MORGAN & CLAYPOOL PUBLISHERS

BATS Codes

Theory and Practice

Shenghao Yang
Raymond W. Yeung

Synthesis Lectures on Communication Networks

R. Srikant, *Series Editor*

BATS Codes

Theory and Practice

Synthesis Lectures on Communication Networks

Synthesis Lectures on Communication Networks is an ongoing series of 50- to 100-page publications on topics on the design, implementation, and management of communication networks. Each lecture is a self-contained presentation of one topic by a leading expert. The topics range from algorithms to hardware implementations and cover a broad spectrum of issues from security to multiple-access protocols. The series addresses technologies from sensor networks to reconfigurable optical networks.
The series is designed to:

- Provide the best available presentations of important aspects of communication networks.
- Help engineers and advanced students keep up with recent developments in a rapidly evolving technology.
- Facilitate the development of courses in this field

Sharing Network Resources
Abhay Parekh and Jean Walrand
2014

Wireless Network Pricing
Jianwei Huang and Lin Gao
2013

Performance Modeling, Stochastic Networks, and Statistical Multiplexing, Second Edition
Ravi R. Mazumdar
2013

Packets with Deadlines: A Framework for Real-Time Wireless Networks
I-Hong Hou and P.R. Kumar
2013

Energy-Efficient Scheduling under Delay Constraints for Wireless Networks
Randall Berry, Eytan Modiano, and Murtaza Zafer
2012

NS Simulator for Beginners
Eitan Altman and Tania Jiménez
2012

Network Games: Theory, Models, and Dynamics
Ishai Menache and Asuman Ozdaglar
2011

An Introduction to Models of Online Peer-to-Peer Social Networking
George Kesidis
2010

Stochastic Network Optimization with Application to Communication and Queueing Systems
Michael J. Neely
2010

Scheduling and Congestion Control for Wireless and Processing Networks
Libin Jiang and Jean Walrand
2010

Performance Modeling of Communication Networks with Markov Chains
Jeonghoon Mo
2010

Communication Networks: A Concise Introduction
Jean Walrand and Shyam Parekh
2010

Path Problems in Networks
John S. Baras and George Theodorakopoulos
2010

Performance Modeling, Loss Networks, and Statistical Multiplexing
Ravi R. Mazumdar
2009

Network Simulation
Richard M. Fujimoto, Kalyan S. Perumalla, and George F. Riley
2006

BATS Codes: Theory and Practice

Shenghao Yang and Raymond W. Yeung

www.morganclaypool.com

ISBN: 9781627055970 paperback
ISBN: 9781627057158 ebook

DOI 10.2200/S00794ED1V01Y201708CNT019

A Publication in the Morgan & Claypool Publishers series
SYNTHESIS LECTURES ON COMMUNICATION NETWORKS

Lecture #19
Series Editor: R. Srikant, *University of Illinois at Urbana-Champaign*
Founding Editor Emeritus: Jean Walrand, *University of California, Berkeley*
Series ISSN
Print 1935-4185 Electronic 1935-4193

BATS Codes

Theory and Practice

Shenghao Yang
The Chinese University of Hong Kong, Shenzhen

Raymond W. Yeung
The Chinese University of Hong Kong

SYNTHESIS LECTURES ON COMMUNICATION NETWORKS #19

MORGAN & CLAYPOOL PUBLISHERS

ABSTRACT

This book discusses an efficient random linear network coding scheme, called BATched Sparse code, or BATS code, which is proposed for communication through multi-hop networks with packet loss. Multi-hop wireless networks have applications in the Internet of Things (IoT), space, and under-water network communications, where the packet loss rate per network link is high, and feedbacks have long delays and are unreliable. Traditional schemes like retransmission and fountain codes are not sufficient to resolve the packet loss so that the existing communication solutions for multi-hop wireless networks have either long delay or low throughput when the network length is longer than a few hops. These issues can be resolved by employing network coding in the network, but the high computational and storage costs of such schemes prohibit their implementation in many devices, in particular, IoT devices that typically have low computational power and very limited storage.

A BATS code consists of an outer code and an inner code. As a matrix generalization of a fountain code, the outer code generates a potentially unlimited number of batches, each of which consists of a certain number (called the batch size) of coded packets. The inner code comprises (random) linear network coding at the intermediate network nodes, which is applied on packets belonging to the same batch. When the batch size is 1, the outer code reduces to an LT code (or Raptor code if precode is applied), and network coding of the batches reduces to packet forwarding. BATS codes preserve the salient features of fountain codes, in particular, their rateless property and low encoding/decoding complexity. BATS codes also achieve the throughput gain of random linear network coding. This book focuses on the fundamental features and performance analysis of BATS codes, and includes some guidelines and examples on how to design a network protocol using BATS codes.

KEYWORDS

network coding, BATS code, multi-hop network, packet loss, degree distribution, finite-length analysis, BP decoding, inactivation decoding

Contents

Preface

Driven by new applications, both the scale and the scope of network communications are going to expand significantly in the next several decades. The Internet of Things (IoT) is going to connect tens or hundreds of billions of devices together into networks. New network infrastructures like space communication networks formed by low-orbit satellites or unmanned aerial vehicles (UAVs) are being built and tested. The exploration of outer space and deep sea requires network communications in areas that have not been covered before.

One of the trends is that communication networks will employ more and more wireless links than today. Most existing communication networks, for example WiFi, cellular networks, and satellite networks, involve at most two wireless links, namely the first hop and the last hop. In contrast, multi-hop wireless networks will dominate many new applications.

Wireless links suffer from packet loss due to fading, shadowing, hand off, interference, and other effects. Different from packet loss due to congestion, packet loss due to the above effects in wireless links cannot be reduced by rate control. Extensive research on TCP for combating these wireless link effects was conducted around the year 2000, when WiFi and cellular data were becoming popular. For multi-hop wireless networks, however, modifying TCP cannot prevent the significant rate decrease as the number of hops increases. It is therefore necessary to design new network communication protocols based on a different philosophy for combating packet loss.

Network coding provides a general theory for designing network protocols and achieves the theoretical communication limit of wireless networks with packet loss. In this book, we discuss an efficient random linear network coding scheme called BATched Sparse code, or BATS code. Proposed for communication through networks with packet loss, a BATS code consists of an outer code and an inner code. As a matrix generalization of a fountain code, the outer code generates a potentially unlimited number of batches, each of which consists of a certain number (called the batch size) of coded packets. The inner code comprises (random) linear network coding at the intermediate network nodes, which is applied on the packets belonging to the same batch. When the batch size is 1, the outer code reduces to an LT code (or Raptor code if precode is applied), and network coding of the batches reduces to packet forwarding. BATS codes preserve the salient features of fountain codes, in particular, their rateless property and low encoding/decoding complexity. BATS codes also achieve the throughput gain of random linear network coding.

Applying network coding for communication networks is much more complicated than applying a new channel coding technique for wireless communications, for example, which involves only a modification of the physical layer of the network protocol and is transparent to all

the higher layers. When applying network coding, however, both the transport layer and the network layer must be completely redesigned, and the link/MAC layer and even the physical layer need to be properly tuned to optimize the performance. For a particular application, a network protocol based on BATS code must be designed with specific requirements and constraints. In this book, we provide guidelines and examples on how to design a network protocol using BATS codes.

ORGANIZATION

In Chapter 1, some background information is provided and various schemes for multi-hop networks are compared. We discuss in detail the fundamentals of the design and analysis of encoding, recoding, and decoding of BATS codes in Chapter 2–8. Chapter 2 presents the basic approaches to BATS code encoding and decoding. Chapter 3 introduces a simple BATS code network protocol. Chapter 4 discusses some advanced recoding techniques. The first four chapters are essential for readers who want to conduct BATS code-related research.

Chapters 5–9 comprise the major technical contents for the performance analysis and coding design for BATS codes. Chapter 5 analyzes the asymptotic performance of BP decoding. Chapter 6 discusses the achievable rates of BATS codes. Chapter 7 focuses on the finite-length analysis of BP decoding. Chapter 8 is devoted to inactivation decoding, including the finite-length analysis and practical design. Chapter 9 discusses how to apply BATS codes in a general network topology.

For readers with different interests, this book can be read in part as follows. To learn how to design a network protocol using BATS codes, Chapter 3 is the place to start with, and Chapters 4 and 9 include some further discussions. To understand how a degree distribution is designed, Section 5.1 should be read followed by Chapter 6 and Section 7.4. To study finite-length analysis, the first three sections of Chapter 7 and Section 8.2 should be read.

A major part of this book is rewriting and unifying the previous works of the authors [54, 95, 96, 97, 98, 101, 102]. There is also a significant part of the book consisting of new results that have not been published before. These include Chapter 4 on advanced recoding techniques, Sections 6.3 and 6.4 on BATS codes for multiple rank distributions, and Section 8.3 on practical design of inactivation decoding. Variations of BATS codes, e.g., BATS codes with variable batch sizes [98], quasi-universal BATS codes [91], and BATS codes with unequal error protection [90], are not discussed in this book.

Shenghao Yang and Raymond W. Yeung
August 2017

Acknowledgments

We thank Jun Ma, Haiwen Cao, Yu Liu, Hoover Yin, and Zhiheng Zhou for reading the drafts of the manuscript.

This work was partially supported by the NSFC Grant No. 61471215, and the University Grants Committee of the Hong Kong Special Administrative Region, China under Project No. AoE/E-02/08.

Shenghao Yang and Raymond W. Yeung
August 2017

CHAPTER 1

Preliminaries

1.1 COMMUNICATION THROUGH NETWORKS WITH PACKET LOSS

One fundamental task of communication networks is to distribute a bulk of digital data, called a *file*, from a source node to one or multiple destination nodes. We consider this file distribution problem in *packet networks*, where the file is divided into multiple packets, referred to as the *input packets*. The packets transmitted on the network links may suffer from various distortions (e.g., errors and malicious modifications). Here we assume that they are either correctly received or *lost*. A network link with packet loss is also called a lossy link.

Packet loss occurs due to various reasons. The communication media of a network link may have noise, fading and interference, which result in the failure of the physical layer decoding. A falsely decoded packet can be detected and deleted, and hence regarded as a packet loss. Usually, wireless communication links, e.g., satellite/underwater communications, wireless LAN and mobile communication networks, are more vulnerable to such packet losses due to interference and fading than wireline communication links, e.g., fiber and coaxial cable. Even when the network links are perfect, packet loss may occur due to faulty network hardware/software, insufficient processing power, or buffer overflow.

The network topology considered here can be very general. In our network model, there are three types of nodes: source nodes, destination nodes, and intermediate nodes.

- A *source* node has the file (or the input packets) for transmission.
- A *destination* node demands the file. A network with a single destination node is called a *unicast* and a network with multiple destination nodes that demand the same file is called a *multicast*.
- An *intermediate* node, also called a *relay* node, does not demand the file but helps the transmission of the file.

Figure 1.1 gives the simplest example of a network that consists of all the three types of network nodes. A network with at least one intermediate node is also called a *multi-hop network*. We focus on the transmission of a single file for both unicast and multicast in multi-hop networks, where the intermediate nodes have *a constant amount of computation power and storage (buffer)*.

A multi-hop network can be found in many applications, for example, wireless mesh/ad hoc networks, satellite communications, and underwater communications. An emerging trend

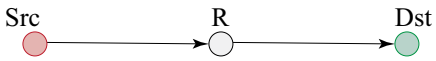


Figure 1.1: A three-node network. Node Src is the source node, node Dst is the destination node, and node R is the intermediate node that does not demand the input packets. Network links exist only between two neighboring nodes. Both links have packet loss rate ϵ.

is that more and more wireless networks will deploy relays. Both wireless LAN and 5G mobile networks will use millimeter waves (30–60 GHz) to support high-speed transmission, which travel solely by line-of-sight and are blocked by building walls [23]. To extend the network coverage in indoor and urban environments, it would be necessary to deploy relays. Multi-hop networks also exist in wireless sensor networks, Internet of vehicles, etc.

Routing is not a good solution for multi-hop networks with packet loss. Consider a *line network* which consists of a source node, a destination node and a sequence of consecutively connected intermediate nodes. For example, the three-node network in Figure 1.1 is a line network with two hops. Suppose each network link in the line network can transmit one packet per unit time and has a packet loss rate 0.2. The throughput of the line network with l hops using routing is 0.8^l, which decreases very fast even for a small l. For example, the throughput decreases to 0.512 when $l = 3$, and to 0.107 when $l = 10$.

The classical approaches for resolving packet loss in multi-hop networks include *retransmission* and *erasure coding*. As we will discuss later in this chapter, both approaches are effective only for special cases. As a consequence, in most of the existing networking practices, (i) a great amount of effort has been put in the physical layer so that the communication links have as low packet loss rate as possible, and (ii) only a small number of network links (typically, at most two) use wireless media, which may incur relatively high packet loss rate, while all the other links use more reliable wireline media like fiber. These classical approaches, however, cannot guarantee feasible performance for many practical multi-hop networks, for example networks with a relatively large number of concatenated lossy links (e.g., 10 or more) and networks with a long link-by-link delay.

In this monograph, we study how to effectively communicate through multi-hop networks with packet loss. We assume that *an intermediate node can only store a fixed number of packets and perform a fixed amount of operations on the stored packets per unit time*. These assumptions make it possible for our schemes to be implemented in real network devices. For many cases, (e.g., a network with a large number of concatenated lossy links), it is necessary to use *network coding* to obtain a reasonable throughput, where an intermediate network node may transmit new packets generated using the packets it has received. In contrast, existing network protocols mostly use store-and-forward at an intermediate network node, i.e., a network node only transmits the packets that it has received.

In the remainder of this chapter, we will review the traditional approaches for multi-hop transmission and then give a brief introduction to network coding. In this book, we assume the following problem settings.

1. We represent a packet by a sequence of finite field symbols. Specifically, fix a finite field $\mathbb{F}_q$ with size q, called the *base field*. A packet is denoted by a column vector in $\mathbb{F}_q^T$, which has T symbols in the base field.[1]

2. Unless otherwise specified, we assume that time is slotted. A network transmission starts at time 0, and one packet is sent on a link per time slot.

1.2 LINK-BY-LINK RETRANSMISSION

Consider the networks where an intermediate network node only stores and forwards the packets it has received. Retransmission is the most widely adopted approach for resolving packet loss in existing communication protocols, e.g., TCP, wireless LAN, and LTE.

