optimal ARQ-based scheme and determine λ, which was defined as the growth constant in the $^1/_\lambda \log n$ asymptotic behavior of expected playback delay.

4.2.1 Streaming ARQ scheme

We propose the simple ARQ-based scheme illustrated in Figure 4-2 for the source to adapt its transmission strategy based on the feedback. In every slot, the source

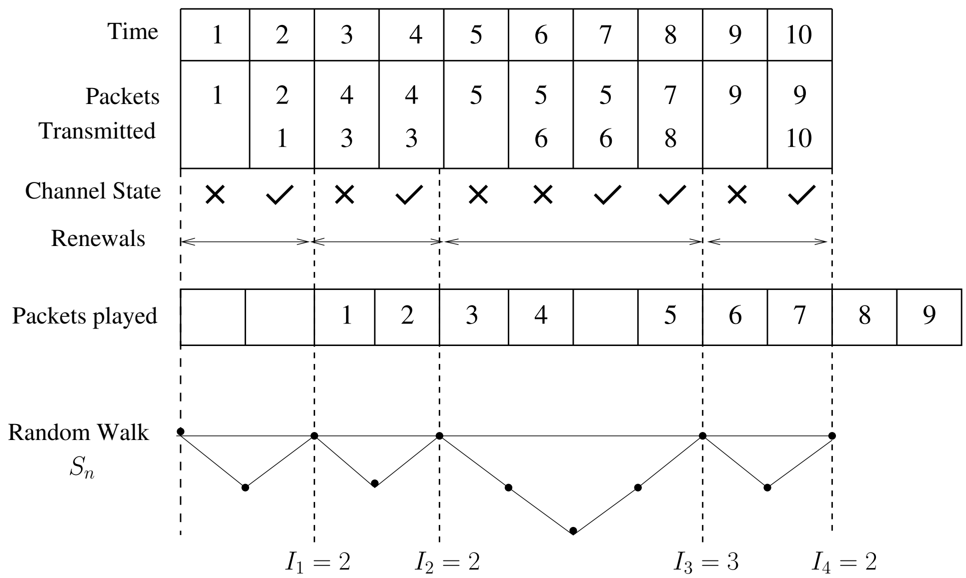


Figure 4-2: The optimal ARQ-based scheme for streaming with instantaneous feedback

transmits the b minimum-index packets that have not been decoded yet. Due to instantaneous feedback, the source can transmit uncoded packets instead of linear combinations like the no-feedback case in Chapter 3. If the channel is erased in that slot, the packets are retransmitted in the next slot. Otherwise, the source moves ahead and transmits the next b packets. Note that in Figure 4-2 the source transmits less than b packets in some slots if more packets have not been generated yet.

This ARQ-based scheme is optimal for streaming with instantaneous feedback because in every unerased slot, b packets become available in the order of playback. This is unlike the no-feedback case, where if b packets were received, they may not be available for playback because they may be unsolved combinations, or be out of order.

The dynamics of the source and receiver buffers can be modeled by considering an equivalent queueing system shown in Figure 4-3. One packet enters the source queue

Figure 4-3: A queuing system modeling the number of packets at the source and receiver for the streaming ARQ scheme.

in every slot, and b packets depart from the queue with probability ρ. The departures from the source queue enter the receiver queue which plays one packet in every slot. There is an interruption in playback when the receiver queue becomes empty.

4.2.2 Analysis of playback delay

We now study the evolution of playback delay with the optimal streaming ARQ scheme and prove the following main theorem about expected playback delay.

Theorem 4.1 (Expected Playback delay with instantaneous feedback)**.** *The expected playback delay of the optimal ARQ-based for streaming with instantaneous feedback satisfies*

$$\mathbb{E}[P_n] = \frac{1}{\lambda_1} \log n + O(\log \log n). \tag{4.1}$$

The growth constant $\lambda_1 = \log(1/\alpha)$ *where* α *is the real positive root of*

$$\frac{\alpha^b - 1}{\alpha - 1} = \frac{1}{\rho}\,, \qquad \alpha \neq 1. \tag{4.2}$$

To prove this theorem we first model the system as a random walk and express playback delay in terms of threshold crossing of the random walk. The proof follows from Lemma 4.1 below and arguments similar to the proof of Lemma 3.3 in Chapter 3.

Random Walk model for the system

We can model the system by a random walk $S_n = X_1 + X_2 + ..X_n$ where X_i's are i.i.d. binary random variables which are $b - 1$ with probability $1 - \rho$ and -1 with probability $1 - \rho$. The sum S_n is the difference between the number of packets decoded at receiver and number of packets generated at source up to time n. Thus, S_n increases by $b - 1$ in every successful slot and decreases by 1 otherwise. Figure 4-2 illustrates this random walk with $b = 2$. The condition $\rho b > 1$ on the bandwidth and channel success probability implies that $\mathbb{E}[X] > 0$; that is, the random walk has a positive drift.

Define a renewal as the instant when the random walk crosses 0 and let R_k as the k^{th} inter-renewal time. A renewal also corresponds to the instant when the source queue becomes empty. Let I_k be the number of packets remaining in the playback queue at the k^{th} renewal instant. The random variables I_k are i.i.d. since each belongs to a different renewal interval. For the first renewal, I_1 is equal to the number of interrupted slots in that interval. For the k^{th} renewal interval, an interruption will occur only if I_k is greater than I_i for all $1 \leq i < k$. Thus, the playback delay P_n of packet p_n is

$$P_n = \max(I_1, I_2, I_3, .. I_K) \tag{4.3}$$

where K is the smallest integer such that $\sum_{k=1}^{K} R_k \geq n$.

Asymptotic behavior of $\Pr(I > t)$

Since we are interested in the expected playback delay $\mathbb{E}[P_n]$ for large n, we analyze the asymptotic behavior of the tail distribution of I and determine its exponential decay rate.

Lemma 4.1. *For the streaming ARQ scheme, the tail distribution of interruption time I in a renewal decays is $Pr(I > t) \doteq \exp(-\lambda_1 t)$ with $\lambda_1 = \log(1/\alpha)$ where α is the real positive root of*

$$\frac{\alpha^b - 1}{\alpha - 1} = \frac{1}{\rho}\ , \qquad \alpha \neq 1. \tag{4.4}$$

Proof. Consider two thresholds 0 and $-t$ such that the random walk stops permanently when it crosses any one of them. A renewal corresponds to crossing threshold 0. Consider the first renewal of the system. The number of packets remaining in the playback queue at the renewal instant, I_1, is also equal to one more than the minimum value attained by the random walk S_n in that renewal interval. Thus, the tail distribution of I equals

$$\Pr(I > t) = \Pr\left(\bigcup_n \{S_n \leq -t\}\right), \tag{4.5}$$

which is the probability that the random walk crosses threshold $-t$ before crossing

0. The Kingman bound [10] is an asymptotically tight bound on this probability. It states that,

$$\Pr\left(\bigcup_n \{S_n < -t\}\right) \doteq e^{rt} \tag{4.6}$$

where r is the negative root of $\gamma(r)$, the semi-invariant moment generating function of X. For the binary random variable X defined above,

$$\gamma(r) = \log\left(\rho e^{r(b-1)} + (1-\rho)e^{-r}\right) \tag{4.7}$$

Replacing $\alpha = e^r$ we get (4.4). Thus, the tail distribution $\Pr(I > t)$ decays with rate $\lambda_1 = \log(1/\alpha)$. $\square$

Using this value of λ_1 we can evaluate upper and lower bounds on the expected value of playback delay in (4.3) by applying the same arguments used in the proof of Lemma 3.3 in Chapter 3.

4.2.3 Achievable region of λ

Thus we have shown that the dominant term in the growth of playback delay with time index n is $1/\lambda \cdot \log n$. We derived the largest value of λ as a function of b and ρ for the no-feedback and instantaneous feedback cases.

For streaming without feedback, the proposed coded repetition scheme achieves $\lambda = \lambda_\infty = D(1/b||\rho)$. With instantaneous feedback, a simple ARQ based scheme achieves $\lambda = \lambda_1 = \log(1/\alpha)$ where α is the real positive root of (4.4). The behavior of λ with bandwidth b is illustrated in Fig. 4-4. As b approaches infinity, both schemes converge to $\log(1/1-\rho)$. However the instantaneous feedback converges at a much faster rate. The area between the two curves in Fig. 4-4 is the achievable region of growth constant $\lambda = \lambda_d$ for streaming with feedback to the source after a delay of d slots. In the next section we propose a coding scheme for streaming with delayed feedback and analyze how λ_d varies as a function of delay d.