When using retransmission, a network node does not remove a packet right after it has been transmitted on one of its outgoing links. Rather, it waits for a positive feedback indicating that the packet has been received correctly by the node(s) in the next hop. If no positive feedback is received after a certain time or a negative feedback is received, the network node retransmits the same packet (called a retry). This retransmission may be repeated for a preset number of times or until a positive feedback is received.

Although retransmission has been widely adopted and has demonstrated stable performance on the Internet, it has clear issues for general scenarios which limit the application of retransmission for reliable communication in, for example, wireless multi-hop networks. We will discuss the issues of retransmission by using several examples, but these issues also exist in general networks.

1.2.1 RETRANSMISSION IN LINE NETWORKS

In the ideal case that (i) feedback is reliable and has no communication cost, and (ii) each intermediate node has unlimited storage, link-by-link retransmission can achieve the network transmission capacity for line networks with packet loss. In the network in Figure 1.1, suppose both links have packet loss rate ϵ and one packet is sent on each link per unit time. Using link-by-link retransmission, the achievable throughput is $1-\epsilon$, which can be seen by using a queuing theoretic argument. Moreover, using the cut-set bound from information theory, we see that $1-\epsilon$ is indeed the network capacity.

In addition to the transmission rate, an important performance measure is the *transmission delay*, i.e., the time at which the destination node can decode the whole file. Suppose the packet loss rate on all the network links is ϵ. We know that on average, it takes $1/(1-\epsilon)$ transmissions

[1]Typically, we use $q = 2^8$, i.e., we use 8 bits (a byte) to represent a finite field symbol.

for a packet to be received by the node in the next hop. Therefore, when transmitting a single packet, the average transmission delay is $l/(1-\epsilon)$ time slots for a line network with l hops. When transmitting a sequence of K packets, the first packet takes $l/(1-\epsilon)$ time slots to reach the destination node, and the remaining $K-1$ packets take at least $(K-1)/(1-\epsilon)$ time slots to be received by the destination node. Therefore, a lower bound on the expected transmission delay is $(K+l-1)/(1-\epsilon)$ time slots for any transmission scheme.

We use the three-node network as an example to analyze the transmission delay and the storage requirement at the relay node. Let X_n be the number of packets stored at node R in Figure 1.1 at time n. First, X_0 has the value 0. At each of the subsequent time slot $0 < i \leq n$, with probability $1-\epsilon$, a new packet is added to the storage, and if $X_{i-1} > 0$ (i.e., the buffer is not empty at the last time slot), with probability $1-\epsilon$ a packet is removed from the storage (in the case that the packet is successfully received by node Dst). The dynamics of X_n can be analyzed as a random walk with a reflecting boundary at 0, and we have $\mathbb{E}[X_n] = \Omega(\sqrt{n\epsilon(1-\epsilon)})$ (see [58]).

For a file of K packets, the source node needs at least $K/(1-\epsilon)$ time slots to complete the transmission. When $n = K/(1-\epsilon)$, the expected number of packets storaged at node R is $\Omega(\sqrt{\epsilon K})$.

- The storage at node R must increase with K so that all the packets can be received by the destination node. For streaming applications with an unbounded number of packets, the required storage is also unbounded.
- To complete the transmission of these $\Omega(\sqrt{\epsilon K})$ packets in the storage, node R uses $\Omega(\sqrt{\epsilon K}/(1-\epsilon))$ extra time slots. So the transmission delay is $K/(1-\epsilon) + \Omega(\sqrt{\epsilon K}/(1-\epsilon))$ time slots.

For a line network with $l > 1$ hops, we can repeat the above analysis hop-by-hop. Note that an intermediate node may not transmit the K packets consecutively since new packets may not arrive after transmitting all the previous packets, but it still takes at least $K/(1-\epsilon)$ time slots (which may not be consecutive) to complete the transmission. So our analysis at node R becomes an upper bound, i.e., the expected number of packets stored at an intermediate node is $O(\sqrt{\epsilon K})$ and the end-to-end expected transmission delay is $K/(1-\epsilon) + O(l\sqrt{\epsilon K}/(1-\epsilon))$ time slots. Note that in terms of transmission delay, the link-by-link retransmission scheme is optimal.

Buffer Overflow

In practice, an intermediate node has a limited storage. An overflow occurs when the storage is full and a new packet is received. We may delete the new packet or replace an existing packet by the new one. One mechanism for alleviating buffer overflow is to stop the retransmission and discard the packet after a preset number of retries. Such a mechanism has been applied in many link layer protocols. For example, in the IEEE 802.11 MAC layer, the maximum number of

retries is set to 4. Thus, retransmission with a finite storage at the intermediate network nodes cannot guarantee the reliable transmission of every packet, even with ideal feedback.

Another mechanism for alleviating buffer overflow is to control the transmission rate of the previous node. For example, when the MAC layer of a node sees that its buffer is going to be full, it may issue a control message to the previous node to reduce the transmission rate. In the existing Internet protocol stack, the rate control is not implemented hop-by-hop, but end-to-end. TCP handles the end-to-end rate control to reduce the packet loss generated by buffer overflow, and employs an end-to-end retransmission scheme to guarantee the reliable transmission from the source node to the destination node.

Feedback Issues

Both link-by-link and end-to-end retransmissions, as well as rate control, however, require feedback, which may not be reliable/available and it incurs communication cost. In outer-space, satellite and underwater communications, for example, the propagation delay of each hop is very long, e.g., a few minutes, and feedback is unreliable and has a long delay. Moreover, feedback incurs communication resources. In LTE, a dedicated physical channel is allocated for feedback (as well as other control messages) and a stronger error correction code is used to protect the feedback messages. In WLAN, feedback shares the same physical channel with data messages and takes up a fraction of the total transmission time. Therefore, in practical communication systems, retransmission based reliable transmission schemes not only suffer from long delay but also are far from being throughput optimal.

1.2.2 WIRELESS ERASURE RELAY NETWORK

A *wireless erasure relay (WER) network*, illustrated in Figure 1.2, is similar to the three-node network except that there exists an extra link between node Src and node Dst. Here we assume that these links share the same physical communication media and node Src and node R apply a TDMA strategy so that they do not transmit simultaneous at any time slot. The transmission of node Src is broadcasting so that the same packet is transmitted on both its outgoing links. Node Dst is supposed to be much further away from node Src than node R, and hence the reception at node Dst from node Src is usually not as reliable as at node R. Therefore, we assume $\epsilon_1 > \epsilon$. (If $\epsilon_1 \leq \epsilon$, the relay node is not useful because it is always better to use (Src, Dst) than (R, Dst) to transmit a packet.)

The capacity of this network is $\frac{1-\epsilon\epsilon_1}{1+\epsilon_1}$ [61], which can be achieved by the following retransmission scheme when all the nodes can cooperate ideally.[2] Node Src keeps on retransmitting a packet until either node R or node Dst receives it correctly. If node Dst receives the packet, node R does nothing and node Src transmits another packet. If node R receives the packet but node

[2]Note that when $\epsilon_1 = 1$ (i.e., the link (Src, Dst) cannot transmit anything), the capacity is $(1-\epsilon)/2$, half of the similar line network with 2 hops. The reason is that in the wireless relay network, both node R and node Src use only half of the time for transmission; while in the line networks, we assume all the links have dedicated communication media.

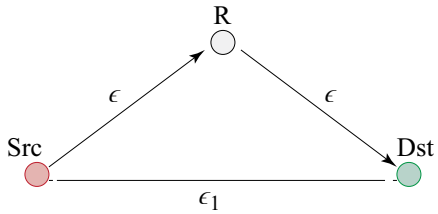


Figure 1.2: A wireless erasure relay network. Node Src is the source node, node Dst is the destination node, and node R is the intermediate node that does not demand the input packets. The links (Src, R) and (R, Dst) have packet loss rate ϵ, the link (Src, Dst) has packet loss rate ϵ_1. One packet is sent on each link per time slot.

Dst does not, node R transmits the packet it has received to node Dst by retransmission. After node Dst receives the packet, node Src continues to transmit another packet. The optimality of the retransmission scheme depends on the existence of ideal feedback for all links.

The above cooperative scheme, however, requires ideal feedback among all nodes. In practice, a higher loss rate on the (Src, Dst) link usually means the feedback on the reverse link is also not reliable. We will introduce later in this chapter a coding approach that does not require cooperation among the nodes.

1.2.3 RETRANSMISSION FOR MULTICAST

Here we use an example to show that, even with ideal feedback (instantaneous, reliable and cost free), retransmission can be far from optimal as the number of destination nodes increases. Note that though we use a single-hop network in the example, the same issue occurs in a multi-hop network.

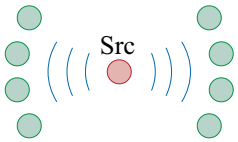


Figure 1.3: An example of wireless multicast. The source node Src uses a wireless communication media sends the same message to all the destination nodes. Suppose all the destination nodes experience the same packet loss rate ϵ, and the packet losses are independent among all the destination nodes.

Consider the wireless network in Figure 1.3. Suppose the source node has a packet to transmit using retransmission, and the feedback from all the destination nodes is instantaneous, reliable and cost free. Let N be the number of destination nodes and let T_i, $1 \leq i \leq N$, be the time used by the i-th destination node to receive the packet. We know that $\Pr\{T_i \geq t\} = \epsilon^{t-1}$

for $t \geq 1$. The time used by the source node to complete the transmission of a packet is $T^*(N) = \max\{T_1, \ldots, T_N\}$, and

$$\begin{aligned}\Pr\{T^*(N) \geq t\} &= 1 - \Pr\{T^*(N) < t\} \\ &= 1 - \Pr\{T_i < t, i = 1, \ldots, N\} \\ &= 1 - \prod_{i=1}^{N} \Pr\{T_i < t\} \\ &= 1 - (1 - \epsilon^{t-1})^N. \end{aligned} \tag{1.1}$$

When the number of users N tends to ∞, we see that $\Pr\{T^*(N) \geq t\}$ tends to 1, i.e., the source node cannot stop transmitting after any fixed amount of time.

Now, let us show that

$$\lim_{N\to\infty} \mathbb{E}[T^*(N)] = \infty. \tag{1.2}$$

In other words, the achievable multicast rate of retransmission in this example decreases to 0 as $N \to \infty$. For all N, by Markov's inequality, for all $t \geq 0$, we have

$$\mathbb{E}[T^*(N)] \geq t \Pr\{T^*(N) \geq t\}.$$

Then

$$\lim_{N\to\infty} \mathbb{E}[T^*(N)] \geq t \lim_{N\to\infty} \Pr\{T^*(N) \geq t\} = t,$$

implying (1.2).

The capacity of this network, however, is $1 - \epsilon$ for any given number of N, which can be achieved using erasure codes (see the next section). In general, network coding must be employed to achieve the network capacity of multicast network (see Section 1.4).

1.3 ERASURE CODING

To resolve the feedback related issues of the retransmission approach, researchers were motivated to consider erasure codes for networks with packet losses, which can reduce the use of feedbacks for reliable transmission. Examples of erasure codes include Reed-Solomon codes and LDPC codes. We use *fountain codes* as example to discuss how to use erasure codes to resolve packet loss. The use of erasure codes for Internet generated a lot of research interests in the 1990s, where fountain codes were the most celebrated achievement.

1.3.1 INTRODUCTION TO FOUNTAIN CODES

The concept of fountain codes first appeared in [9], where a potentially unlimited sequence of encoded packets can be generated from a given set of input packets such that the original file can be recovered from any subset of the encoded packets of size equal to or only slightly larger than the number of input packets. Fountain codes do not have a fixed coding rate as Reed-Solomon

codes and LDPC codes. A source node can keep transmitting the encoded packets of a fountain code until the destination node decodes all the input packets correctly. Therefore, fountain codes are also called *rateless codes*.

Examples of fountain codes include LT codes [40], Raptor codes [68] (a.k.a. online codes [50]). LT codes are a class of (binary) fountain codes introduced in [40]. An LT encoder generates encoded packets from K *input packets* using a degree distribution $\mathbf{\Psi} = (\Psi_0, \Psi_1, \cdots, \Psi_K)$. To generate an encoded packet, the degree distribution is sampled and an integer value d is returned with probability Ψ_d. Then d distinct input packets are chosen randomly and they are added together to yield the encoded packet. A robust soliton distribution for $\mathbf{\Psi}$ that guarantees successful *belief propagation decoding* using any n encoded packets, where n is slightly larger than K, is used. The encoding and decoding complexity of LT codes is $O(\log(K)T)$ bit-wise XOR operations per packet, where T is the number of symbols in a packet.

Raptor codes [68] further reduce the encoding/decoding complexity of LT codes by precoding. The input packets are first encoded by an erasure code, called the precode, the outputs of which are called the *intermediate packets*. These intermediate packets are then encoded by a variation of LT codes. This variation of LT codes only guarantees the recovery of a given fraction of the intermediate packets and the precode code is capable of recovering all the input packets in face of a given fraction of erasures.

Readers are referred to [46] for an excellent review of fountain codes and to [69] for more detailed discussion on LT/Raptor codes.

1.3.2 FOUNTAIN CODES FOR WIRELESS BROADCAST

The rateless property makes fountain codes more suitable than retransmission for many scenarios, in particular for multicast. Consider the wireless broadcast network studied in Figure 1.3, for which retransmission cannot achieve the capacity when the number of destination node is larger than one (see Section 1.2.3). Here we discuss a fountain code approach for broadcasting in this wireless network, where the erasure rate for each destination node may not be the same. Instead of transmitting the original input packets, the source node transmits the encoded packets of a fountain code until all the destination nodes decode all the input packets correctly. No feedback is required for each individual transmitted packet. As long as the fountain code is capacity achieving for erasure channels, this approach can achieve the capacity of wireless broadcasting for an arbitrary number of destination nodes.

1.3.3 FOUNTAIN CODES FOR LINE NETWORKS

Fountain codes were originally proposed for the Internet, where the end-to-end communications can be modeled as an erasure channel, i.e., a line network with one hop. Several schemes have been discussed in the literature for using fountain codes in a line network with more than one hop. However, all these schemes have various issues so that they cannot meet the require-

ments stated in Section 1.1, namely that the intermediate nodes have a constant amount of computation power and storage.

Consider a line network with l hops, where the packet loss rate is ϵ on all the network links. The source node or a relay node needs to transmit at least $K/(1-\epsilon)$ packets so that the node in the next hop receives at least K packets, which is a necessary requirement such that the destination node can decode all the input packets.

Scheme 1.1 Link-by-link Encoding/Decoding The first scheme is to apply fountain codes link-by-link [58]. The source node transmits the fountain code encoded packets. The first intermediate node completely decodes the input packets, and then encode the (decoded) input packets using a fountain code. The same operation in the first link is repeated in the subsequent links.

This scheme is capacity achieving and does not depend on link-by-link feedbacks for reliability.[3] However, this scheme requires both the memory size and the computation cost at the intermediate nodes to increase linearly with the file size for transmission. Moreover, the decoding and encoding at an intermediate node generate significant processing delay that accumulates hop-by-hop.

Assume that the decoding of fountain codes is instantaneous. From the decoding of the file in the previous node, a relay node or the destination node needs to wait for $t \approx K/(1-\epsilon)$ time slots before receiving sufficient coded packets for decoding the K input packets. A relay node only starts to transmit after all the input packets have been decoded. So the overall time taken by the destination node to decode all the input packets is $lt \approx lK/(1-\epsilon)$ time slots. Moreover, a relay node needs to buffer at least K packets to perform decoding and encoding.

Scheme 1.1 can be applied consecutively to achieve the capacity of the network. Suppose that the source node has N files each of which has K packets. The link-by-link fountain code scheme is used and each node uses only t time slots to transmit each file. The source node uses the first t time slots to transmit the first file, and then switch to the second file in the second t time slots. There is no extra delay between the transmission of different files. The overall time used for decoding the N files is $lt + (N-1)t$ so that the transmission throughput converges to $1-\epsilon$ as $N, K \to \infty$.

Scheme 1.2 Re-encoding In another scheme, the source node transmits the fountain code encoded packets, and an intermediate node re-encodes the packets it receives using fountain codes without decoding [18, 58]. In other words, an intermediate node treats the packets it receives as the input packets and apply a fountain code to encode the received packets. The destination node decodes multiple layers of fountain codes.

[3] This scheme may not completely eliminate the requirement of feedbacks in network communication protocols. For example, a feedback may be required when an intermediate node completely decodes the input packets. But it does not depends on feedback for reliable transmission of each input packets.

Compared with the first scheme, this scheme achieves the capacity and reduces the decoding cost at the intermediate node, but it does not change the fact that both the storage size and the computation cost at the intermediate nodes need to increase linearly with the file size for transmission. On the other hand, the decoding complexity at the destination node increases linearly with the network length, since the number of layers of fountain codes required to be decoded is equal to the network length. Moreover, each layer of fountain codes introduces extra coding overheads, which increases linearly the network length.

In the re-encoding scheme, an intermediate node needs to collect at least K packets before completing the re-encoding, so that the processing delay and the storage cost at relay nodes are the same as the link-by-link scheme.

Scheme 1.3 Systematic Re-encoding This scheme is similar to the re-encoding scheme, but a systematic fountain code is used at the intermediate nodes [58]. Specifically, all the received packets at an intermediate node are transmitted as the systematic packets of the fountain code. In addition to the systematic packets, an intermediate node also transmits a number of re-encoded packets for compensating the packet loss in its outgoing link.

Compared with the re-encoding scheme, the systematic re-encoding scheme can reduce the intermediate nodes storage and computation cost, as well as the transmission delay. In this scheme, a relay node transmits K packets it receives from the previous node, and $K/(1-\epsilon) - K = K\epsilon/(1-\epsilon)$ packets it generates from re-encoding. Note that the relay node does not need to keep all the received packets for recoding. For each received packet, the relay node can delete it after transmitting and adding it to the re-encoded packets. Re-encoding is completed after receiving K packets. So the storage cost at a relay node is $K\epsilon/(1-\epsilon)$ packets, and each relay node incurs an extra delay of $K\epsilon/(1-\epsilon)$ time slots. The network transmission delay is $K/(1-\epsilon) + (l-1)K\epsilon/(1-\epsilon)$.

1.3.4 FOUNTAIN CODES FOR WIRELESS ERASURE RELAY NETWORK

We use the WER network in Figure 1.2 to illustrate some limitations of using fountain codes in more complicated network topologies than line networks. Recall that the capacity of this network is $\frac{1-\epsilon\epsilon_1}{1+\epsilon_1}$ when $\epsilon_1 > \epsilon$. The retransmission scheme we have discussed can achieve the capacity, but it depends heavily on the availability of ideal feedback. To alleviate the need of feedback for reliable communication, we will discuss two schemes using fountain codes.

The WER network is similar to the three-node network, except for (i) the extra link (Src, Dst), and (ii) the common wireless media for all links. Schemes 1.1 and 1.2 for the three-node network can be applied on the WER network by ignoring the (Src, Dst) link. In order to benefit from the link (Src, Dst), we need to modify these schemes without significantly increasing the encoding/decoding/re-encoding complexity. These (modified) schemes, however, share

the same issues of Schemes 1.1 and 1.2 for the three-node network. Moreover, these schemes cannot achieve the capacity.

Scheme 1.4 The source node encodes the input packets by a fountain code and transmits the fountain code encoded packets until node R can decode the input packets. Node R decodes and re-encodes the (decoded) input packets using another fountain code, and transmits the fountain code encoded packets until node Dst can decodes the input packets. Node Dst uses the packets received from both Src and R for decoding.

In the above scheme, the source node needs to transmit $t_1 \approx K/(1-\epsilon)$ packets so that node R can decode successfully the input packets, while node Dst can receive about $K(1-\epsilon_1)/(1-\epsilon)$ packets. To decode the input packets, node Dst needs to receive $K - K(1-\epsilon_1)/(1-\epsilon)$ extra packets from node R. Thus, node R needs to transmit $t_2 \approx (K - K(1-\epsilon_1)/(1-\epsilon))/(1-\epsilon)$ packets to meet the requirement. Therefore, the achievable rate of the scheme is $K/(t_1+t_2) \approx (1-\epsilon)^2/(1-2\epsilon+\epsilon_1)$, which is less than $\frac{1-\epsilon\epsilon_1}{1+\epsilon_1}$, the capacity. This scheme is not optimal due to the transmission phase of Src. Specifically, the number of packets transmitted by Src is superfluous because it is not necessary that R can decode all the input packets.

Scheme 1.5 The source node encodes the input packets by a fountain code and transmits a sufficient number of the fountain code encoded packets so that the packets received by R and/or Dst together can jointly decode the input packets. Node R re-encodes the received packets using another fountain code, and transmits the fountain code encoded packets until node Dst can decode the received packets of R. Node Dst uses the packets received from Src and the decoded packets from R for decoding.

In the above scheme, each packet transmitted by Src can be received by at least one of R and Dst with probability $1-\epsilon\epsilon_1$. Therefore, Src needs to transmit $t_1' \approx K/(1-\epsilon\epsilon_1)$ packets so that the packets received by R and/or Dst together can jointly recover the input packets. Meanwhile, R receives about $K(1-\epsilon)/(1-\epsilon\epsilon_1)$ packets. Node R can use another fountain code to transmit the packets it has received to Dst using $t_2' \approx K/(1-\epsilon\epsilon_1)$ time slots. Therefore, the achievable rate of this scheme is $K/(t_1'+t_2') \approx (1-\epsilon\epsilon_1)/2$, which is also less than the capacity.

The above scheme is not optimal due to the second phase. In principle, node R only needs to transmit the packets that are not received by Dst. There are about $K(1-\epsilon)\epsilon_1/(1-\epsilon\epsilon_1)$ such packets. Since we do not assume any feedback from Dst to R, the latter does not know which packets have been received by the former.

1.4 NETWORK CODING

Network coding [2] allows an intermediate node to generate and transmit new packets using the packets it has received. In general, achieving the capacity of a network requires the use of network coding. In particular, when the network is error-free, linear network coding was proved to be sufficient for multicast communications [32, 35] and can be realized distributedly by random linear network coding (RLNC) [12, 21, 26, 64]. Readers are referred to [104] for a brief history of network coding and to [15, 22, 52, 103] for more detailed and general discussions on network coding. Here, we focus on the network coding schemes for networks with packet loss.

1.4.1 RANDOM LINEAR NETWORK CODING

The following network coding method has been proved to achieve the multicast capacity for networks with packet loss in a wide range of scenarios [13, 45, 88], which is referred to as the *baseline random linear network coding scheme (baseline RLNC scheme)*.

Scheme 1.6 Baseline RLNC The source node transmits random linear combinations of the input packets and an intermediate node transmits random linear combinations of the packets it has received. Note that no erasure codes are required for each link although packet loss is allowed. Network coding itself plays the role of end-to-end erasure coding. A destination node can decode the input packets when it receives enough coded packets with linearly independent coding vectors.

Let us use the three-node network in Figure 1.1 as an example to illustrate how this scheme works. Suppose the source node Src has K packets for transmission, denoted by $b_1, b_2, \ldots, b_K$. We equate a set of packets to a matrix formed by juxtaposing the packets in this set. For example, we denote the set of input packets by the matrix

$$\mathbf{B} = \left[b_1, b_2, \cdots, b_K\right].$$

Let $K' = (1+\delta)K$, where $\delta > 0$. The source node generates independently K' random linear combinations of these K packets, each of the form $\sum_{i=1}^{K} \alpha_i b_i$, where α_i is selected uniformly at random from the base field. The generated packets can be collectively represented by

$$\mathbf{X} = \mathbf{BA},$$

where $\mathbf{A}$ is a $K \times K'$ matrix with i.i.d. uniformly distributed entries (which is also called a totally random matrix).

Suppose each link of the network is used K' times. The K' linear combinations in X are transmitted by node Src in K' time slots. The packet loss on link $(\mathrm{Src}, \mathrm{R})$ or $(\mathrm{R}, \mathrm{Dst})$ can be modeled by a $K' \times K'$ random *diagonal* matrix $\mathbf{E}$ with independent components, where a diagonal component is 0 with probability ϵ and is 1 with probability $1-\epsilon$. The network coding

at the intermediate node can be modeled by a $K' \times K'$ upper triangular matrix Φ, where all the upper triangular entries are i.i.d. and uniformly distributed over the base field. With these matrices, the received packets at the sink node can be represented by

$$\mathbf{Y} = \mathbf{X}\mathbf{E}_1\Phi\mathbf{E}_2 = \mathbf{B}\mathbf{A}\mathbf{E}_1\Phi\mathbf{E}_2,$$

where $\mathbf{E}_1$ and $\mathbf{E}_2$ are independent random matrices with the same distribution as $\mathbf{E}$. We usually call $\mathbf{H} = \mathbf{E}_1\Phi\mathbf{E}_2$ the *transfer matrix*.

Now let us see how to decode $\mathbf{B}$ at the destination node. First, the destination node knows the instance of $\mathbf{AH}$ by means of the transmission of a set of *coefficient vectors* by the source node [21]. Suppose for each input packet b_i, K out of the T symbols ($T > K$) of the packet are used to transmit a coefficient vector, where the coefficient vectors of all the input packets form the identity matrix of dimension K. Therefore, we have

$$\mathbf{Y} = \mathbf{B}\mathbf{A}\mathbf{H} = \begin{bmatrix} \mathbf{I} \\ \mathbf{B}' \end{bmatrix} \mathbf{A}\mathbf{H} = \begin{bmatrix} \mathbf{A}\mathbf{H} \\ \mathbf{B}'\mathbf{A}\mathbf{H} \end{bmatrix}, \tag{1.3}$$

and the instance of $\mathbf{AH}$ can be recovered as part of the received packets at the destination node. Now, if the instance of $\mathbf{AH}$ is also full rank, the matrix $\mathbf{B}'$ can be recovered by solving a system of linear equations. It can be shown that for any $\delta > \epsilon/(1-\epsilon)$, the matrix $\mathbf{AH}$ is full rank with a probability converging to 1 as the size of the base field tends to infinity (or for the binary field as K tends to infinity).

The above analysis can be generalized for a line network with any fixed number of hops. For a line network with l hops, the transmission delay is $(K + l - 1)/(1-\epsilon)$ time slots, achieving the lower bound given in Section 1.2.1. The baseline RLNC scheme achieves the optimal transmission delay of link-by-link retransmission [70], and hence the expected transmission delay of the baseline RLNC scheme is $K/(1-\epsilon) + \mathrm{O}(l\sqrt{\epsilon K}/(1-\epsilon))$.

The baseline RLNC scheme also achieves the capacity of the wireless broadcast network in Figure 1.3 and the WER network in Figure 1.2 without feedback for reliable communication, though we may need, for example, the feedback for the notification of successful decoding from the destination node or the feedback for scheduling the node transmissions. For the WER, we need to know the packet loss rates on all the links in advance so that the transmission time of node Src and R can be optimally scheduled.

In particular, for the WER network in Figure 1.2, node Src transmits about $t_1' = K/(1-\epsilon\epsilon_1)$ random linear combinations of the input packets, so that the total number of packets received by R or Dst is about K, which is sufficient to recover all the input packets. Meanwhile, the number of packets received by R only is about $K(1-\epsilon)\epsilon_1/(1-\epsilon\epsilon_1)$. It is not necessary for R to transmit exactly these packets to Dst. Instead, node R only needs to transmits about

$$t_2'' = \frac{K(1-\epsilon)\epsilon_1/(1-\epsilon\epsilon_1)}{1-\epsilon} = \frac{K}{1-\epsilon\epsilon_1}\epsilon_1$$

random linear combinations of these packets, so that node Dst receives a total of at least $t_1'(1-\epsilon_1)+t_2''(1-\epsilon)=K$ independent linear combinations of the input packets.

The random linear combinations generated by the source node and the intermediate nodes play two roles. The first is a link-level erasure coding, which resolves the packet loss on network links. The second is network coding, which achieves the multicast capacity of error-free networks. Note that the intermediate network nodes do not perform any decoding operations, which is the same as in Scheme 1.2. The destination node decodes the encoding at the source node and the intermediate network node jointly, and decodes the network coding and the link-level erasure coding jointly.

Complexity Issues of Baseline RLNC

The baseline RLNC scheme has been implemented for small numbers of input packets, e.g., 32 (see [10]), but the scheme is difficult to be implemented efficiently when the number of input packets is relatively large (e.g., several hundreds) due to the computational and storage complexities and the coefficient vector overhead to be discussed below. Note that several hundred packets does not form a very big file. For example, a 500 KByte file may consist of 500 packets in IEEE 802.11 and 5,000 packets in IEEE 802.15.4.