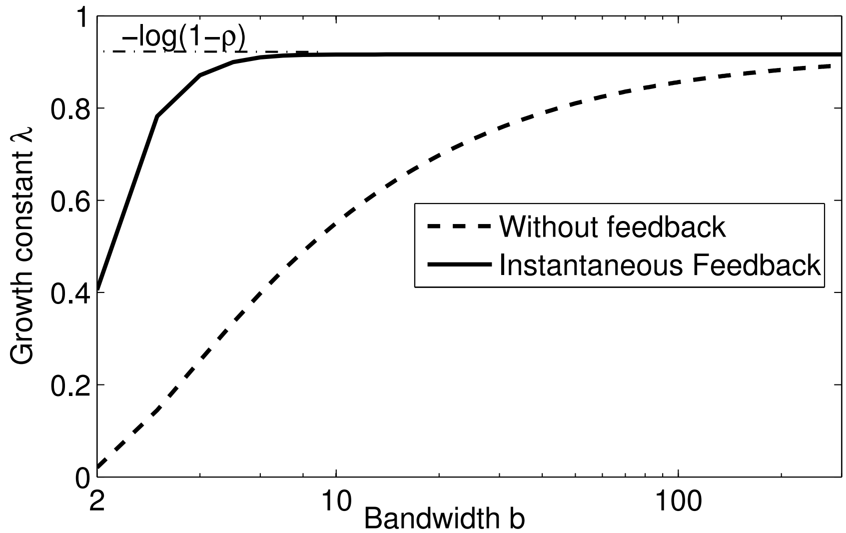


Figure 4-4: Comparison of the behavior of λ with bandwidth b for the no-feedback and instantaneous feedback cases. The success probability $\rho = 0.6$, and logarithms are taken to the natural base.

4.3 Streaming with delayed feedback

We now generalize the analysis of instantaneous feedback to streaming with a feedback delay of d slots. In the instantaneous feedback case the source has complete information about past erasures to adapt its future transmission strategy. Thus, we know that the streaming ARQ scheme described in Section 4.2.1 is optimal in terms of growth of playback delay.

However, finding the optimal scheme is challenging when the feedback is delayed by d slots. This is because there is an window of $d - 1$ slots, $[n - d + 1, n - 1]$ whose erasure pattern is unknown to the source in slot n. The performance of a transmission scheme depends on the assumption that it makes about the unknown erasures, and there is a trade-off between being too optimistic or pessimistic about the unknown erasures.

In Section 4.3.1 we propose a greedy scheme for streaming with delayed feedback which is optimistic about the unknown erasures. In Chapter 3 we showed that the greedy coded repetition scheme is optimal among all time-invariant schemes. We believe that even with feedback, a greedy strategy is at least close to optimal. In

Algorithm 4.1 Greedy coding for streaming with delayed feedback

```
for i = indices of packets in descending order do
  set_of_patterns ← possible erasure patterns of slots [n − d + 1, n − 1] for which
  p_i is not decoded
  keep_in_combn[i] ← false
  while set_of_patterns is not empty do
    if p_i is the least undecoded packet and this combination helps decoded it then
      keep_in_combn[i] ← true
      break from while loop
    else
      Remove current pattern from set_of_patterns
    end if
  end while
end for
Form a combination with p_i, with keep_in_combn[i] = true
```

Section 4.3.3 we analyze the variation in λ_d, the growth constant of playback delay as a function of feedback delay d.

4.3.1 Greedy transmission scheme

In Chapter 3 we optimized coded repetition scheme using Algorithm 3.1. The algorithm includes minimum number of packets in each transmitted combination while ensuring that it is innovative. A packet p_k is added to a combination transmitted in slot n only if there exists some erasure pattern of slots $[1, n-1]$ for which this combination leads to p_k being decoded.

We can extend the same algorithm to get Algorithm 4.1, the greedy coding scheme for streaming with delayed feedback. The only difference is that since the erasure pattern up to slot $n-d$ is already known, the algorithm only has to check channel patterns of slots $[n-d+1, n-1]$ to ensure that the transmitted combination is innovative. This algorithm is optimal among the class of greedy schemes which guarantee innovation because of the constraint that packet p_k added to a combination should be the least index undecoded packet for some channel pattern. This constraint helps older packets get decoded as early as possible thus reducing interruptions in playback.

Figure 4-5 illustrates Algorithm 4.1 for $b = 2$ and $d = 3$. We explain the operation

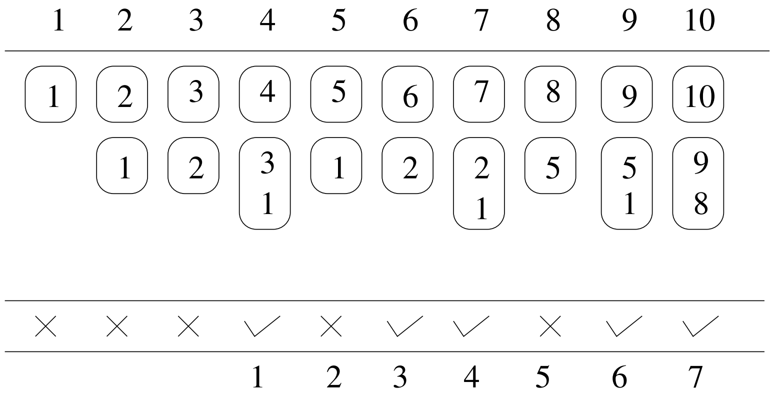


Figure 4-5: Illustration of Algorithm 4.1 for $b = 2$ and $d = 3$. The bottom row denotes the number of past slots for which feedback has been received. For example, before transmitting in slot 6, the source has feedback about erasures up to slot 3.

of the algorithm in slot 5. Consider the first linear combination transmitted in slot 5. If slot 3 and 4 are successful, p_5 is the least undecoded packet for this path. Thus, we keep it in this combination. Adding any other packet in this combination will not make it help decode it because p_5 is being transmitted for the first time. Thus, the first linear combination contains only p_5. Now we find that second linear combination transmitted in slot 5. For all the erasure patterns (0,0), (0,1) and (1,0) of slots 3 and slot 4, p_1 is the least undecoded packet. Thus, it has to be the only packet added to the second linear combination.

The concept of decodability introduced in Chapter 2 plays an important role in streaming with feedback. When the source knows that packet p_k is decodable at the receiver, it can delete p_k from its transmit buffer since it is not necessary to include it in any future linear combinations. Thus, only non-decodable packets are candidates for inclusion in any combination to be transmitted.

4.3.2 Packet decoding in terms of threshold crossing

For the no-feedback case, we expressed packet decoding in terms of time when number of combinations exceeds number of variables. As illustrated in Figure 2-5, decoding occurs when the trajectory of combinations versus variables crosses the slope one line.

This is not true for the greedy scheme for delayed feedback because some packets

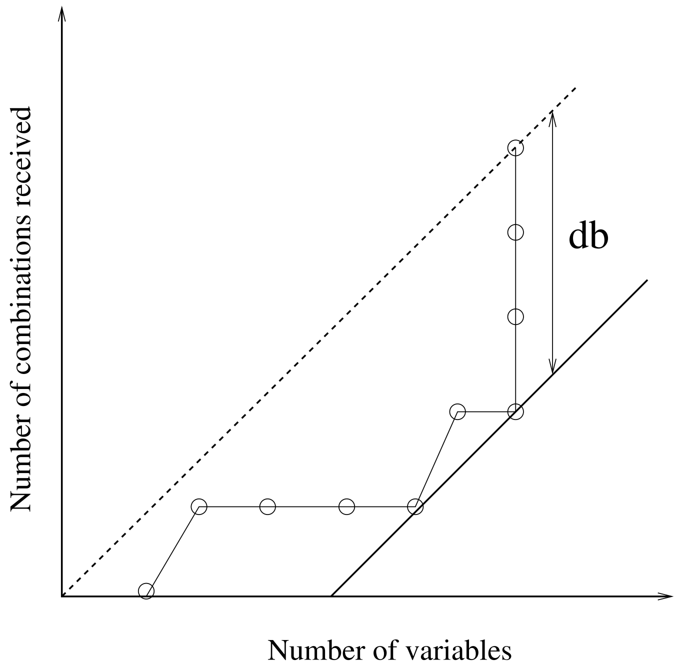


Figure 4-6: Threshold crossing interpretation of packet decoding. Decoding occurs when the trajectory crosses the $y = x$ line. It cannot cross $y = x - db$ boundary db at any point, because the difference between variables and equations can be atmost db.

may be decoded before a renewal takes place. To simplify the analysis, we consider a simplified version of the greedy scheme where packet decoding cannot occur until the total number of combinations exceeds the number of variables in the system. For example, in Figure 4-5 only 6 combinations are received till slot 7, but the number of variables is 7. In the simplified scheme, we assume that no packet is decoded although the greedy scheme all packets except p_5. Packet decoding occurs for the first time only in slot 10. Thus, analysis of packet decoding reduces to counting combinations and variables. The playback delay with the simplified scheme is greater than the actual greedy scheme for every channel realization. Thus, finding λ_d for the simplified scheme gives a lower bound on the growth constant for the actual scheme.

The difference between the delayed feedback and no-feedback cases is that with delayed feedback there can be at most db variables in the system at any time. The intuition is that based on the feedback up to slot $n - d$, the source can avoid adding new variables in slot n. In terms of threshold crossing, an upper limit of db on the number of variables means that we have a boundary db units below the slope one line as shown in Figure 4-6. The trajectory of combinations versus variables cannot to the right side of this boundary. Due to this, packet decoding occurs at a faster rate. In the extreme case of instantaneous feedback where $d = 1$, the boundary is b units below the slope one line. Since b combinations are received in every successful slot, b packets are decoded every time the channel is successful.