Consider transmitting K packets where each packet consists of T symbols in a finite field. The encoding of a packet at the source node takes $O(TK)$ finite field operations. A finite field operation refers to the addition or multiplication of two field elements. At an intermediate node, all the packets it has received need to be buffered for network coding, so in the worst case, the storage cost is K packets and the computation cost of encoding a packet at an intermediate node is $O(TK)$ finite field operations. Decoding using Gaussian elimination costs on average $O(K^2+TK)$ finite field operations per packet. Although these complexities are polynomials in K, the baseline RLNC scheme is still difficult to implement for K large than, say, 300.

Coefficient vectors are used in the base line RLNC scheme to recover the instance of **AH**. For transmitting K input packets, the scheme requires that each packet includes a coefficient vector of K symbols. Hence, the coefficient vector overhead is K symbols per packet of T symbols. Network communication systems usually have a maximum value for T, e.g., several thousands of symbols. Therefore, for large values of K, the coefficient vector overhead is significant.

Many research works have been conducted with an aim to devise low-complexity, practical RLNC schemes. We first review some of these works.

1.4.2 FOUNTAIN CODES WITH NETWORK CODING

Fountain codes have been considered to be used with networks employing linear network coding. Specifically, in the baseline RLNC scheme, we can modify the source node to use a fountain code for encoding the input packets [58]. Using fountain codes reduces the encoding complexity of the baseline RLNC, but it is not straightforward to see how to reduce the decoding complexity,

the intermediate node storage/computation complexity and the coefficient vector overhead of the baseline RLNC.

The low complexity decoding scheme of fountain codes requires that the degrees of the received packets follow certain distributions (e.g., the robust soliton distribution). However, network coding at the intermediate nodes changes the degrees of the recoded packets, so that it is difficult to guarantee that the degrees of the received packets follow a specific distribution.

Heuristic algorithms have been proposed for special cases [5, 6, 11, 18, 38, 81, 83], but these approaches cannot achieve a performance similar to that of fountain codes over an erasure channel (i.e., larger coding overhead is incurred), and are difficult to be extended to general network settings. Moreover, they require the intermediate nodes to have a buffer size that increases linearly with the number of packets for transmission.

A different degree distribution problem of fountain codes has been studied for a network with two or more source nodes, one relay node and one destination node [37, 60, 66], called the distributed LT code problem. In these works, the relay node combines the encoded packets from multiple source nodes so that the destination node observes a desirable degree distribution. The focus of these works is how to reduce the coding overhead for networks with more than one source node. When reducing to one source node, the schemes of this works becomes the one that only forward the packets at the relay node. But as we have discussed, even the coding overhead is zero, using forwarding at the relay node is far from rate optimal.

See also a review of this line of works in [1] for packet loss in wireless sensor networks. These works do provide a possible direction for solving the complexity issues of the baseline RLNC scheme. However, fountain codes were not designed with network coding taken into consideration, and so these works fail to give a general approach that can reduce both the coefficient vector overhead and the decoding complexity.

1.4.3 CHUNKS

Another approach is to use disjoint *chunks*, each of which is a subset of the input packets.[4] A large file can be separated into a number of small chunks, and each chunk is transmitted independently, i.e., network coding is applied only for packets belonging to the same chunk [12]. Using this approach, the matrix $\mathbf{AH}$ in (1.3) becomes a block diagonal matrix. The use of small chunks can effectively reduce the encoding/decoding computational complexity and coefficient vector overhead. This idea is used in many earlier implementations of random linear network coding [10, 17, 30, 85] for demonstrating the advantage of network coding. However, the use of disjoint chunks introduces new issues.

Since all the chunks are disjoint and individually decoded, it is necessary to guarantee that they are all decoded correctly at the destination node, which is not a simple task. Suppose we use chunks of size M and we have totally n chunks. Since the transmission of all the chunks are independent, the problem is equivalent to transmitting n files, each of which has M packets,

[4] A chunk is also called a segment, class, group, or batch in literature.

using the baseline RLNC. The optimality of the baseline RLNC requires M to be large, while in the current approach we are particularly interested in the use of small M, e.g., 16 and 32.

There are different scheduling approaches for transmitting the n chunks from the source node to the destination nodes. Suppose the source node only transmits random linear combinations of the packets of a chunk. The first approach is called *sequential scheduling* [12], where the chunks are transmitted sequentially. The transmission of a chunk is stopped only when the source node receives the feedback from the destination node indicating that the current chunk has been decoded correctly. This sequential scheduling is exactly like TCP, except that the transmission unit is now a chunk instead of a packet.

For line networks, we can use link-by-link feedback instead of the end-to-end feedback. A network node keeps transmitting a chunk until it receives the feedback message indicating that the chunk is decodable at the next node (this can be verified by checking the number of linearly independent coefficient vectors). This link-by-link sequential scheduling of chunks is similar to the link-by-link retransmission scheme, except that the transmission unit is now a chunk instead of a packet. Similar to the analysis of link-by-link retransmission, we see that the transmission delay is $K/(1-\epsilon) + O(l\sqrt{\epsilon K}/(1-\epsilon))$ time slots, and the storage requirement at a relay node is $O(\sqrt{\epsilon K})$.

The issues of using feedbacks in sequential scheduling are similar to those of link-by-link retransmission that have been discussed in Section 1.2. For example, sequential scheduling of chunks is not scalable for multicast.

In round-robin scheduling [12], the chunks are served in a round-robin fashion, one chunk in each time slot. When a chuck is served, it is used for generating the coded packet to be transmitted. This process continues until all the chunks are decoded. In random scheduling [51], at each transmission opportunity, a chunk is randomly picked and used for generating the coded packet to be transmitted.

Both round-robin scheduling and random scheduling have similar issues. First, random/round-robin scheduling of chunks requires the intermediate nodes to cache all the chunks. So they are not suitable for line networks with limited storage at the intermediate nodes. Second, random/round-robin scheduling of chunks becomes less efficient when a fraction of chunks have been decoded. For example, after 50% of the chunks have been decoded at the destination node, at least half of the transmissions are redundant. Although it has been shown that random scheduling can be asymptotically capacity achieving when the chunk size is at or higher than the logarithmic order of the number of input packets [51], for a fixed chunked size, the achievable rate tends to zero as the number of input packets tends to infinity [79].

To resolve the issue of random scheduling of chunks, a precoding technique similar to that of Raptor codes has been considered [51]. Precoding allows the input packets to be recovered when only a fraction of all the chunks have been successfully decoded. This can be studied under the general framework of *chunked codes*. The first class of chunked codes proposed uses overlapping chunks [19, 36, 73]. An already decoded chunk can help the decoding of the other

chunks that overlap with it. Chunked codes using LDPC were proposed in [47, 48]. In [19, 36, 47, 48, 73], the design of chunked codes focuses on improving the performance of random chunk scheduling.

1.4.4 BATS CODES

BATS codes were proposed in parallel with overlapping chunked codes [95, 97]. A BATS code consists of an outer code and an inner code. As a matrix generalization of a fountain code, the outer code generates a potentially unlimited number of *batches*, each of which consists of M coded symbols. The inner code comprises (random) linear network coding [21, 32, 35] at the intermediate network nodes, which is applied on the symbols belonging to the same batch. When $M = 1$, the outer code becomes an LT code (or Raptor code if precode is applied), and network coding of the batches becomes forwarding.

The outer-code-inner-code structure provides a new framework to design and analyze the performance of efficient network coding schemes, which is also adopted in the design of chunked codes in [77, 78, 79, 94]. In this monograph, we study the design of both the outer code and the inner code of a BATS code.

The outer code of a BATS code preserves the salient features of fountain codes, in particular, their rateless property and low encoding/decoding complexity. Here, the rateless property means that the number of batches that can be generated by a BATS code is unlimited theoretically, and any subsets of the same number of batches generated at the source node have the same nature. This is the major difference between BATS codes and chunked codes—only a fixed number of chunks can be generated for all the chunked codes discussed above.[5]

This rateless property makes BATS codes more flexible than chunked codes in practical applications. For example, consider the transmission in a line network, where the source node does not know the packet loss rate on all network links. The coding rate of a chunked codes cannot be optimally determined so that the source node has to apply round-robin or random scheduling for chunk transmissions, which has been proven to be far from optimal for line networks. When using BATS code, since all the batches are statistically identical, different batches can be transmitted sequentially until the file can be decoded at the destination node.

As another example, consider the transmission from multiple source nodes, each of which has the same file, to the same destination node. When using chunked codes, some collaboration among the source nodes is required to avoid the transmission of the same chunks, which may reduce the benefit of using multiple source nodes. When using BATS codes, however, the collaboration among the source nodes is not required since each the batches generated by different source nodes are different with high probability.

[5]Chunked codes have been said to have a different "rateless" property in the sense that a network node can transmit an unlimited number of linear combinations of a chunk. We would say that this is the rateless property of an individual chunk (instead of the chunked code) since only the individual chunk can be recovered with high probability when a sufficient number of linear combinations of this chunk are received. In fact, we should avoid transmitting too many linear combinations of the same chunk.

To effectively use BATS codes, as well as other chunked codes, we also need a proper network coding scheme for transmitting the batches, which is also called the *inner code* of a BATS code. BATS and other (fixed rate) chunked codes can share the same inner codes with little modifications. Compared with the existing frameworks of chunked codes, like EC codes [78], Gamma codes [47], and L-chunked codes [94], BATS codes generally achieve higher rates and have the extra feature that an unlimited number of batches can be generated. Although we focus on BATS codes, the inner code discussed in this monograph applies to general chunked codes as well.

1.4.5 OTHER APPROACHES

In addition to the above approaches, there are techniques focusing on certain specific issues or scenarios. For example, if the intermediate nodes can cache only a small number of packets in the baseline RLNC scheme, a significant percentage of the min-cut can still be achieved in a two-hop line network [44], but the coefficient vector overhead and the encoding/decoding complexity remain to be the same as those of the baseline RLNC scheme. An error correction code based approach is proposed to reduce the coefficient vector overhead [25]. This approach puts a limit on the number of packets that can be combined together, but does not take the decoding complexity into consideration. Link-by-link feedback can be used to reduce the storage at the intermediate nodes [14, 31, 76]. A binary permutation matrix based approach to reduce the complexity of the finite field operations in linear network coding has been proposed in [27].

Lifted rank-metric codes [72] are designed for also correcting additive errors in the baseline RLNC scheme. Hence, as network coding schemes for networks with packet loss, they have the same drawbacks as baseline RLNC.

A general approach for resolving the coefficient vector overhead is to use non-coherent network coding schemes, where coefficient vectors are not used explicitly, which was first proposed in a subspace code framework without the chunked structure [33]. The achievable rate of using non-coherent chunked transmission has been studied in [71, 100]. A subspace-matrix superposition framework was proposed to design non-coherent chunked network codes [93]. But more research works are required for efficient non-coherent network coding schemes for networks with packet loss.

Information theoretic studies of the throughput and latency of line networks when the intermediate nodes have finite buffer can be found in [56, 82].

1.5 PERFORMANCE COMPARISON

We end this chapter with a comparison of the performance of different coding schemes for a line network of length l. Suppose the network links have packet loss rate ϵ, the file has K packets of length T, and the batch/chunk size is M. We consider the following performance measures:

- transmission throughput measures the capability of a scheme for continuous data transmission;
- transmission delay is the time from the beginning of transmission to the decoding of the whole file at the destination node;
- requirements of feedback and information on link loss rate for reliable communication or scheduling;
- encoding and decoding complexity at the source node and the destination node, respectively;
- storage and recoding complexity at the relay nodes;
- coding coefficient overhead; and
- rateless property.

In Table 1.1, the performance of BATS codes is compared with five other schemes: link-by-link retransmission, link-by-link fountain codes, systematic re-encoding fountain codes, baseline RLNC and disjoint chunks. We see that BATS codes demonstrate the best overall performance among all these schemes when the batch size is relatively small compared with the packet length and the number of packets. Compared with link-by-link retransmission and disjoint chunks, BATS codes do not require ideal link-by-link feedback and have smaller relay node storage requirement. Compared with RLNC, BATS codes have higher achievable rates, smaller relay node storage and computation cost, and smaller destination node decoding complexity when K is larger than M (which is almost always the case).

The encoding and decoding computation costs are similar to that of fountain codes, so that the outer code of a BATS code can be implemented efficiently by software and/or hardware. In addition to low encoding/decoding complexity, the inner code of a BATS code can be realized with constant computation and storage costs at the intermediate nodes. This desirable property makes BATS code a suitable candidate for the making of universal network coding based network devices that can potentially replace routers.

Table 1.1: Comparison of different schemes for a line network of length l. Here we assume that the network links have packet loss rate ϵ, the file has K packets of length T, and the batch/chunk size is M.

	Link-by-link Retransmission	Fountain Codes		RLNC	Disjoint Chunks	BATS Codes
		Link-by-link	Re-encoding			
Asymp. throughput	$1-\epsilon$	$\approx 1-\epsilon$	$\approx 1-\epsilon$	$\approx (1-\epsilon)\frac{T-K}{T}$	$\approx (1-\epsilon)\frac{T-M}{T}$	$\approx (1-\epsilon)\frac{T-M}{T}$
Transmission delay	$\frac{K+\mathrm{O}(l\sqrt{\epsilon K})}{1-\epsilon}$	$\frac{l\,K}{1-\epsilon}$	$\frac{K+lK\epsilon}{1-\epsilon}$	$\frac{K+\mathrm{O}(l\sqrt{\epsilon K})}{1-\epsilon}$	$\frac{K+\mathrm{O}(l\sqrt{\epsilon K})}{1-\epsilon}$	$\frac{K+\mathrm{O}(l\sqrt{\epsilon K})+Ml}{1-\epsilon}$
Source encoding	–	$\mathrm{O}\left(\frac{KT}{1-\epsilon}\right)$	$\mathrm{O}\left(\frac{KT}{1-\epsilon}\right)$	$\mathrm{O}\left(\frac{K^2T}{1-\epsilon}\right)$	$\mathrm{O}\left(\frac{KMT}{1-\epsilon}\right)$	$\mathrm{O}\left(\frac{KMT}{1-\epsilon}\right)$
Destination decoding	–	$\mathrm{O}(KT)$	$\mathrm{O}(KTl)$	$\mathrm{O}(K^3+K^2T)$	$\mathrm{O}(KM^2+KMT)$	$\mathrm{O}(KM^2+KMT)$
Relay storage	$\Omega(\sqrt{\epsilon K})$	K	$\frac{K\epsilon}{1-\epsilon}$	K	$\mathrm{O}(\sqrt{\epsilon K})$	$\mathrm{O}(M)$
Relay recoding	–	$\mathrm{O}\left(\frac{KT}{1-\epsilon}\right)$	$\mathrm{O}\left(\frac{KT\epsilon}{1-\epsilon}\right)$	$\mathrm{O}\left(\frac{K^2T}{1-\epsilon}\right)$	$\mathrm{O}\left(\frac{KMT}{1-\epsilon}\right)$	$\mathrm{O}\left(\frac{KMT\epsilon}{1-\epsilon}\right)$
Coding coefficient overhead	–	$\mathrm{O}(1)$	$\mathrm{O}(l)$	K	M	M
Ratelessness	–	yes	yes	yes	rateless within chunks	yes
Channel knowledge	not required	not required	useful for reducing relay storage	not required	not required	useful but not necessary
Feedback for reliability	ideal link-by-link feedback for each pocket	not required	not required	not required	ideal link-by-link feedback for each chunk	not required

CHAPTER 2

BATS Codes Basics

In this chapter, we introduce the basic encoding and decoding approaches of BATS codes. The rank of a matrix $\mathbf{A}$ is denoted by $\mathrm{rk}(\mathbf{A})$. In the following discussion, we equate a set of packets to a matrix formed by juxtaposing the packets in this set. For example, we denote the set of input packets by the matrix

$$\mathbf{B} = \begin{bmatrix} b_1 & b_2 & \cdots & b_K \end{bmatrix},$$

where b_i is the i-th input packet. On the other hand, we also regard $\mathbf{B}$ as a set of packets, and so, with an abuse of notation, we also write $b_i \in \mathbf{B}$, $\mathbf{B}' \subset \mathbf{B}$, etc. When we write $\mathbf{B}' \subset \mathbf{B}$, we also regard $\mathbf{B}'$ as a submatrix of $\mathbf{B}$.