We now provide an exact formulation of packet decoding in terms of threshold crossing. Let v_n be the number of new variables added to b transmitted combinations in slot n. Based on the feedback received up to slot $n - d$, let D be the deficit, the difference between the number of variables and combinations received. Assuming that the $d - 1$ unknown slots are successful, the projected deficit D' at the end of $n - 1$ slots is,

$$D' = D - (d-1)b + \sum_{i=1}^{d} v_{n-d+i} \tag{4.8}$$

Thus, the source adds at most $b - D'$ new variables in slot n, provided those many new packets have been generated. Thus the number of variables added in slot n, v_n can be evaluated by using the relation,

$$v_n = \min\left(n - \sum_{i}^{n-1} v_i, \max(b - D', 0)\right) \tag{4.9}$$

A packet decoding instant is defined as the time when deficit D hits zero. Let A_i be time between the $(i-1)^{th}$ and i^{th} decoding instants, and X_i be the number of packets decoded at the i^{th} decoding instant. Then, the plot of S_n, the number of decoded minus number of packets generated versus time n is as shown in Figure 4-7. A renewal occurs whenever $\sum_{i=1}^{m} X_i \geq \sum_{i=1}^{m} A_i$.

4.3.3 Analysis of Playback delay

We now analyze the growth of expected playback delay of the proposed greedy scheme as a function of feedback delay d. Let λ_d be the growth constant of playback delay for a given feedback delay d. We have determined the two extreme values: λ_∞ without feedback in Chapter 3, and λ_1 with instantaneous feedback in Section 4.2. They give

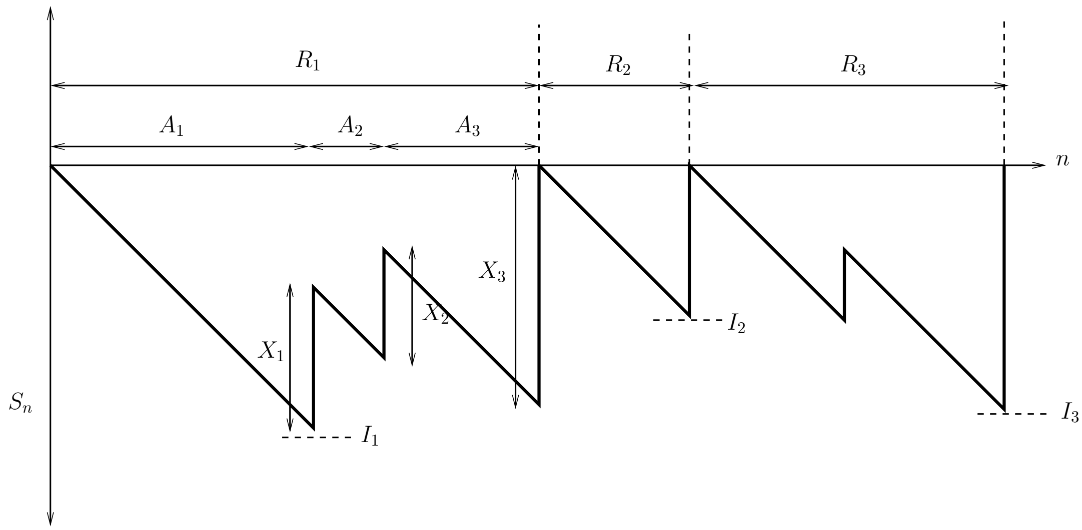


Figure 4-7: Trajectory of S_n the difference between the packets decoded and generated. A_i is the time between the $(i-1)^{th}$ and i^{th} decoding instants, and X_i is the number of packets decoding at the i^{th} instant. A renewal occurs when the trajectory crosses above 0.

lower and upper bounds respectively on λ_d for any d. In this section we determine how λ_d decays from λ_1 to λ_∞ as a function of d.

First, it is easy to show that

$$\lambda_{d_1} \leq \lambda_{d_2} \text{ for all } d_1 \geq d_2.$$

because, a system with a feedback delay of d_2 slots can choose to incorporate the feedback in adapting its transmission strategy only after $d_1 \geq d_2$ slots. Thus, it can achieve growth constant at least λ_{d_1}, the growth constant with a feedback delay of d_1 slots.

Playback Delay in terms of Renewals

Let I_k be the minimum value attained by the trajectory S_n in the k^{th} renewal interval. Then, the playback delay is,

$$P_n = \max(I_1, I_2, \cdots, I_K), \tag{4.10}$$

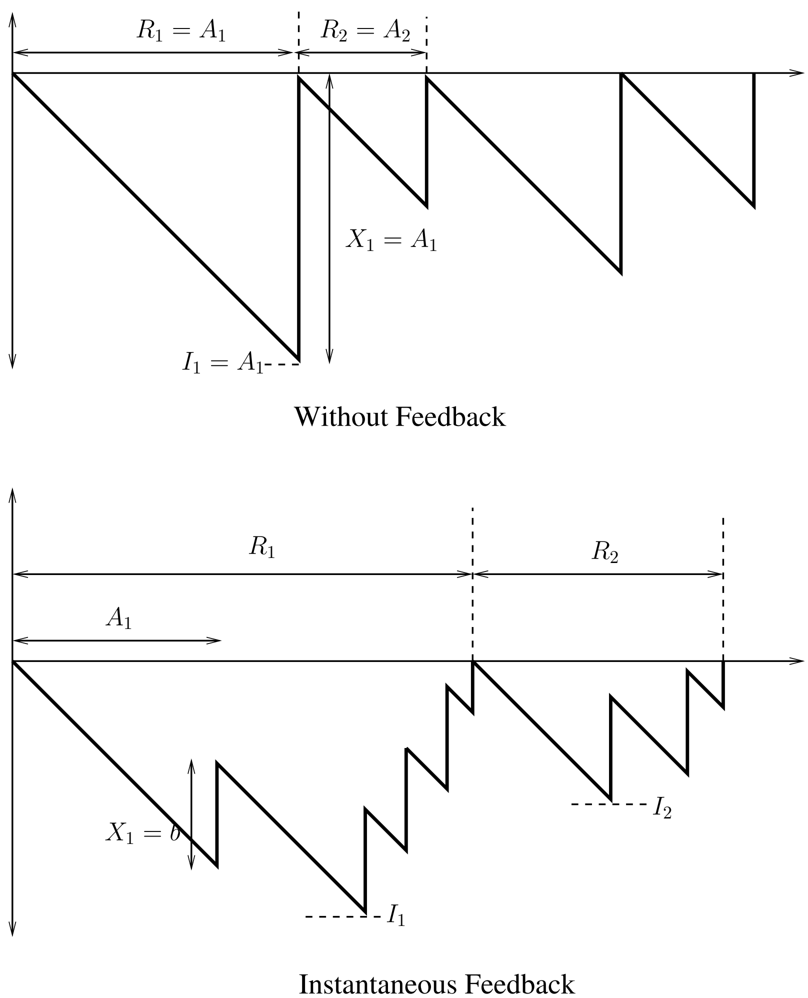


Figure 4-8: The trajectory of S_n for the no-feedback ($d = \infty$) and instantaneous feedback ($d = 1$) cases.

where K is the smallest integer for which $\sum_{k=1}^{K} R_k \geq n$.

The special cases of $d = 1$ and $d = \infty$ are shown in Figure 4-8. We have,

- *Instantaneous feedback* ($d = 1$):

$$Pr(A_i = n) = (1-\rho)^{n-1}\rho \tag{4.11}$$

$$X_i = b \tag{4.12}$$

- *Without feedback* ($d = \infty$):

$$Pr(A_i = n) = \left(1 - \frac{b(\lceil \frac{n}{b} \rceil - 1)}{n-1}\right) \binom{n-1}{\lceil \frac{n}{b} \rceil - 1} \rho^{\lceil \frac{n}{b} \rceil} (1-\rho)^{n - \lceil \frac{n}{b} \rceil} \tag{4.13}$$

$$X_i = A_i \tag{4.14}$$

$$I_i = A_i \tag{4.15}$$

Simulation Results

For general d, it is difficult to evaluate closed form expressions for λ_d. Thus, we use simulations to determine how λ_d varies with feedback delay d. We generate A_i and X_i according to the threshold crossing interpretation of packet decoding described in Section 4.3.2. Using these we construct the trajectory S_n as shown in Figure 4-7 from which we get i.i.d. samples I_k for every renewal R_k. λ_d is the rate of decay of the probability distribution of I_k.

Figure 4-9 gives a plot of λ_d for $1 \leq d \leq 20$. The system parameters are $b = 2$ packets/slot and $\rho = 0.6$. We generate the S_n trajectory for 50000 slots to obtain the empirical probability distribution of I and determine its exponential decay rate λ_d. The extreme values λ_1 and λ_∞ are theoretically computed using the analysis and marked in Figure 4-9. We observe that they match well with the simulation results.