2.1 ENCODING OF BATCHES

2.1.1 OUTER CODE: GENERATION OF BATCHES

A BATS code consists of an *outer code* and an *inner code*. Let us first describe the outer code of a BATS code, which generates coded packets in batches. A *batch* is a set of M coded packets generated from a subset of the K input packets. For $i = 1, 2, \ldots$, the i-th batch $\mathbf{X}_i$ is generated from a subset $\mathbf{B}_i \subset \mathbf{B}$ of the input packets by the operation

$$\mathbf{X}_i = \mathbf{B}_i \mathbf{G}_i,$$

where $\mathbf{G}_i$, a matrix with M columns, is called the *generator matrix* of the i-th batch. We call the packets in $\mathbf{B}_i$ the contributors of the i-th batch, and denote by A_i the index set of the packets in $\mathbf{B}_i$. For example, if $\mathbf{B}_i = \begin{bmatrix} b_1 & b_5 & b_7 \end{bmatrix}$, then $A_i = \{1, 5, 7\}$.

The value $\mathrm{dg}_i \triangleq |A_i|$ is called the *degree* of the i-th batch, which are independent random variables following a distribution $\mathbf{\Psi} = (\Psi_1, \ldots, \Psi_K)$, so that

$$\Pr\{\mathrm{dg}_i = d_i, i = 1, \ldots, n\} = \prod_{i=1}^{n} \Psi_{d_i}, \quad n = 1, 2, \ldots.$$

We call $\mathbf{\Psi}$ the *degree distribution*, which is one of the crucial parameters of a BATS code.

We focus on random encoding in this monograph, i.e., the generator matrix $\mathbf{G}_i$ is a $\mathrm{dg}_i \times M$ *totally random matrix*, in which all the entries are i.i.d. and uniformly distributed over the base field. Random generator matrices not only facilitate analysis but are also readily implementable. For example, $\mathbf{G}_i$, $i = 1, 2, \cdots$ can be generated by a pseudorandom number generator and can be recovered at the destination nodes by the same pseudorandom number generator.

There are no limits on the number of batches that can be generated. This rateless property can benefit the network transmission from many aspects.

- When the source node does not know the complete network status, e.g., the packet loss rate on the network links, the source node can just transmit different batches sequentially until the decoding of the file.
- Consider the transmission from multiple source nodes, each of which has the same file, to the same destination node. These source nodes can generate the batches independently without collaboration, and the destination node can decode the batches generated at different source node jointly.
- Consider the transmission from a source node to multiple destination nodes, each of which has received some batches. The source node generates new batches for transmission without knowning the batches in the destination nodes.

The approach to handling a large number of batches will be discussed in the next chapter.

The batch encoding process can be described by a Tanner graph. The Tanner graph has K *variable nodes*, where variable node i corresponds to the i-th input packet b_i, and a number of *check nodes*, where check node j corresponds to the j-th batch $\mathbf{X}_j$. Check node j is connected to variable node i if b_i is a contributor of $\mathbf{X}_j$. Associated with each check node j is the generator matrix $\mathbf{G}_j$. Figure 2.1 illustrates an example of a Tanner graph for encoding batches. Henceforth, we equate a variable node with an input packet, and a check node with a batch.

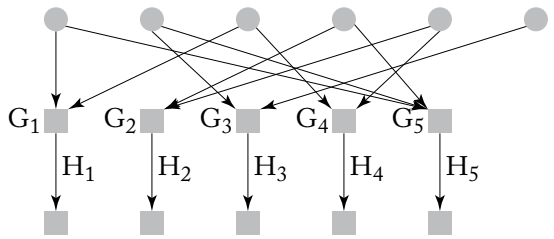


Figure 2.1: A Tanner graph for the inner and the outer code of a BATS code. The nodes on the first row are the variable nodes representing the input packets. The nodes on the second row are the check nodes representing the batches generated by the outer code. The nodes on the third row are the check nodes representing the batches processed by the inner code.

2.1.2 INNER CODE: TRANSMISSION OF BATCHES

Now we turn to the inner code of a BATS code, which is the linear network coding scheme at the network nodes and also called *recoding*. The batches generated by the outer code are transmitted in a network employing linear network coding, possibly with multiple destination nodes. We

assume that the end-to-end transformation of each batch from the source node to a destination node is a linear operation. Fix a destination node. Let $\mathbf{H}_i$ be the transfer matrix of the i-th batch and $\mathbf{Y}_i$ be the output (received) packets of the i-th batch. We have

$$\mathbf{Y}_i = \mathbf{X}_i \mathbf{H}_i = \mathbf{B}_i \mathbf{G}_i \mathbf{H}_i . \tag{2.1}$$

The number of rows of $\mathbf{H}_i$ is M. The number of columns of $\mathbf{H}_i$ corresponds to the number of packets received for the i-th batch, which may vary for different batches and is finite. If no packets are received for a batch, $\mathbf{Y}_i$ ($\mathbf{H}_i$) is the empty matrix of 0 columns.

We assume that $\mathbf{H}_i$, $i = 1, 2, \ldots$ are independent of the encoding process. The instance of $\mathbf{H}_i$ is known for decoding through the coefficient vector in the packet header [21] (see Section 1.4.1). We call $\mathrm{rk}(\mathbf{H}_i)$ the *rank of the i-th batch*.

To guarantee the end-to-end network operation on the batches as stipulated in (2.1), we may require that a network node can only apply network coding on the packets belonging to the same batch (otherwise $\mathbf{Y}_i$ may depend on $\mathbf{X}_i$ for $j \neq i$, not just on $\mathbf{X}_i$). Packet loss and dynamic network topology are allowed during the network transmission. The benefits of applying network coding within batches includes the following.

- The network coding complexity at a network node is $O(MT)$ finite field operations per packet, which does not depend on K.
- The coefficient vector overhead is bounded by M base field symbols. When the packet length T is sufficiently larger than M, this overhead is negligible.
- It is not necessary to keep all the batches at an intermediate network node for the purpose of network coding. As we will show in the following chapters, it is sufficient to cache one or several batches at an intermediate node.

Note that the requirement of network coding within a batch is not a necessity. For example, it is possible that network coding between packets of different batches is applied locally so that the coded packets of different batches at a network node can be decoded directly at the nodes in the next hop. Readers can find such an example of the application of BATS codes in Huang et al. [24] and Zhang et al. [107]. We will also discuss the use of network coding acrossing batches in Chapter 9.

The transfer matrices of batches are determined jointly by the inner code and the network topology between the source node and the destination node. Under the principle that only packets of the same batch can be recoded, we have a lot of freedom in designing the inner code, including how to manage the buffer contents, how to schedule the transmission of batches/packets, and how to use the feedback messages. We will use a typical network topology to illustrate how to design the inner code so that the benefit of BATS codes is maximized (see Chapter 3).

The empirical rank distribution of the transfer matrices is an important parameter for the design of BATS codes, which is also simply called the *rank distribution*. Note that, in general,

we do not assume that the transfer matrices of batches are i.i.d. As we will make it clear later, the rank distribution determines the maximum achievable rate of the outer code and provides sufficient information for a nearly optimal outer codes design. It is also not necessary for the source node to know the exact rank distribution. When we have partial or no knowledge about the rank distribution, certain achievable rates can also be guaranteed. In other words, we do not need to have the complete knowledge about the distribution of the transfer matrices for designing BATS codes; the rank distribution alone is sufficient.

2.2 GAUSSIAN ELIMINATION DECODING

Consider the decoding of the BATS code described above with $n \geq 1$ received batches $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n$. Assume that the sink node knows also $\mathbf{G}_i \mathbf{H}_i$ and A_i for $i = 1, \ldots, n$. A necessary and sufficient condition that the K input packets can be decoded is that the linear system

$$\mathbf{Y}_i = \mathbf{B}_i \mathbf{G}_i \mathbf{H}_i, \quad i = 1, \ldots, n$$

has a unique solution, which is the case only if $K \leq \sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i)$. Suppose

$$\frac{\sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i)}{n} \xrightarrow{\mathrm{P}} C \text{ as } n \to \infty. \tag{2.2}$$

Then C is an asymptotic upper bound on the achievable rate of BATS codes in packets per batch.

This upper bound can be achieved using the degree distribution that $\Psi_K = 1$, i.e., all the input packets are involved in generating every batch, so that $\mathbf{B}_i = \mathbf{B}$ for all i. A BATS code with $\Psi_K = 1$ is also called a random linear code, for which

$$\begin{bmatrix} \mathbf{Y}_1 & \cdots & \mathbf{Y}_n \end{bmatrix} = \mathbf{B} \begin{bmatrix} \mathbf{G}_1 & \cdots & \mathbf{G}_n \end{bmatrix} \begin{bmatrix} \mathbf{H}_1 & & & \\ & \mathbf{H}_2 & & \\ & & \ddots & \\ & & & \mathbf{H}_n \end{bmatrix}.$$

Write $\tilde{\mathbf{G}} = \begin{bmatrix} \mathbf{G}_1 & \cdots & \mathbf{G}_n \end{bmatrix}$ and $\tilde{\mathbf{H}} = \mathrm{diag}(\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_n)$. By construction, each row of $\tilde{\mathbf{G}}\tilde{\mathbf{H}}$ is a vector chosen uniformly at random from the subspace spanned by the rows of $\tilde{\mathbf{H}}$. If the K rows of $\tilde{\mathbf{G}}\tilde{\mathbf{H}}$ are linearly independent, the system of linear equations has a unique solution. Let $R = \sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i)$. We have

$$\Pr\{\mathrm{rk}(\tilde{\mathbf{G}}\tilde{\mathbf{H}}) = K | R = r\} = \zeta_K^r,$$

where

$$\zeta_K^r = \zeta_K^r(q) \triangleq \begin{cases} (1-q^{-r})(1-q^{-r+1})\cdots(1-q^{-r+K-1}) & 0 < K \leq r, \\ 1 & K = 0. \end{cases} \tag{2.3}$$

Therefore for any $\epsilon > 0$ and $K = n(C - 2\epsilon)$,

$$\begin{aligned}\Pr\{\mathrm{rk}(\tilde{\mathbf{G}}\tilde{\mathbf{H}}) = K\} &= \sum_{r \geq n(C-\epsilon)} \Pr\{\mathrm{rk}(\tilde{\mathbf{G}}\tilde{\mathbf{H}}) = K | R = r\} \Pr\{R = r\} \\ &= \sum_{r \geq n(C-\epsilon)} \zeta_K^r \Pr\{R = r\} \\ &\geq \sum_{r \geq n(C-\epsilon)} \zeta_K^{\lfloor n(C-\epsilon) \rfloor} \Pr\{R = r\} \\ &= \zeta_K^{\lfloor n(C-\epsilon) \rfloor} \Pr\{R \geq n(C-\epsilon)\}.\end{aligned}$$

Since both $\zeta_K^{\lfloor n(C-\epsilon) \rfloor}$ and $\Pr\{R \geq n(C-\epsilon)\}$ converge to 1 as n tends to infinity, $\Pr\{\mathrm{rk}(\tilde{\mathbf{G}}\tilde{\mathbf{H}}) = K\} \to 1$ as n tends to infinity. Therefore, C is achievable.

In the above analysis, we see that

- the achievability does not depend on the field size. Even binary field works; and
- the random linear code universally achieves C regardless of the distribution $\mathbf{H}_i$, $i = 1, \ldots, n$.

However, the random linear code has an encoding complexity of $O(KT)$ finite field operations per packet, and solving the above linear system by Gaussian elimination (GE) has a complexity of $O(K^3 + TK^2)$ finite field operations. Though the random linear code achieves C, in practice, we need to design BATS codes with lower encoding and decoding complexity.

One approach to achieve low encoding/decoding complexity is to use *sparse* encoding and *belief propagation (BP)* decoding. Another apporach, to be discussed in Chapter 8, is *inactivation decoding* that combines BP decoding with Gaussian elimination.

2.3 BELIEF PROPAGATION DECODING

We describe two BP decoders that can be used for different purposes.

2.3.1 BP(n) DECODER

The time index starts at 0 and increases by 1 after each decoding step. The decoding algorithm updates A_i, $\mathbf{G}_i$, and $\mathbf{Y}_i$ in each step. For each batch i and time t, let $A_i^{(t)}$, $\mathbf{G}_i^{(t)}$, and $\mathbf{Y}_i^{(t)}$ be the values of A_i, $\mathbf{G}_i$, and $\mathbf{Y}_i$ at time t, respectively. When $t = 0$, we have $A_i^{(0)} = A_i$, $\mathbf{G}_i^{(0)} = \mathbf{G}$ and $\mathbf{Y}_i^{(0)} = \mathbf{Y}_i$. Iterative formulae will be given for these variables at $t > 0$. We call $|A_i^{(t)}|$ the degree of batch i at time t.