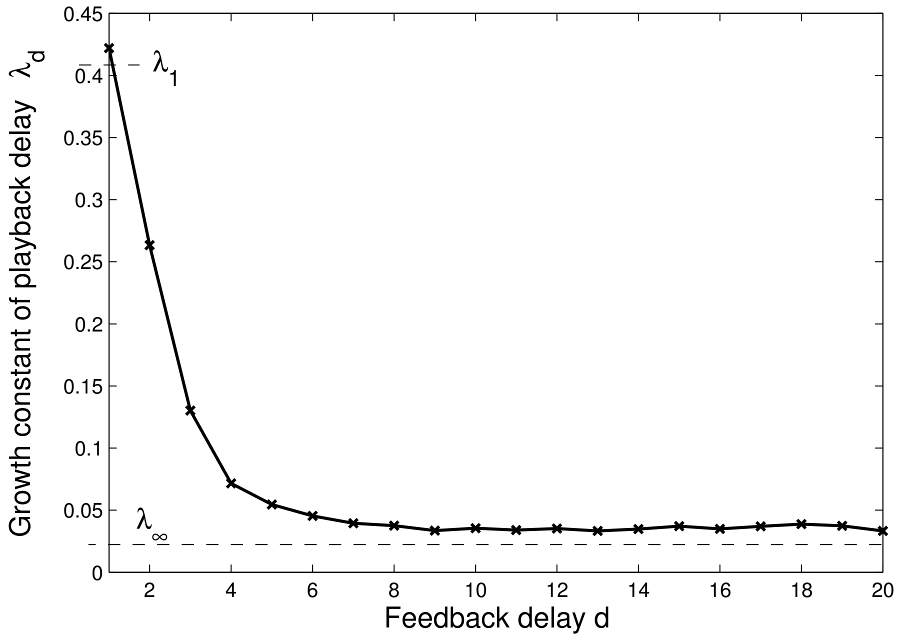


Figure 4-9: Simulation plot of λ_d for feedback delay $1 \leq d \leq 20$ for streaming over a channel with bandwidth $b = 2$ packets/slot and success probability $\rho = 0.6$.

4.4 Conclusions

In this chapter we analyzed streaming with feedback. For the instantaneous feedback case, we propose a simple ARQ-based scheme which is optimal in terms of playback delay. We show that the expected playback delay grows logarithmically with the slot index n and determine the pre-log term in Theorem 4.1.

When the feedback is delayed by d slots, it is difficult to find the optimal scheme and the corresponding growth constant λ_d. We present a greedy scheme which is an extension of the coded repetition scheme presented in Chapter 3. We analyze the playback delay of this scheme and obtain an interesting interpretation in terms of threshold crossing of random walks.

Chapter 5

Streaming Broadcast with instantaneous feedback

5.1 Introduction

So far we considered streaming over a point-to-point channel. However, many practical applications involve the source broadcasting a common stream to a set of users. In this chapter we consider such a broadcast streaming scenario where the source is transmitting packets to N users over independent erasure channels with instantaneous feedback.

The use of network coding for broadcast has been studied in the following previous work. In [14] the problem of minimizing delay in the broadcast scenario with instantaneous feedback is analyzed. The notion of delay used in this paper is the number of coded packets that are successfully received, but do not allow immediate decoding of a source packet. It can be shown that for $N = 2$ users, a simple greedy coding scheme is delay optimal. However, optimality of this scheme has not been proved for $N = 3$ or more users. In [15], the authors propose an algorithm for $N = 3$ and use simulations to show that it achieves asymptotically optimal decoding delay as the ratio of arrival and departure rate from the source queue goes to 1.

Both these papers focus on minimizing the number of coded packets that do not lead to immediate decoding at the receiver. However, for streaming data we have additional order constraints on the playback of packets. Thus, even if a scheme gives minimum delay in terms of immediate decoding, it may not be optimal in terms of playback delay, because packets are not necessarily decoded in order.

Decoding delay is a more natural delay metric than the metric in terms of immediate decoding used in the papers described above. In [11], the authors analyze decoding delay given by the greedy coding scheme for $N = 2$. For the model where the two channels have different erasure probabilities, the authors propose a method to ensure packet decoding over the weak channel.

In contrast to [11], we consider playback delay as the performance metric and analyze the rate of growth of expected playback delay as done in previous chapters. While previous works give coding schemes only for the $N = 2$ and $N = 3$ cases, we can extend the scheme to an arbitrary number of users. The main contribution is an analysis of how the growth constant λ scales with N.

This chapter is organized as follows. In Section 5.2 we describe the system model. In Section 5.3 we propose a greedy scheme to transmit packets to the N users. In Section 5.4 we analyze the playback delay using a random walk interpretation of packet decoding. We use this idea in Section 5.5 to analyze how the playback delay scales with the number of users in the system. Finally Section 5.6 concludes the chapter and provides directions for future work.

5.2 System Model

The system model for the broadcast scenario is as shown in Figure 5-1. It is a direct extension of the model used in the previous chapters. As considered earlier, one packet per slot is generated at the source, and each user plays one packet per slot strictly in order. In every slot, the source uses the packets generated so far to create b combinations. It broadcasts these combinations to N users over independent erasure channels with same success probability ρ. As assumed earlier, the condition $\rho b > 1$ is necessary to ensure that we are operating below the capacity of the erasure channel. The source receives instantaneous and error-free feedback about erasures on each of

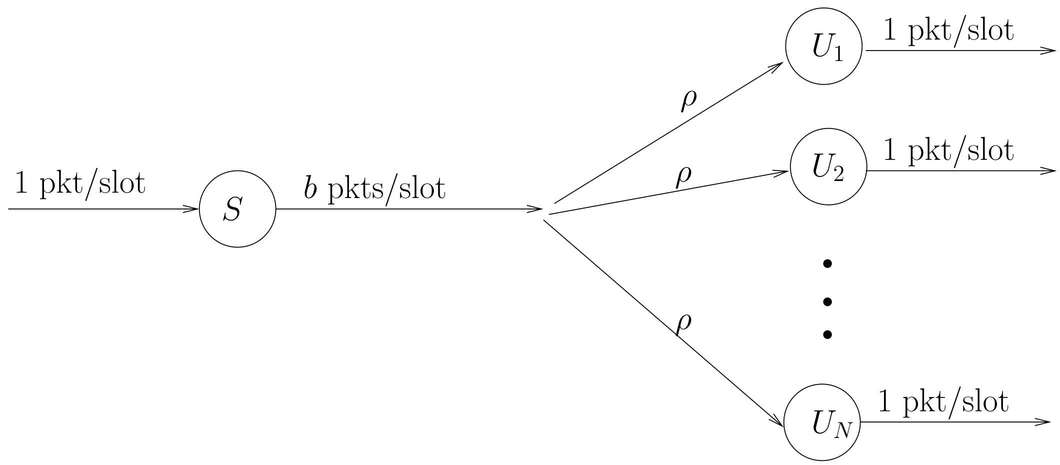


Figure 5-1: System Model for broadcast streaming with instantaneous feedback. In every slot, the source transmits copies of b encoded packets to each of the N users over independent erasure channels with success probability ρ.

the channels.

If the channels have different erasures probabilities we can have two main types of transmission strategies. If the source transmits packets greedily, users with strong channels will get priority and the playback of users with weak channels will grow faster than $O(\log n)$ with time slot n. On the other hand, if the source gives equal priority to weak channels, the rate of growth of playback delay for all users will be governed by the success probability of the weakest channel. Thus, in this case we need a suitable delay metric which takes the different channel qualities into account. The optimal transmission scheme that minimizes this metric should achieves a trade-off between these extreme cases described above. In this chapter, we assume equal success probabilities on all channels so that the expected playback delay is same for all users and it can be used as the delay metric.

5.3 Proposed coding scheme

At any given time, each user has decoded a different subset of the packet stream based on the erasures on the channel to that user. Thus, the transmission scheme needs to combine packets in such a way that each user decodes its required packets.

In this section we propose a greedy coding scheme to achieve low playback delay at all users. First, we introduce a notion of rank of a user to represent which packet it requires for playback. The proposed coding scheme uses the ranks of users to determine which packets need to be combined in each coded packet.

5.3.1 Assigning Ranks to users

Recall the notion of decodability that we defined in Chapter 2: a packet p_k is said to be decodable when the user can construct a linear combination containing only packets p_1 through p_k. In other words, packet p_k can be decoded if all previous packets are decoded. We use this notion of decodability to define the rank of every user served by the source.

The rank of a user is defined as the index of the oldest non-decodable packet. We refer to the packet with index equal to rank of user U_i as the 'required' packet for user U_i. The ranks of users are updated after forming every combination. For example, before constructing the j^{th} combination of slot n, $(1 \leq j \leq b)$ the ranks of users are evaluated using the feedback about erasures till slot $n-1$, and assuming that combinations 1 through $j-1$ in slot n are received successfully. For example, in Figure 5-2, before forming the second combination in slot 3 the ranks of the users are 4, 2 and 1 respectively.

Suppose the ranks of the users take K distinct values, r_k's $1 \leq k \leq K$ which are arranged in descending order, that is $r_k > r_j$ for all $k < j$. We divide the users into classes $\mathcal{S}_k$, $1 \leq k \leq K$ where users in class $\mathcal{S}_k$ have to k^{th} highest rank, r_k. A user is included in exactly one set, but the converse is not true. The users in set $\mathcal{S}_1$ with the highest rank r_1 are referred to as the leaders. For the example shown in Figure 5-2, users U_3 is the leader after slot 4.

5.3.2 Greedy coding scheme

We propose a coding scheme that minimizes the number of packets in each combination while guaranteeing innovation in every slot. The main idea is that source tries to include each user's required packet to the combination. Thus, it transmits a com-

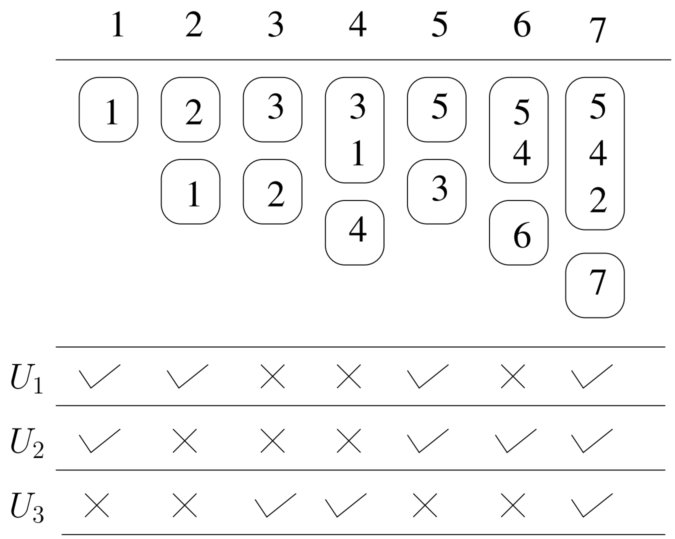


Figure 5-2: Reduced coding scheme for $b = 2$ and $N = 3$ where the minimum number of packets are included in every combination while ensuring that every combination is innovative.

bination of packets p_{r_k} for all $1 \leq k \leq K$. If the channel is successful in that slot, users in set $\mathcal{S}_1$, the leaders, will decode the required packet p_{r_1}. A user in $\mathcal{S}_j$ decodes p_{r_j} only if it has decoded the packets p_{r_k} for all $k < j$.

We can minimize the number of packets included in every combination in a manner similar to Section 3.5. For every $1 \leq j \leq K$, packet p_{r_j} is included in the combination only if at least one user in $\mathcal{S}_j$ has decoded packets p_{r_k} for all $k < j$. This reduced scheme is illustrated in Figure 5-2. Algorithm 5.1 gives a formal statement of this transmission strategy in every time slot. The output of the algorithm is *combn_to_send* which is a vector of the indices of packets to be combined by the source encoder and transmitted over the broadcast channel. Note that the reduced version is equivalent to the original scheme in the sense that it gives exactly same playback delay for every packet at each user.

5.4 Analysis of Playback Delay

In this section we express packet decoding at each user in terms of threshold crossings of random walk and use it to analyze the playback delay. In general, the decoding of

Algorithm 5.1 Reduced coding scheme for broadcast streaming

```
combn_to_send ← []
for i = k → K do
    if Packet with index r_k has been generated at the source then
        if At least one user in S_k can decode packet p_{r_k} on adding it to this combination then
            Append r_k to combn_to_send
        end if
    end if
end for
Create a linear combination of packets with indices combn_to_send
```

packets at a user in every slot depends on its rank r_k and the set of packets p_j, $k > r_i$ that are already decodable. Since the rank of a user and the number of distinct ranks changes after every time slot, it is difficult to exactly determine the distribution of decoding delay of a packet. We simplify this analysis by considering a scheme that gives an upper bound on the playback delay of the proposed greedy coding scheme.

5.4.1 Simplified greedy scheme

Instead of the greedy coding scheme, we analyze a simplified greedy scheme in which packet decoding occurs exactly when the number of combinations received exceeds number of unknowns. In the actual greedy scheme, some packets may be decoded earlier. Hence, the playback delay of every packet is lower with the greedy scheme as compared to its simplified version.

In the simplified scheme, the source transmits b combinations of all packets p_k, $r_K \le k \le r_1$, where r_1 and r_K are the maximum and minimum ranks among all users. Recall that the greedy scheme transmitted a linear combination of only K packets, p_{r_k} for $1 \le k \le K$. Thus, the simplified scheme is still greedy and ensures innovation, but adding more packets to each combination delays the decoding of every packet.

5.4.2 Packet decoding with the simplified scheme

Packet decoding for the simplified greedy scheme is easy to analyze because it reduces to counting number of combinations (equations) received and number of packets (variables) included in those combinations. Decoding occurs at a given user when the number of equations exceeds that number of variables in the system. In every slot a user receives b equations of source packets. Since the algorithm is greedy, it adds new variables in slot n whenever the leader(s) decodes the packets transmitted up to slot $n-1$. Thus, the total number of variables added upto slot n is equal to number of packets decoded by the leader at the end of $n-1$ slots, plus up to b new variables added in slot n.

Let $E_i[n]$ be the total number of equations received at user U_i in n slots,

$$E_i[n] = \min(n, E_i[n-1] + b \cdot \mathbb{1}[Z_i[n] = 1]) \tag{5.1}$$

where $Z_i[n]$ a binary random variable representing the state of the erasure channel to U_i at time n. $\mathbb{1}[A]$ is the indicator random variable with takes value 1 is event A occurs and 0 otherwise. The initial condition for the recursion is $E_i[n] = 0$ for all $1 \leq i \leq N$. In every slot the number of equations received increases by b with probability ρ. However, the total $E_i[n]$ cannot exceed n because only n packets have been generated at that time. We define $E^*[n]$ as the maximum number of equations received in every slot as follows

$$E^*[n] = \max_{1 \leq i \leq N} E_i[n] \tag{5.2}$$

The total number of variables in the system after n slots is given by

$$V[n] = \min(n, E^*[n-1] + b) \tag{5.3}$$

because the source can add b new variables to $E^*[n-1]$, the number of equations received by the leader(s) until slot $n-1$. We set the initial condition $V[0] = 0$. Packet decoding at user U_i occurs in slot n if $E_i[n] \geq V[n]$. From (5.1)-(5.3) we can see that this happens only when $E_i[n-1] = E^*[n-1] < n-1$ and $Z_i[n] = 1$, that is the channel to U_i is not erased in state n. Figure 5-3 illustrates the evolution of $E_i[n]$ and $V[n]$ for $N = 3$ users.

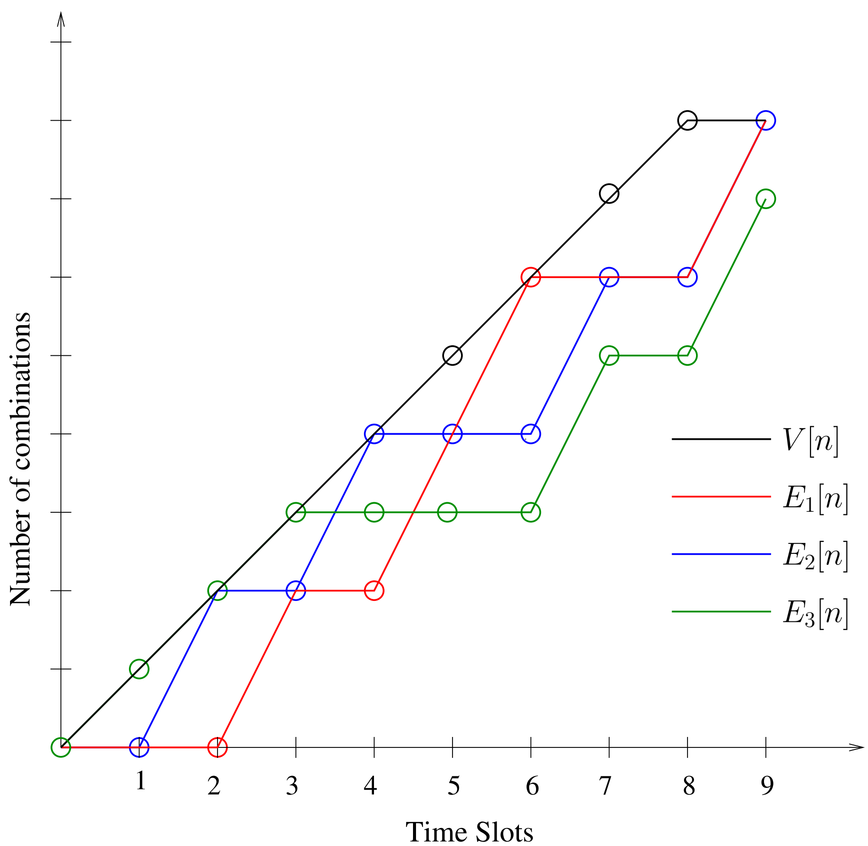


Figure 5-3: Evolution of $E_i[n]$ and $E^*[n]$ for $N = 3$ users. User U_i decodes all transmitted packets in slot n if $E_i[n] = V[n]$. In this example, U_1 decodes in slot 6 and 9, U_2 decodes in slots 2, 4, and 9 and U_3 in slots 1, 2, and 3.

5.4.3 Playback Delay

We use the above analysis of packet decoding to construct the random walk S_n as described in Section 4.3 in the previous chapter for a particular user. This trajectory is then used to analyze the playback delay for that user.