We say a batch i is *decodable* at time t if $\mathrm{rk}(\mathbf{G}_i^{(t)}\mathbf{H}_i) = |A_i^{(t)}|$ (i.e., its degree is equal to the rank of $\mathbf{G}_i^{(t)}\mathbf{H}_i$), and an input packet is *decodable* at time t if it contributes to a decodable batch at time t. Denote by $\mathbf{B}_i^{(t)}$ a matrix formed by juxtaposing the input packets with indices in $A_i^{(t)}$. The associated linear system of batch i at time t is

$$\mathbf{Y}_i^{(t)} = \mathbf{B}_i^{(t)} \cdot \mathbf{G}_i^{(t)} \cdot \mathbf{H}_i.$$

Batch i at time t is decodable means that the above linear system, with $\mathbf{B}_i^{(t)}$ as the variable, has a unique solution.

The decoding algorithm operates as follows. For each time t, a decodable input packet is selected (if there is more than one such packets), substituted into the undecodable batches that it contributes to, and marked as *decoded*.[1] Suppose that the j-th input packet b_j is decoded at time t. We then substitute the decoded input packet into the batches it contributes to: For each batch i,

1. if $j \in A_i^{(t)}$, then $A_i^{(t+1)} = A_i^{(t)} \setminus \{j\}$, $\mathbf{G}_i^{(t+1)}$ is formed by removing the row g of $\mathbf{G}_i^{(t)}$ corresponding to the j-th input packet b_j, and $\mathbf{Y}_i^{(t+1)} = \mathbf{Y}_i^{(t)} - b_j g \mathbf{H}_i$; and

2. if $j \notin A_i^{(t)}$, then $A_i^{(t+1)} = A_i^{(t)}$, $\mathbf{G}_i^{(t+1)} = \mathbf{G}_i^{(t)}$ and $\mathbf{Y}_i^{(t+1)} = \mathbf{Y}_i^{(t)}$.

The decoding stops when there are no decodable input packets.

The BATS code decoding algorithm described above uses a given number n of batches, and is denoted by BP(n). For BP(n), we are interested in the time when the decoding stops, which is equal to the number of input packets that are decoded. For example, if BP(n) stops at time zero, then no input packets are decoded; if BP(n) stops at time K, then all the input packets are decoded. We will characterize the distribution of the stopping time of BP(n) for finite values of n (see Chapter 7).

2.3.2 RATELESS BP DECODER

Now let us see how to benefit from the unlimited number of batches. Suppose the encoder generates n batches. When BP(n) stops without all the input packets decoded, the encoder can generate more batches to resume the BP decoding procedure. We define the following *rateless BP decoder* BP* that consumes the batches one by one. BP* starts by fetching the first batch. When n batches are fetched ($n = 1$ to start with), BP(n) is applied. If BP(n) stops with all the input packets decoded, BP* stops; otherwise, one more batch is fetched and BP($n + 1$) is applied. Since the number of batches is unlimited, BP* will eventually stop with all the input packets decoded.

For BP*, we are interested in the number of batches consumed when the decoding stops. We will characterize the distribution of the number of batches consumed as well as the expected number of batches consumed by BP* in Chapter 7.

2.3.3 BP DECODING COMPLEXITY

In the following, computational complexity is expressed in the finite field operations. Suppose T and M are given, and K and n are the variables that tend to infinity in the big O notation.

[1] Note that in each step, the choice of the decodable input packet to substitute does not affect the time when the decoding stops (see [97, Appendix B]).

To generate a batch of degree d, we combine d packets together M times, each time with a different linear combination. Thus, generating a batch with degree d costs $O(TMd)$ finite field operations, and so the encoding complexity of n batches is $O(TM\sum_{i=1}^{n} d_i)$, which converges to $O(TMn\bar{\Psi})$ finite field operations when n is large, where $\bar{\Psi} = \sum_d d\Psi_d$ is the average degree.

Let $k_i = \mathrm{rk}(\mathbf{H}_i)$ and let k_i' be the rank of $\mathbf{G}_i\mathbf{H}_i$ when check node i becomes decodable. It is clear that $k_i' \le k_i \le M$. By the definition of the decodability of a check node, k_i' is also the degree of check node i when it becomes decodable. Since the degree of a check node can only decrease at each step of the decoding process, we have $k_i' \le d_i$. The decoding processing involves two parts: the first part is the decoding of the decodable check nodes, which costs $O(\sum_i k_i'^3 + T\sum_i k_i'^2)$ finite field operations; the second part is the updating of the decoding graph, which costs $O(T\sum_i (d_i - k_i')M)$ finite field operations. So the total complexity is $O(\sum_i k_i'^3 + T\sum_i k_i'^2 + T\sum_i (d_i - k_i')M)$, which can be simplified to $O(nM^3 + TM\sum_i d_i)$. When n is large, the complexity converges to $O(M^3 n + TMn\bar{\Psi})$ finite field operations. Usually, T and $\bar{\Psi}$ is considerably larger than M, so that the second term is dominant.

2.3.4 SOLVABILITY OF A BATCH

Recall that $\mathbb{F}_q$ is the finite field with q elements. For fixed integers $r, m > 0$, we say an $r \times m$ matrix over $\mathbb{F}_q$ is *totally random* if all the entries of the matrix are independently and uniformly chosen at random from $\mathbb{F}_q$. We also say that the entries of the matrix are *uniform i.i.d.* over $\mathbb{F}_q$.

We first review some counting results about totally random matrices over $\mathbb{F}_q$, which have been discussed in previous works (see for example [4, 16]). Recall the definition

$$\zeta_r^m = \begin{cases} (1-q^{-m})(1-q^{-m+1})\cdots(1-q^{-m+r-1}) & 0 < r \le m, \\ 1 & r = 0. \end{cases}$$

For $0 < r \le m$, it can readily be checked that ζ_r^m is the probability of the $r \times m$ totally random matrix is full rank.

For integers $0 \le k \le r, m$, define

$$\zeta_k^{m,r} \triangleq \frac{\zeta_k^m \zeta_k^r}{\zeta_k^k q^{(m-k)(r-k)}}. \tag{2.4}$$

Then $\zeta_k^{m,r}$ is the probability that the $r \times m$ totally random matrix has rank k. In particular, when $k = r$, $\zeta_k^{m,r} = \zeta_r^m$. For $m = 0, 1, \ldots, M$, let $\mathbf{G}^{(s)}$ be the $m \times M$ totally random matrix and $\mathbf{H}$ be a random matrix with M rows. Then for any $r \le M$,

$$\Pr\{\mathrm{rk}(\mathbf{G}^{(m)}\mathbf{H}) = k \,|\, \mathrm{rk}(\mathbf{H}) = r\} = \zeta_k^{m,r}. \tag{2.5}$$

Let us check the probability that a batch is decodable when its degree has a specific value. According to the algorithm BP(n), if a batch is decodable at time t, it is decodable at all time $t' > t$ until the associated linear system has no variable left. We say a batch is decodable for the first time at time t if it is decodable at time t, but is not decodable at time $t-1$.

For $s = 0, 1, \ldots, M$, let $\mathbf{G}^{(s)}$ be an $s \times M$ totally random matrix over the base field $\mathbb{F}_q$. Suppose all the batch transfer matrices $\mathbf{H}_i$ have the same distribution as $\mathbf{H}$. Define

$$\hbar_s \triangleq \Pr\left\{\mathrm{rk}\left(\begin{bmatrix}\mathbf{G}^{(1)}\\ \mathbf{G}^{(s)}\end{bmatrix}\mathbf{H}\right) = \mathrm{rk}(\mathbf{G}^{(s)}\mathbf{H}) = s\right\}, \tag{2.6}$$

$$\hbar'_s \triangleq \Pr\{\mathrm{rk}(\mathbf{G}^{(s)}\mathbf{H}) = s\}, \tag{2.7}$$

where $\mathbf{G}^{(1)}$ and $\mathbf{G}^{(s)}$ are statistically independent. Note that $\hbar_s$ is the probability that a batch with transfer matrix $\mathbf{H}$ is decodable for the first time when its degree is s. Once a batch becomes decodable, it remains to be decodable until all its contributors are decoded. Note that

$$\hbar'_s = \sum_{k \geq s} \hbar_k \tag{2.8}$$

for $0 \leq s \leq M$ and $\hbar_s = 0$ for $s > M$. As an exercise, it can be shown that

$$\hbar_s = \sum_{k=s}^{M} \frac{\zeta_s^k}{q^{k-s}} h_k \tag{2.9}$$

$$\hbar'_s = \sum_{k=s}^{M} \zeta_s^k h_k, \tag{2.10}$$

where $h_k \triangleq \Pr\{\mathrm{rk}(\mathbf{H}) = k\}$ is the rank distribution of $\mathbf{H}$. By (2.8) and (2.10), we have

$$\sum_{k \geq s} \hbar_k = \sum_{k=s}^{M} \zeta_s^k h_k. \tag{2.11}$$

Note that when the field size is large enough, e.g., $q = 2^8$, the difference between h_k and $\hbar_k$ becomes negligible. In other words, if we use $\mathrm{GF}(2^8)$ as the base field for BATS code encoding, a batch becomes decodable with high probability when its degree is reduced to the rank of the batch transfer matrix.

2.3.5 LAYERED DECODING GRAPH

In BP decoding, it is possible that more than one input packets are decodable simultaneously. Here we show that the order of substituting the decodable input packets back into the batches they contribute to does not affect the number of input packets decoded when $\mathrm{BP}(n)$ stops.

Lemma 2.1 *For a deterministic instance of BATS code of n batches, the order of the decodable input packets for substitution does not affect the time that* $\mathrm{BP}(n)$ *stops.*

Proof. Consider a deterministic instance of BATS code represented by a graph $\mathbb{T}$. Let $\mathbb{T}^0 = \mathbb{T}$. Label by L_c^0 all the decodable check nodes in $\mathbb{T}^0$ and label by L_v^0 all the variable nodes in $\mathbb{T}^0$ connected to the check nodes with label L_c^0. We repeat the above procedure as follows. For $i = 1, 2, \ldots$, let $\mathbb{T}^i$ be the subgraph of $\mathbb{T}$ obtained by removing all the nodes with labels L_c^j and L_v^j for $j < i$, as well as the adjacent edges. (The generator matrices of the check nodes are also updated as the BP decoding.) Label by L_c^i all the decodable check nodes in $\mathbb{T}^i$ and label by L_v^i all the variable nodes in $\mathbb{T}^i$ connected to the check nodes with label L_c^i. This procedure stops when $\mathbb{T}^i$ has no more decodable check nodes. Let i_0 be the index where the procedure stops, i.e., $\mathbb{T}^{i_0}$ has no decodable check nodes. The above labeling procedure is deterministic and generates unique labels for each variable nodes and check nodes.

With these labels, we generate a layered subgraph $\mathbb{T}'$ of $\mathbb{T}$. In $\mathbb{T}'$, layer j, $j = 0, 2, 4, \ldots, 2i_0 - 2$ contains all the check nodes with label $L_c^{j/2}$, and layer $j+1$, $j = 0, 2, 4, \ldots, 2i_0 - 2$ contains all the variable nodes with label $L_v^{j/2}$. Only the edges connecting two nodes belonging to two consecutive layers are preserved in $\mathbb{T}'$. By the assigning rule of the labels, it is clear that a variable node on layer $2i+1$ must connect to one check node on layer $2i$, $i = 0, 1, \ldots, i_0 - 1$, because otherwise the variable node is not decodable. Further, a check node on layer $2i$ must connect to some variable nodes on layer $2i-1$, $i = 1, \ldots, i_0 - 1$, because otherwise the check node should be on layer $2i-2$.

By the definition of decodability, any BP decoding strategy must process the variable/check nodes in $\mathbb{T}'$ following an order such that a variable/check node is processed after all its lower layer descendant variable/check nodes have been processed, and stops with the residual graph $\mathbb{T}^{i_0}$. □

2.4 PRECODING

After the BP decoding has stopped with a fraction of the input packets decoded, we can try to decode the remaining input packets using Gaussian elimination. Can we guarantee that the Gaussian elimination succeeds with a small coding overhead? The answer is actually negative if we want to have a constant decoding complexity with respect to K (in finite field operations per packet).

Consider that we want to recover all the K input packets using n batches with probability at least $1 - 1/K^c$ for some positive constant c. Similar to the analysis of LT codes (cf. [68, Proposition 1]), no matter what decoding algorithm is applied, the expected degree of a batch must be lower bounded by $c' \frac{K}{n} \log K$ for some positive constant c'. When K/n converges to a constant positive value, the expected degree of a batch is lower bounded by $c'' \log K$ for some positive constant c''.

One way to resolve the above issue is to use the *precoding* technique which has been used in Raptor codes. Suppose we have K' input packets to transmit. Before applying the batch encoding process in Section 2.1, the input packets are first encoded using a traditional erasure code (called a *precode*), where the generated K packets are called the *precoded input packets*. The

batch encoding process is applied to the K precoded input packets generated by the precode. At a destination node, the BP decoding of BATS code is first used to recover at least K' out of the K precoded input packets. The precode decoder is further applied to recover the K' input packets.

Although the two-step decoding is easier for performance analysis, in practice, its better to jointly decode the precode and the BATS code. For example, when an LDPC code can be used as the precode, the BP decoding of LDPC codes and that of BATS codes can be combined into one BP decoding process.

Precoding is particularly useful when used with *inactivation decoding*, which will be discussed in Chapter 7. Due to similar requirements, the precode of BATS codes can be designed similar to that of Raptor codes. Readers can find the detailed discussion of the precode techniques of Raptor codes in [39, 69].

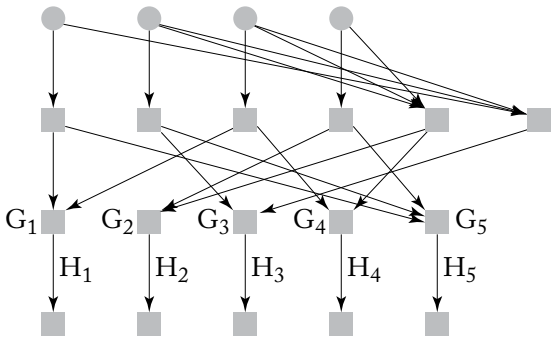


Figure 2.2: Precoding of BATS codes. Nodes in the first row represent the input packets. Nodes in the second row represent the intermediate packets generated by the precode.