Let A_j be the time between the $(j-1)^{th}$ and j^{th} decoding instants and X_j be the number of packets decoded at the j^{th} instant. We can evaluate these from the trajectories $E_i[n]$ for $1 \leq i \leq N$ and $V[n]$. Thus, for every user we can plot S_n, the trajectory of the number of packets generated at source minus the packets decoded in order at the receiver as shown in Figure 4-7. The analysis of playback delay is same as Section 4.3 where we defined that a renewal occurs whenever S_n crosses above 0. The expected playback delay is,

$$\mathbb{E}[P_n] = \max(I_1,\ I_2,\ \cdots,I_k) \tag{5.4}$$

where I_m is the minimum value attained by the random walk S_n in the m^{th} renewal interval, and k is smallest number such that $\sum_{m=1}^{k} R_m \geq n$.

5.5 Scaling of delay with the number of users

In this section we analyze how λ_N, the growth constant of playback delay, scales with the number of users N. We derive exact expressions for the extreme cases $N = 1$ and $N = \infty$. For arbitrary N we present a numerical bound on λ_N.

5.5.1 Single user, $N = 1$

The single user case, $N = 1$, corresponds to the point-to-point streaming with instantaneous feedback analyzed in Section 4.2. For $N = 1$, the greedy scheme is equivalent to the streaming ARQ scheme presented in Section 4.2.1. The state S_n of the system is the random walk $S_n = Z_1 + Z_2 + \cdots + Z_n$, where Z_n are i.i.d and taking value $b-1$ with probability ρ and -1 with probability $1-\rho$. The random variables $X_i = b$ and $\Pr(A_i = n) = (1-\rho)^{n-1}\rho$. As proved in Theorem 4.1, we can apply the Kingman bound to this random walk and determine the growth constant of playback delay $\lambda_1 = \log(1/\alpha)$ where α is the real positive root of

$$\frac{\alpha^b - 1}{\alpha - 1} = \frac{1}{\rho}\ , \qquad \alpha \neq 1.$$

5.5.2 Infinite users, $N = \infty$

Now consider the case where infinite number of users are being served by the source. We can determine a closed expression for the growth constant λ_∞ of playback delay for this case. For $N = \infty$, the simplified greedy scheme described in Section 5.4.1 becomes equivalent to the coded repetition scheme proved to be optimal for streaming without feedback in Chapter 3. This is because in (5.3), $E^*(n-1)$ is always $n-1$ because among the infinite number of users, there exists with probablity one a user which has not experienced any erasure until slot $n-1$. Thus, $V[n] = n$ for all n and $E_i[n]$ increases by b with probability ρ in every slot which is exactly the evolution of variables and equations in the without feedback case. Thus, as shown by Theorem 3.1 the growth constant of playback delay is,

$$\lambda_\infty = \lambda_c = D\left(\frac{1}{b}||\rho\right)$$

In addition, the expected ordered decoding delay $\mathbb{E}[C_k]$ is as derived in Section 3.3.

5.5.3 Arbitrary number of users N

Unlike the extreme cases $N = 1$ and $N = \infty$, it is difficult to find a closed form expression for the growth constant λ_N for an arbitrary number of users. We use Monte Carlo simulations to study the variation of λ_N with N. We generate erasure patterns of the N channels, and use the trajectories $E_i[n]$ resulting from these patterns to evaluate A_i and X_i as described above. Using these we construct the trajectory S_n as shown in Figure 4-7 from which we get i.i.d. samples I_k for every renewal R_k. λ_N is the rate of decay of the probability distribution of I_k.

Figure 5-4 shows a simulation plot of λ_N versus the number of users N with

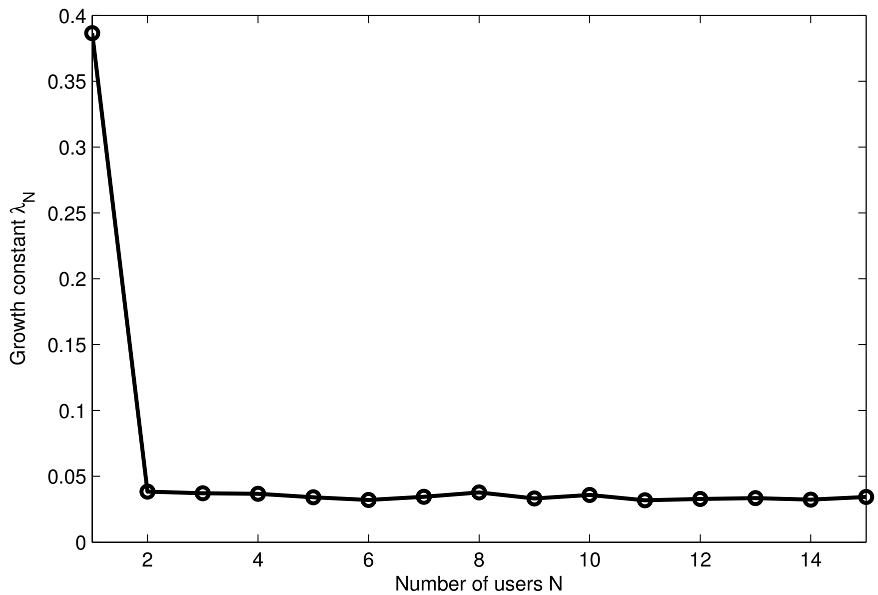


Figure 5-4: Simulation plot of λ_N for number of users $1 \leq N \leq 15$ for streaming over a channel with bandwidth $b = 2$ packets/slot and success probability $\rho = 0.6$.

channel bandwidth $b = 2$ packets/slot and success probability $\rho = 0.6$. It shows a sharp decrease in the growth constant from $N = 1$ to $N = 2$ or more users. From this we observe that the delay increases drastically when we stream to more than one users. Also, the curve of λ_N is essentially flat for all values $N \geq 2$. This indicates that the number of users N served by the source does not affect λ_N for $N \geq 2$.

5.6 Summary

In this chapter we extended the analysis of delay in point-to-point streaming to a broadcast scenario where the source transmits a common packet stream to N users over independent erasure channels with instantaneous feedback. Since at any given time, each users has decoded a different subset of the source packets, the source has to combine packets in such a way that every user decodes its required packets.

We proposed a greedy coding scheme in which the source transmits b linear combinations of the required packets for each user. We used the idea of modeling packet decoding in terms of threshold crossing of a random walk to show that expected playback delay grows logarithmically and determine the growth constant λ_N in terms of the number of users N.

Future work includes proving whether the proposed greedy scheme is optimal. First we aim to prove its optimality among schemes that guarantee innovation in every slot, and then show that there is no gain in the growth constant by using not sending innovative packets in every slot. Although, we are able to reduce the analysis of playback delay to a random wal k threshold crossing problem, it is difficult to determine an exact expression for λ_N. We aim to at least get an approximation for λ_N in terms of N, b and ρ.

Chapter 6

Conclusions and Future Work

6.1 Major implications

In this thesis we addressed the problem of designing optimal codes for packet streaming over an erasure channel. These codes are relevant to a wide-range of audio/video applications that impose delay constraints on packets. Design of optimal codes is a challenging problem because, when the available bandwidth is limited and the source received delayed or no feedback about past erasures, there is a trade-off between transmitting new packets and retransmitting old ones.

Previous work on codes with delay-constraints optimize decoding delay. However, these codes are not necessarily optimal for applications such as live streaming and remote desktop that require in-order playback at the receiver, immediately after packets are decoded. Our work fills this gap and proposes codes that are optimal in terms of playback delay.

One major implication of this work is to define a suitable notion of delay to compare streaming codes. We analyzed three streaming scenarios: without feedback, delayed feedback and broadcast with instantaneous feedback. We showed that in all cases the expected playback delay is asymptotically equal to $^1/_\lambda \log n$. The pre-log term λ is referred to as the growth constant. We used this quantity as the metric of interest and design codes with the objective of maximizing λ.

The second main contribution of this thesis is that we proved the optimality of simple greedy coding schemes in the no feedback and instantaneous feedback cases and determined the corresponding values of growth constant λ. The growth constant with feedback is strictly better that the one without, but they have the same asymptotic value in the limit of infinite bandwidth. By this analysis, we have found the limits on delay in streaming over a channel with any finite feedback delay.

A direct application of our analysis of playback delay is to help system designers estimate the size of the source and receiver buffers required to ensure that packets are not dropped due to buffer overflow.

6.2 Summary of results

We described the system model and introduced the concept of renewals in packet decoding in Chapter 2. Modeling renewals in terms of threshold crossing of a random walk is the main tool used for analysis of delay in the subsequent chapters.

We studied the no-feedback case in Chapter 3 and showed that the optimal value is $\lambda = D(1/b||\rho)$ where b is the bandwidth in packets per slot and ρ is the success probability of the erasure channel. We proved that the simple coded repetition scheme where in every slot the source transmits combinations all packets generated so far achieves the largest λ among the class of time-invariant schemes.

In presence of feedback, the source can adapt its transmission strategy based on past erasures. We proposed greedy coding scheme and analyzed playback delay for streaming with feedback in Chapter 4. With instantaneous feedback, the ARQ scheme is optimal and we can determine the exact expression for λ. For the delayed feedback case we determined a lower bound on λ_d as a function of feedback delay d.

Finally, we extended the analysis to a broadcast streaming scenario with instantaneous feedback where the source is transmitting a common packet stream to N users over independent erasure channels. We proposed a greedy coding scheme and analyzed its playback delay by modeling packet decoding as threshold crossing of a random walk. Using this analysis we determined how the growth constant λ_N scales with the number of the users N.

6.3 Future perspectives

We have shown that greedy coding is optimal for the without feedback and instantaneous feedback cases. However we have not yet proved its optimality for the delayed feedback and broadcast streaming. This is a major part of ongoing research efforts. The first step is to prove that the proposed scheme is optimal among the class of schemes that guarantee innovation in every slot. Then we need to prove there is no further reduction in delay by using a non-innovative packet transmission scheme.

An immediate research direction is to extend the results of this thesis to streaming over packet networks. In [6], the authors have shown that greedy coding where every node in the network transmits combinations all available packets is capacity-achieving for unicast or multicast over lossy packet networks. However, delay performance of such codes has not been analyzed. A scheme based on fountain coding to minimize decoding delay over line networks is proposed in [16]. These codes may not be optimal for streaming applications which require playback at the receiver.

Although we have only considered the i.i.d erasure channel in this thesis, it is possible to generalize the results to other channel models. In Section 3.6 we showed that the expected playback delay has the same logarithmic growth even for certain channels with memory such as the two-state Markov erasure channel. One could also analyze streaming with lossy feedback to the source in contrast to the error-free feedback assumed in this thesis.

There are several interesting open problems in the broadcast streaming scenario. Firstly, we aim to get a better characterization of the decay of λ_N with the number of users N. From a system design perspective, it would be useful to determine the required increase in bandwidth as N grows. Another research direction is to consider different priority classes among users. In this cases, there will be an achievable region of growth constants unlike the same λ for all users when all users have equal priority. Even for the simple case $N = 2$, it would be interesting to analyze this trade-off between the growth constants of the two users.

In this thesis we used the idea of expressing packet decoding in terms of threshold crossings of a random walk renewals of a stochastic process to study the behavior of playback delay. From a broader perspective, The random walk simply represents the evolution of the information asymmetry between two parties that are communicating over a lossy medium. A renewal occurs when the asymmetry reduces to zero. Thus, it can be a novel analysis tool useful in variety of applications beyond packet streaming. For example, in financial setting, the random walk could represent the evolution of the uncertainty in predicting a stock price in the future, when we are receiving information about it in every time instant.

Appendix A

Standard Results Used

Theorem A.1 (Generalized Ballot Theorem). *Let ξ_j, $1 \leq j \leq n$ be i.i.d. non-negative integer valued random variables, let $S_k = \xi_1 + \xi_2 + .. + \xi_k$ and let $G = \{S_j < j\, for\, 1 \leq j \leq n\}$. Then,*

$$P(G|S_n) = \left(1 - \frac{S_n}{n}\right)^+ \tag{A.1}$$

Theorem A.2 (Strong Law of Renewal Processes). *For a renewal process with mean inter-renewal time $\mathbb{E}[R] < \infty$, the number of renewals $X(n)$ up to time n satisfies*

$$\lim_{n\to\infty} \frac{X(n)}{n} = \frac{1}{\mathbb{E}[R]} \tag{A.2}$$

Theorem A.3 (Kingman Bound). *Let $S_n = \sum_{j=1}^{n} X_j$ be a random walk with X_j are i.i.d with $\mathbb{E}[X] < 0$. For thresholds $\alpha < 0$, $\beta > 0$ such that the random walk stops permanently after crossing either of them. The probability that the random walk crosses threshold α before crossing β is,*

$$\Pr\left(\bigcup_n \{S_n < \alpha\}\right) \leq e^{r^*\alpha} \tag{A.3}$$

where r^* is the largest positive root of the log-MGF of X given by $\log \mathbb{E}[e^{rX}]$.

Theorem A.4 (Wald's identity). *Let $\{X_i; i \geq 1\}$ be IID, and let $\gamma(r) = \mathbb{E}(e^{rX})$ be the semi-invariant moment generating function of each X_i. Assume $\gamma(r)$ is finite in* *the open interval* (r^-, r^+) *with* $r^- < 0 < r^+$. *For each* $n \geq 1$, *let* $S_n = X_1 + X_2 + ..X_n$. *Let* J *be the smallest* n *for which either* $S_n \geq \alpha$ *or* $S_n \leq \beta$. *Then for* $r \in (r^-, r^+)$,

$$\mathbb{E}(e^{(rS_J - J\gamma(r))}) = 1 \tag{A.4}$$

Theorem A.5 (Expected maximum of geometric random variables)**.** *Let* M_n *be the maximum of* n *i.i.d. geometric random variables with mean* $1/p$ *where* $1 - p = e^{-\lambda}$. *Then, the expected value of* M_n *satisfies*

$$\frac{1}{\lambda}\sum_{k=1}^{n}\frac{1}{k} \leq \mathbb{E}[M_n] \leq 1 + \frac{1}{\lambda}\sum_{k=1}^{n}\frac{1}{k} \tag{A.5}$$

Theorem A.6 (Pythagoras theorem for distributions)**.** *Let* $\mathbf{p}$ *be a probability distribution of* X, *and* $\mathcal{Q}_\gamma$ *be a family of distributions* $\mathbf{q}$ *such that* $\mathbb{E}_{\mathbf{q}}[X] = \gamma$. *Then the distribution* $\mathbf{q}^*$ *in* $\mathcal{Q}_\gamma$ *that minimizes* $D(\mathbf{q}||\mathbf{p})$ *satisfies,*

$$q(x) = p(x) \cdot \exp(-rx) \textit{ for all } x \tag{A.6}$$

and for some scalar parameter r.

Appendix B

Proof of Theorem 3.2

We compare the coded repetition scheme with other time-invariant schemes by introducing the concept of renewal epochs in Section B.1. The proof of Theorem 3.2 follows from Lemma B.1 and Lemma B.2 for two types of renewals epochs, presented in Section B.3. For simplicity of notation, we refer to the coded repetition scheme as Scheme 1, and any other time-invariant scheme as Scheme 2.

B.1 Renewal Epoch

From the definition of time-invariant schemes, it is clear that for every channel realization, the number of combinations received with Scheme 1 is greater than or equal to that with Scheme 2, at every time slot. As a result, for every renewal of Scheme 2 there are always one or more renewals of Scheme 1. We define the interval between

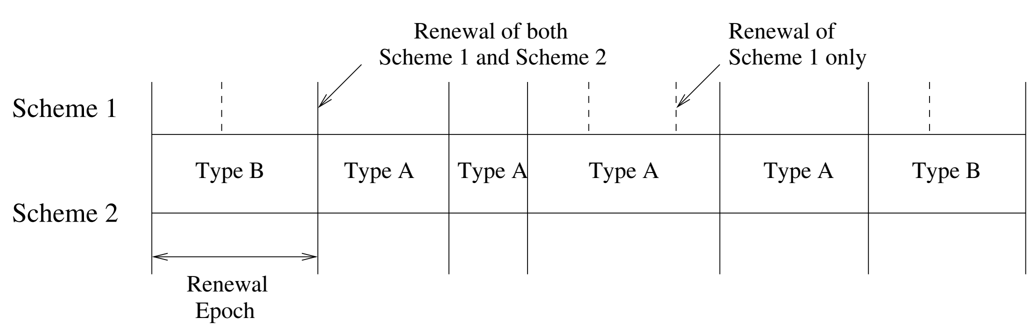


Figure B-1: Illustration of Type A and Type B renewal epochs

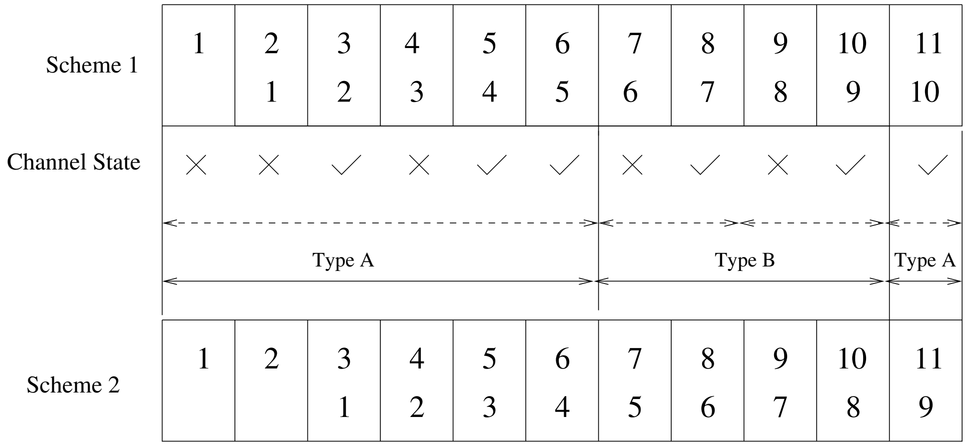


Figure B-2: Example of channel patterns that cause Type A and Type B renewal epochs. Dotted lines indicate renewals of Scheme 1 and solid lines indicate renewals of Scheme 2

two successive renewals of Scheme 2 as a renewal epoch. Renewal epochs can be classified into two types:

1. **Type A renewal epochs:** Epochs in which there is only one renewal of Scheme 1 for a renewal of Scheme 2. A channel pattern in which the first b slots are erased gives rise to this type of renewal epoch.