2.5 PERFORMANCE METRICS

The design of the outer code and the inner code can be separated. The batch transfer matrices $\mathbf{H}_i$ capture the effects of the network, including the network coding at the intermediate nodes on the batches, i.e., the inner coding. For designing the outer code, it is not necessary to know the details of the packet loss, network topology, et al. Evidently, the knowledge of the distribution of $\mathbf{H}_i$, $i = 1, 2, \ldots$ is sufficient. As we have discussed, the solvability of a batch depends only on the rank of the batch transfer matrix. Hence, the knowledge of the rank distribution of the batch transfer matrices is indeed already sufficient.

Let us take a closer look at how the batch transfer matrices affect the performance of BATS codes. To evaluate the performance of a BATS code, we define several notations that are related to transfer matrices. Suppose a destination node decodes successfully after receiving n batches with transfer matrices $\{\mathbf{H}_i, i = 1, \ldots, n\}$. The first is about the *communication cost* of

a batch i, denoted by CC_i. There are different ways to measure the communication cost of a batch, depending on the system design constraints. For example, we may use the number of columns of $\mathbf{H}_i$ as CC_i, which determines the cost for receiving the batch at the destination node. Alternatively, we may use the total number of packets belonging to the batch i transmitted by all the network nodes as CC_i if we care about the *total cost* for transmitting a batch from the source node to the destination node. The communication cost is affected by the design of the inner code, and can be separated from the design of the outer code.

Define the *design coding rate* of the outer code of a BATS code as K/n, and define the *(normalized) coding overhead* as

$$\mathrm{CO} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i) - \frac{K}{n}.$$

We should design batch encoding and decoding schemes such that the coding overhead is as small as possible, which can be understood from a linear operator channel point-of-view.

The operation of the network on the batches in (2.1) can be modeled as a channel with input $\mathbf{X}_i$ and output $\mathbf{Y}_i = \mathbf{X}_i\mathbf{H}_i$, $i = 1, 2, \ldots$, where the instance of $\mathbf{H}_i$, regarded as the state of the channel, is known by the receiver. This channel model is called a *linear operator channel (LOC) with receiver side channel state information*. (Similar channel models have been studied without the channel state information [33, 74].) The LOC is not necessary to be memoryless since $\mathbf{H}_i$, $i = 1, 2, \ldots$ are not assumed to be independent. With receiver side channel state information, the capacity of the LOC can be easily characterized. Consider that

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i)}{n} \xrightarrow{P} \bar{h}.$$

The channel capacity of the above channel is upper bounded by $\bar{h}$ and the upper bound can be achieved by random linear codes [99]. As a channel code for the LOC, the maximum achievable rate of the outer code of a BATS code is bounded by $\bar{h}$ for any inner code with the average rank of the transfer matrices converging to $\bar{h}$.

For a BATS code, the *coding rate normalized by the communication cost* is

$$\mathrm{CR} = \frac{K}{\sum_{i=1}^{n} \mathrm{CC}_i} = \frac{\frac{1}{n}\sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i) - \mathrm{CO}}{\frac{1}{n}\sum_{i=1}^{n} \mathrm{CC}_i}.$$

When n tends to infinity, we have

$$\lim_{n\to\infty} \mathrm{CR} = \frac{\bar{h} - \lim_{n\to\infty} \mathrm{CO}}{\overline{\mathrm{CC}}},$$

where $\overline{\mathrm{CC}} = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathrm{CC}_i$. From the above analysis, we obtain two fundamental design guidelines of BATS codes for both the outer coding and the inner coding, respectively.

1. The outer code should minimize CO (or $\lim_{n\to\infty}$ CO) by designing a proper degree distribution. Suppose that the empirical rank distribution of $\mathbf{H}_i$, $i = 1, 2, \ldots$ converges in probability to $(h_0, h_1, \ldots, h_M)$, so that

$$\lim_{n\to\infty} \frac{\sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i)}{n} \xrightarrow{P} \sum_{i=0}^{M} i h_i \triangleq \bar{h}.$$

   As we will discuss later, we have an outer code (a degree distribution) that can achieve a rate very close to $\bar{h}$, i.e., $\lim_{n\to\infty}$ CO can be very small and close to zero.

2. The design of the inner code is about the operations at an intermediate node, which affect the distribution of $\mathbf{H}_i$, $i = 1, 2, \ldots$, and hence the rank distribution $(h_0, h_1, \ldots, h_M)$. The inner code affects both $\bar{h}$ and $\overline{\mathrm{CC}}$. Suppose the outer code has $\lim$ CO $= 0$. To maximize $\lim$ CR, we should maximize $\bar{h}/\overline{\mathrm{CC}}$. For a finite-length code, we want to design an inner code that maximizes $\sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i)/\sum_{i=1}^{n} \mathrm{CC}_i$, which is the average rank per communication cost.

2.6 SPECIAL CASE: LT CODES

When the batch size is one, BATS codes as described above become LT codes. In this case, since each batch has only one coded packet, network coding at the intermediate nodes becomes forwarding as we do not allow coding across batches. Then h_0, the probability that the batch transfer matrix has rank zero, can be regarded as the end-to-end packet loss rate.

Due to the random generator matrix, the batch degree distribution $\boldsymbol{\Psi}$ is not the same as the degree distribution usually referred to for LT codes. The degree of a batch may be larger than the degree of the coded packet[2] in the batch because certain entries of the generator matrix may be equal to 0. For a batch with degree d, the degree of the coded packet in the batch is k $(k \le d)$ with probability $\binom{d}{k}(1-q^{-1})^k q^{-(d-k)}$.

Our analysis (to be provided) uses the degree distribution of batches, which can be converted into the degree distribution of coded packets. While this can be done, we have a simpler approach to applying our analytical results to LT codes with respect to the degree distribution of the coded packets.

When $M = 1$, to make $\boldsymbol{\Psi}$ the same as the coded packet degree distribution, we can use the generator matrix with all entries being the identity of the base field to replace the random generator matrix. Then the degree of a batch is the same as the degree of the coded packet in the batch. Redefining (2.6) and (2.7) for $\mathbf{G}^{(s)}$ containing only the identity of the base field, we have

$$\hbar_0 = h_0, \hbar'_0 = 1 \text{ and } \hbar_1 = \hbar'_1 = h_1.$$

[2] A coded packet can be expressed as a linear combination of the input packets. The degree of the coded packet is defined as the number of non-zero coefficients in the linear combination.

So when $M = 1$, substituting the above values of $\hbar_s$ and $\hbar'_s$ into the formulae to be obtained in this paper, we obtain the corresponding results for LT codes with respect to the degree distribution of the coded packets.

2.7 SUMMARY AND PERFORMANCE COMPARISON

BATS codes, introduced in [95, 97], generalize both fountain codes and random linear network coding. If network coding is applied to all the packets generated by an LT code indistinguishably, the degrees of the received packets will be changed so that the efficient decoding algorithm of LT codes would fail. BATS codes resolve this issue by allowing only network coding for packets belonging to the same batch so that the degrees of batches are not changed by network coding at the intermediate nodes. Sufficient network coding gain can already be obtained by using very modest values for M (e.g., 16 or 32).

Other approaches using fountain codes for multi-hop networks incur much higher transmission delay, intermediate node storage cost, or higher decoding computation cost (see Table 1.1). Compared with baseline random linear network coding schemes, BATS codes not only have lower encoding/decoding complexity, but also smaller coefficient vector overhead and intermediate node caching requirement (see Table 1.1). Compared with other low-complexity random linear network coding schemes like EC codes and L-chunked codes, BATS codes generally achieve higher (but not much higher) rates and have the extra feature that an unlimited number of batches can be generated.

BATS codes can be used with different combinations of encoding/decoding options at the source/destination nodes.

1. BATS encoding (no precoding) and BP decoding—This combination is suitable when only a fraction of the input packets are required to be decoded.

2. BATS encoding with precoding and BP decoding—This combination can decode the BATS code and the precode jointly, and it enables all the input packets to be recovered.

3. BATS encoding with precoding and inactivation decoding (to be discussed in Chapter 8)—This combination enables all the input packets to be recovered, and it uses extra computation to reduce the coding overhead.

CHAPTER 3

First BATS Code Protocol

In this chapter, we introduce a simple BATS code enabled network protocol called *BATS-Pro-0*, which demonstrates how to use BATS codes for network communication.

In this book, we use 0 as the starting index for vectors and matrices. For a vector $\mathbf{a}$ of length k, we denote by $\mathbf{a}[i:j]$ $(0 \le i \le j \le k-1)$ the subvector of $\mathbf{a}$ from the i-th to the j-th component. We also write $\mathbf{a}[i] = \mathbf{a}[i:i]$, $\mathbf{a}[:] = \mathbf{a}[0:k-1]$ and $\mathbf{a}[i:] = \mathbf{a}[i:k-1]$ to simplify notations.

For an $m \times n$ matrix $\mathbf{A}$, we denote by $\mathrm{rk}(\mathbf{A})$ its rank and by $\mathbf{A}[i_1:i_2, j_1:j_2]$ $(i_1 \le i_2, j_1 \le j_2)$ the submatrix of $\mathbf{A}$ formed by the entries from the i_1-th to i_2-th rows and from the j_1-th to j_2-th columns. We also write $\mathbf{A}[i, j_1:j_2] = \mathbf{A}[i:i, j_1:j_2]$, $\mathbf{A}[i, j:] = \mathbf{A}[i:i, j:n-1]$, $\mathbf{A}[:, j] = \mathbf{A}[0:m-1, j:j]$, etc.

3.1 BATS PROTOCOL STACK

Before introducing the details of BATS-Pro-0, we first introduce a general framework of using BATS codes, called the BATS protocol stack. All the applications of BATS codes for network communications can be designed following this framework.[1] The BATS protocol stack has the following five layers, from top to bottom:

- application layer;
- transport layer;
- network layer;
- link layer; and
- physical layer.

Although the layering is similar to that of the Internet protocol stack, the operations of the transport and the network layers are totally different from those of TCP/IP. See Figure 3.1 for an illustration of the protocol stack in a line network with three hops. We give a general description of each layer below.

The application layers at the source node and the destination node represent the two parties that want to communicate. At the source node, the application layer sends a file transmission

[1] BATS codes also find applications in data storage, caching et al., where the framework of using BATS codes is different.

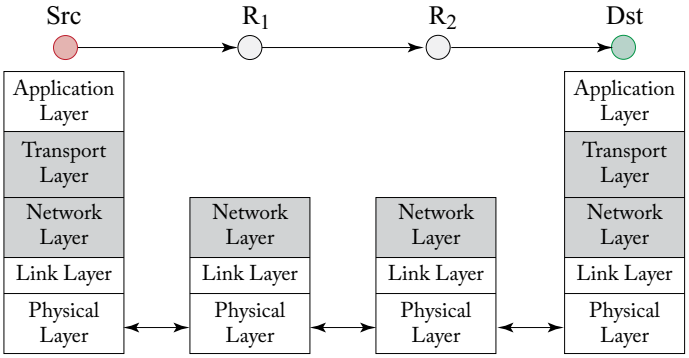


Figure 3.1: BATS protocol stack.

request through an interface of the BATS protocol stack. The file transmission request is first processed at the transport layer.

The transport layers at the source node and the destination node work together. They have two major functions related to the outer code: batch generation and BATS code decoding. At the source node, upon receiving a file transmission request from the application layer, the transport layer will generate batches using the file. These batches will then be forwarded to the network layer. At the destination node, upon receiving batches from the network layer, the transport layer will try to decode the batches to recover the file. For a successfully decoded file, the transport layer will forward the file to the application layer, and optionally send an acknowledgement back to the source node.

The network layer implements the inner code (network coding). Upon receiving packets from either the transport layer or the link layer, the network layer first checks if the current node is one of the destination nodes. If so, the network layer will forward the packet to the transport layer. The network layer then checks whether there is another destination node in the network to which it needs to forward recorded packets. If so, the network layer applies network coding to all the received packets belonging to the same batch, and then forwards the recoded packets to the link layer. See Figure 3.2 for a flow chart for the network layer operations.

We note that for an intermediate network node, e.g., node R_1 or R_2 in Figure 3.1, neither the transport layer nor the application layer exists. Therefore, we do not include them in the figure for these nodes.

As discussed above, the transport layer and the network layer implement respectively the outer code and the inner code of a BATS code. Although the link layer and the physical layer do not directly implement any part of the encoding/decoding algorithm of a BATS code, the properties of BATS codes affect the design of the link layer and the physical layer.

- First, the link layer does not need to retransmit a recoded packet, and hence the acknowledgement indicating the receiving of each recoded packet is not necessary. The link layer

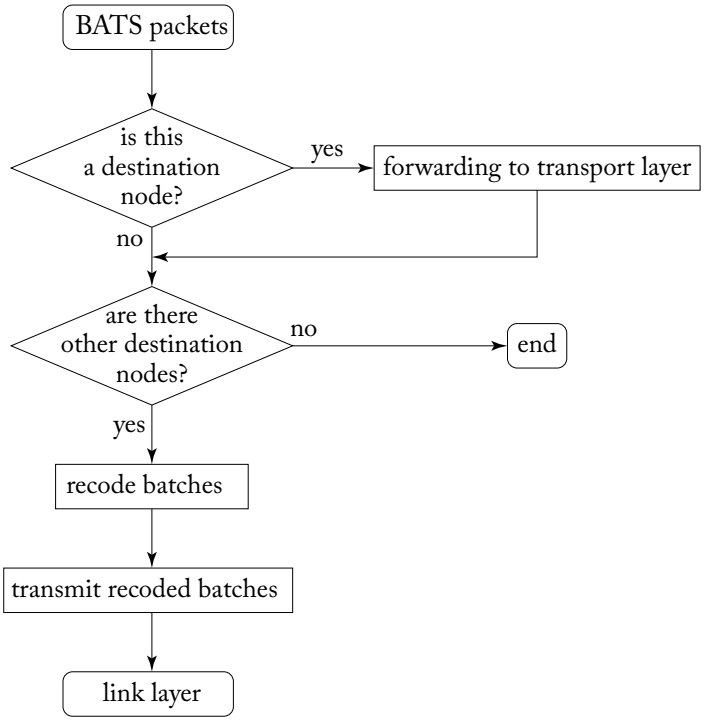


Figure 3.2: Flow chart of the network layer operations.

may, however, employ a feedback mechanism for other purposes, e.g., link loss rate estimation, which is useful in the design of recoding schemes.