2. **Type B renewal epochs:** Epochs in which there are two or more renewals of Scheme 1 for every renewal of Scheme 2. A channel pattern in which at least one of the first b slots is not erased gives rise to this type of renewal epoch.

Thus, we can divide the time axis into renewal epochs as shown in Figure B-1. Figure B-2 illustrates the difference between Type A and Type B epochs for and the channel patterns that that cause them.

B.2 Analogy to a path-paving problem

Suppose there are M_K renewal epochs in a window of K slots. Let S_m be the sum of the decodable delays of packets in the m^{th} renewal epoch. The time-averaged decodable delay is given by,

$$\lim_{K\to\infty} \frac{\sum_{k=1}^{K} D_k}{K} = \lim_{K\to\infty} \frac{\sum_{m=1}^{M_K} S_m}{K} \tag{B.1}$$

If a combination y_k is received when packet p_k is already decodable, it makes packet p_j decodable where $j < k$ is the largest packet index such that p_j is not decodable. This is illustrated in Figure B-3a where the length of arrow indicate the decodable delay of the packet it points to.

We can map this decoding process to an equivalent problem of paving a path with tiles. Consider a path of with n gaps corresponding to a renewal epoch. Our objective is to place tiles on each gap in the path. If the k^{th} gap has been filled by a tile, it implies that packet p_k is decodable. A renewal occurs at time n when the entire path from 1 is n is paved with tiles at time n. When the channel is good in the k^{th} slot, b tiles are generated at point k on the path. Extra tiles are moved backward to fill empty gaps in the path upto point k. For example, the tile paving equivalents of the channel realizations in Figure B-3a and Figure B-3b are shown in Figure B-4a and Figure B-4b respectively. The sum of the decodable delays, S_m is the total backward distance moved by tiles in an epoch.

B.3 Comparison of S_m for a renewal epoch

Now we present two lemmas that prove that for every channel realization, S_m is minimum with the coded repetition scheme for all m. Theorem 3.2 follows from this property of S_m.

Lemma B.1. *For Type A renewal epochs, S_m for Scheme* 1 *and Scheme* 2 *are equal.*

Proof. The sum of decodable delays S_m for a renewal epoch is independent of the order in which gaps are filled. If a tile 1 moves a distance d_1 from slot n_1 to $n_1 - d_1$, and tile 2 moves distance d_2 from n_2 to $n_2 - d_2$ such that $n_1 > n_2 > n_1 - d_1 > n_2 - d_2$. The total backward distance is $d_1 + d_2$. Even if we exchange the destinations of the tiles, the total distance $d_1 + d_2$ remains unaffected. For Type A epochs, Scheme 1

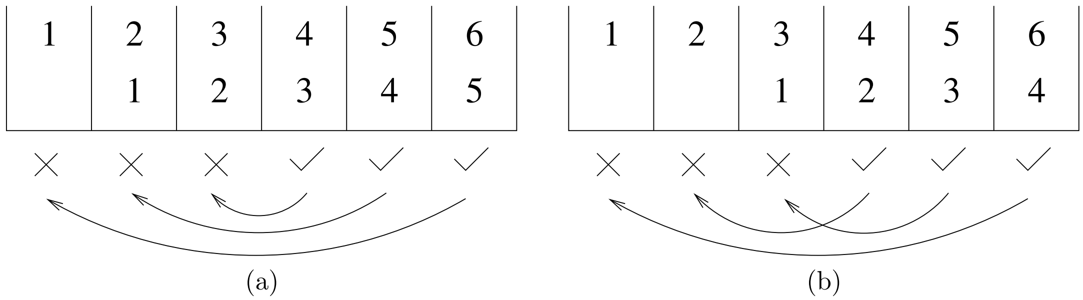


Figure B-3: Illustration of decodable delays of packets in a renewal epoch

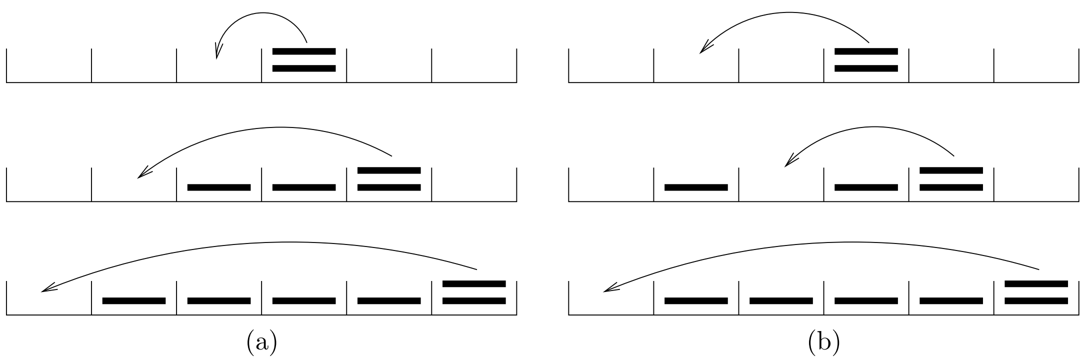


Figure B-4: Analogy to a problem of paving a path with tiles

and Scheme 2 receive the same number of tiles and differ only in the order of filling the gaps. Hence the S_m is equal in Scheme 1 and Scheme 2. □

Lemma B.2. *For Type B renewal epochs, S_m with Scheme 1 is strictly less than that with Scheme* 2.

Proof. For Type B renewal epochs, Scheme 1 at least one equation more than Scheme 2. If the extra tiles are not used to fill any gap, S_m will be the same in both schemes. Suppose the extra tile is at slot i, and there is a gap in the part of the path $[1, i-1]$ which is currently filled by a tile received in slot j, $j > i$. Filling such a gap with the extra tile instead will strictly reduce the total distance. Then, the tile at slot j which was previously used becomes an extra tile. The same process can then be repeated to fill a gap in $[1, j-1]$ with a tile at k where $k > j$. Hence for a Type B epoch, the sum of decodable delays is strictly less with Scheme 1. □

The proof of Theorem 3.2 by applying Lemma B.1 and Lemma B.2 to S_m in (B.1).

Bibliography

[1] E. Berkelamp, *Algebraic coding theory.* New York, USA: McGraw-Hill, 1968.

[2] M. Luby, M. Mitzenmacher, A. Shokrollahi, D. Spielman, and V. Stemann, "Practical loss-resilient codes," in *ACM symposium on Theory of computing*, (New York, NY, USA), pp. 150–159, ACM, 1997.

[3] M. Luby, "LT Codes," in *ACM Symposium on Foundations of Computer Science*, (Washington, DC, USA), IEEE Computer Society, 2002.

[4] A. Shokrollahi, "Raptor codes," *IEEE/ACM Transactions on Networking*, vol. 14, pp. 2551–2567, June 2006.

[5] E. Martinian, *Dynamic Information and Constraints in Source and Channel Coding.* PhD thesis, MIT, Cambridge , USA, Sept. 2004.

[6] D. Lun, M. Médard, R. Koetter, and M. Effros, "On coding for reliable communication over packet networks," *Physical Communication*, vol. 1, Mar. 2006.

[7] J. Sundararajan, D. Shah and M. Médard, "ARQ for Network Coding," in *International Symposium on Information Theory*, pp. 1651–1655, July 2008.

[8] H. Yao, Y. Kochman and G. Wornell, "A Multi-Burst Transmission Strategy for Streaming over Blockage Channels with Long Feedback Delay," *IEEE Journal on Selected Areas in Communications*, Dec. 2011.

[9] G. Joshi, Y. Kochman, and G. Wornell, "On coding for reliable communication over packet networks," *International Symposium on Information Theory*, July 2012.

[10] R. Gallager, *Discrete Stochastic Processes.* Kluwer Academic Publishers, 1st ed., 1996.

[11] J. Barros, R. Costa, D. Munaretto, and J. Widmer, "Effective Delay Control in Online Network Coding," in *International Conference on Computer Communications*, pp. 208–216, Apr. 2009.

[12] R. Durrett, *Probability: Theory and Examples.* Cambridge University Press, 4th ed., 2010.

[13] B. Eisenberg, "On the expectation of the maximum of iid geometric random variables," *Statistics & Probability Letters*, vol. 78, pp. 135–143, Feb. 2008.

[14] L. Keller, E. Drinea and C. Fragouli, "Online Broadcasting with Network Coding," in *Network Coding Theory and Applications*, pp. 1 –6, Jan. 2008.

[15] J. Sundararajan, D. Shah and M. Médard, "Online network coding for optimal throughput and delay: the three-receiver case," in *International Symposium on Information Theory and its Applications*, Dec. 2008.

[16] P. Pakzad, C. Fragouli, and A. Shokrollahi, "Coding schemes for line networks," in *International Symposium on Information Theory*, pp. 1853–1857, Sept. 2005.