- Second, the physical layer does not necessarily have a very low packet loss rate. Most existing wireless communication systems use a large transmission power to guarantee a very low packet loss rate in the worst case of the dynamic channel state to ensure that TCP/IP works properly. In a BATS protocol, however, the physical layer can be redesigned to provide a better tradeoff between the transmission power and the throughput for multi-hop networks.

The link/physical layer of a BATS protocol can be built upon existing wireless communication protocols like IEEE 802.11 wireless LAN and IEEE 802.15.4 wireless PAN so that the BATS protocol can make use of existing wireless transceiver hardware/software.

3.2 BATS-PRO-0 SPECIFICATION

We introduce BATS-Pro-0, the first implementation of the BATS protocol stack, which has the following features.

- The link layer feedback is not required.
- The buffer requirement at the intermediate network nodes can be as small as one batch.
- The recoded packets are generated by *random linear network coding*.
- The recoded packets of a batch are transmitted consecutively.
- The achievable rate is close to optimal when the batch size is relatively large.

BATS-Pro-0 was first introduced in [101].

Let K_0 and T_0 be two positive integers. At the source node, the input file is separated into K_0 packets of length T_0 octets. Therefore, K_0T_0 determines the size of the input file.

In BATS-Pro-0, the batch size M can be a small power of 2, e.g., 8, 16, 32, and 64. Using a larger batch size may not have much throughput advantage, but it significantly increases the computation cost and coefficient vector overhead.

The base field for BATS code encoding is GF(256) (i.e., $q = 256$), which takes 8 bits or an octet to represent a field symbol. With GF(256) as the base field, a batch becomes decodable with high probability when its degree is reduced to the rank of the batch transfer matrix.

3.2.1 COEFFICIENT VECTORS

If the inner code uses the same base field as the outer code, the coefficient vector would have M field symbols from GF(256), i.e., M octets. However, it is not necessary that the inner code uses the same finite field as the outer code. Consider a subfield of GF(256) with q_m elements. If the inner code uses coefficients from GF(q_m) only, the coefficient vector can be represented by M symbols from GF(q_m). For example, when $q_m = 2$, the coefficient vector can be represented in $M/8$ octets.

In addition to the smaller coefficient vector overhead, using a smaller finite field for the inner code can reduce the computation cost for recoding. For BATS-Pro-0, q_m can be 2^k, $k = 1, 2, \ldots, 8$. We choose $q_m = 2$ in the scenarios that the computation power of the network nodes is extremely low, or the packet is very short.

3.2.2 TRANSPORT LAYER

The main operations of the transport layer are illustrated in Figure 3.3. The transport layer is responsible for the encoding and decoding of batches, and performs an integrity check of the transmitted file.

The transport layer takes the K_0 packets of length T_0 octets as the input. An optional precode is first applied to these input packets to generate $K > K_0$ precoded input packets. The transport layer then generates batches using the batch encoding method in Section 2.1. The M packets of a batch forwarded to the link layer have a format given in Figure 3.4.

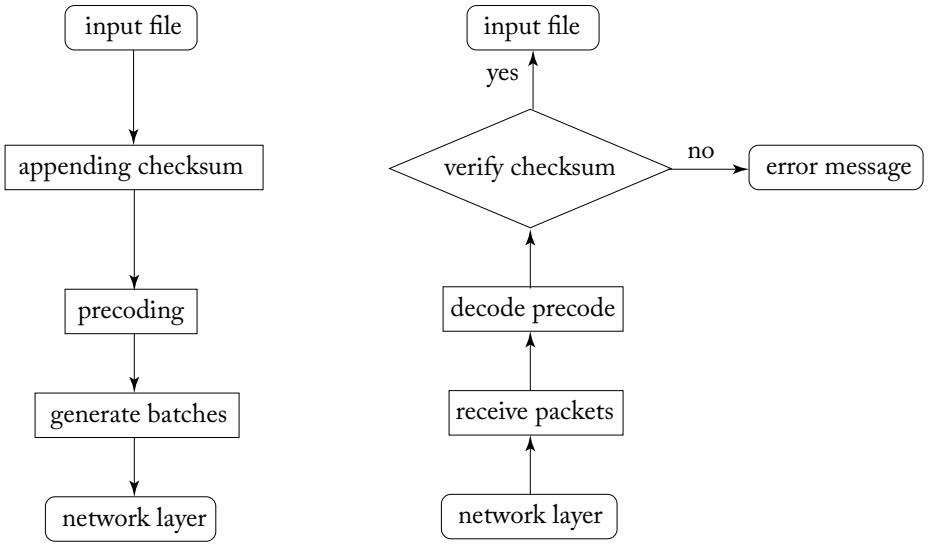


Figure 3.3: BATS-Pro-0 transport layer.

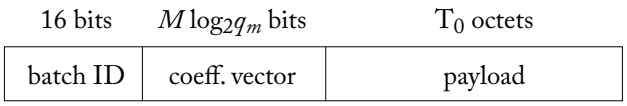


Figure 3.4: The format of a transport-layer packet. A batch ID usually has one or two octets. Taking $M = 16$ and $q_m = 2$ as example, a coefficient vector has two octets.

- The first 16 bits are used for the batch ID to distinguish packets belonging to different batches. The batch ID can also used as the seed to recover the pseudorandomness used in the batch encoding and decoding.
- The coefficient vector consists of M field symbols from GF(q_m), each being represented by $\log_2 q_m$ bits. Initially, the M coefficient vectors of a batch form the $M \times M$ identity matrix over GF(q_m).
- The payload has T_0 octets.

At a destination node, the transport layer decodes the original input file using the batches it has received. The batch IDs are used for distinguishing the packets belonging to different batches and for recovering the generator matrices of the batches. The coefficient vectors of the received packets belonging to the same batch are used to recover the batch transfer matrix. When there is no precoding, the BP decoding of BATS codes (introduced in Section 2.2) is applied

to recover the K_0 input packets or a fraction of them. When precoding is applied at the source node, the decoding approaches introduced in Section 2.4 can be applied to decode the K_0 input packets.

3.2.3 NETWORK LAYER

The operations of the network layer of BATS-Pro-0 follows the flow chart in Figure 3.2, with the details of the batch receiving, recoding, and transmitting operations described as follows.

Batch Receiving

The batch receiving module of the network layer has the flow chart given in Figure 3.5. The design of this module is based on the assumption that the packets are received in the correct order though some of them may be lost along the way. This can be guaranteed by the batch transmitting module to be described. The batch receiving module has two variables: *current batch* and *readied batch*. Initially, both variables are *null*.

The variable *current batch* is equal to the ID of the batch that is being received by the network layer. When a packet is received by the network layer, the batch ID is first checked.

- If the batch ID is equal to the value of *current batch*, the packet is saved in the storage.
- If the batch ID is not equal to the value of *current batch* (which means that the packet just received belongs to a new batch), in addition to saving the packet, the following operations are performed: First, *readied batch* takes the value of *current batch*, and *current batch* takes the value of the batch ID of the newly received packet. Second, a recoding procedure described below is applied to the packets in the storage with batch ID equal to *readied batch* to generate M recoded BATS packets.

Batch Recoding

For each batch such that the network layer receives at least one packet, the recoding generates exactly M packets for the batch. The M recoded packets, generated using *random linear network coding (RLNC)*, are M different random linear combinations of the received packets belonging to that batch, where the coefficients are chosen from GF(q_m) uniformly at random.

Batch Transmitting

BATS-Pro-0 transmits the M recoded packets for a batch consecutively once they are generated. This way, at every non-source node in the network, the network layer receives all the packets belong to one batch before the packets belonging to the next batch.

3.3 PERFORMANCE OF BATS-PRO-0

To simplify the discussion, we consider a line network of length l illustrated in Figure 3.6, and assume that the network links are *homogeneous*, i.e., they have the same transmission rate

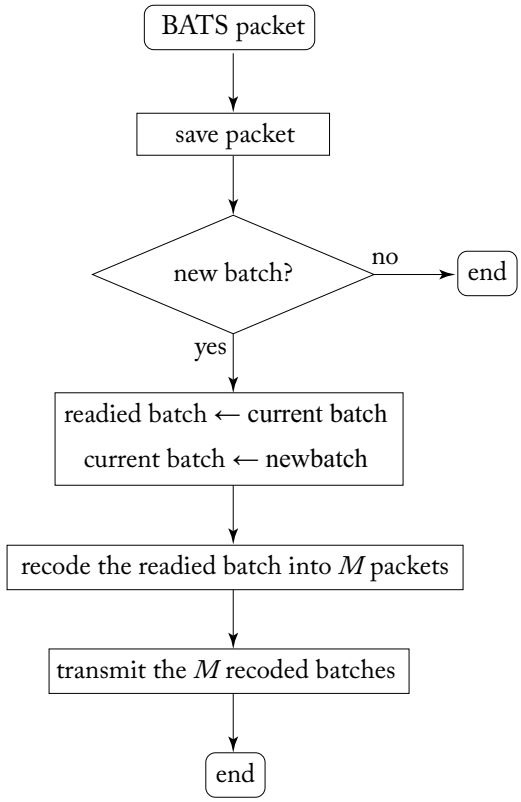


Figure 3.5: Recoding module of the network layer.

and packet loss probability ϵ. When there are no computation and storage constraints at the intermediate network nodes, the min-cut capacity of the line network with length l is $1-\epsilon$ packet per use for any $l > 0$. Here 1 use of the network means the use of each network link at most once; transmitting nothing on a network link in a particular time slot is allowed.

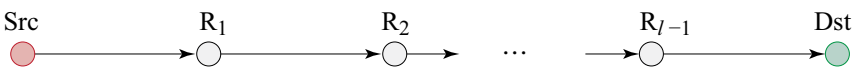


Figure 3.6: Line network of length l.

3.3.1 INTERMEDIATE STORAGE AND TRANSMISSION DELAY

In BATS-Pro-0, the source node generates batches and transmits a packet in each time slot. The M packets of a batch are transmitted in M consecutive time slots, and the batches are transmitted according to the order in which they are generated. In this subsection, we perform an analysis of the storage requirement and transmission delay at an intermediate node under the assumption of zero computation latency.

In the first M time slots, node R_1 can potentially receive M packets belonging to the first batch. In the first $M-1$ time slots, node R_1 saves the received packets in its buffer but transmits nothing. In the M-th time slot, node R_1 generates M coded packets by applying random linear network coding to the packets in its buffer and the packet just received, all of which belong to the same batch. After generating the M coded packets, the original received packets in the buffer are deleted. Node R_1 then transmits one of the coded packets and saves the remaining $M-1$ coded packets in its buffer. In each of the following $M-1$ time slots, node R_1 transmits one of the remaining coded packets of the first batch and then deletes that packet in the buffer immediately. During these time slots, if node R_1 receives a new packet (belonging to the second batch), the new packet is saved in the buffer. From the $2M$-th to the $(3M-1)$-th time slot, node R_1 repeats the above operations on the second batch, so on and so forth. All the other intermediate nodes apply the same operations as node R_1.

From the above analysis, each intermediate node caches at most $M-1$ packets in the buffer. There is a delay for each intermediate node: node R_i can only start to receive packets after $(i-1)(M-1)$ time slots. For a network of fixed length, the delay is neglectable compared with the total transmission time when the file size is large.

3.3.2 RANK DISTRIBUTION FOR RLNC RECODING

For BATS-Pro-0, the network operations on each batch are independent, and the rank distributions of all the batches are the same. Now let us study the rank distribution of a batch for RLNC Recoding. Let π_i, $i = 0, 1, \ldots, l$ be the rank distribution of a batch at node R_i where $\mathrm{R}_0 = \mathrm{Src}$ and $\mathrm{R}_l = \mathrm{Dst}$. For RLNC Recoding, the rank of a batch at node R_i, $i = 0, 1, \ldots, l$ forms a homogeneous Markov chain. Denote by $\mathbf{P}_{\mathrm{R}}$ the probability transition matrix of this Markov chain. Then

$$\pi_i = \pi_0 (\mathbf{P}_{\mathrm{R}})^i,$$

with $\pi_0[M] = 1$. Thus to characterize the rank distribution π_i, we only need to know $\mathbf{P}_{\mathrm{R}}$. Note that $\mathbf{P}_{\mathrm{R}}$ is lower-triangular. Write

$$\zeta_r^m(\tilde{q}) = \begin{cases} (1-\tilde{q}^{-m})(1-\tilde{q}^{-m+1})\cdots(1-\tilde{q}^{-m+r-1}) & 0 < r \le m, \\ 1 & r = 0, \end{cases}$$

and

$$\zeta_k^{m,r}(\tilde{q}) \triangleq \frac{\zeta_k^m(\tilde{q})\zeta_k^r(\tilde{q})}{\zeta_k^k(\tilde{q})\tilde{q}^{(m-k)(r-k)}}.$$

Lemma 3.1 *For* $0 \le j \le i \le M$, $\mathbf{P}_{\mathrm{R}}[i,j] = \sum_{r=j}^{M} \binom{M}{r}(1-\epsilon)^r \epsilon^{M-r} \zeta_j^{i,r}(q_m)$.

Proof. Consider that a node receives a batch of rank i and generates M recoded packets using RLNC recoding. The probability that r $(r \ge j)$ packets of the batch is received at the node in the next hop is $\binom{M}{r}(1-\epsilon)^r \epsilon^{M-r}$. Since each recoded packets is the linear combination of the received packets with the coefficients chosen uniformly at random from $\mathrm{GF}(q_m)$, the probability that the r received packets have rank j is $\zeta_j^{i,r}(q_m)$. The lemma is proved by considering all $r \ge j$. □

3.3.3 OPTIMALITY WHEN BATCH SIZE IS LARGE

According to the following lemma, when M tends to infinity, the normalized expected rank will converge to $1-\epsilon$. Therefore, for line networks with link erasure probability ϵ, BATS-Pro-0 can achieve a normalized rate very close to $1-\epsilon$ when M is sufficiently large.

Lemma 3.2 *For* $l \ge 1$, $\lim_{M\to\infty} \sum_r r\pi_l[r]/M = 1-\epsilon$.

Suppose that each lost packets on a network link is represented by an all-zero packet, and hence each node always receives M packets for a batch. Denote by $\mathbf{H}$ the transfer matrix of a batch at a network node. The recoding at the node is given by a totally random $M \times M$ matrix Φ over $\mathrm{GF}(q_m)$. The transmission of the recoded batch through a network link can be modeled by an $M \times M$ random *diagonal* matrix $\mathbf{E}$ with independent components, where a diagonal component is 0 with probability ϵ and is 1 with probability $1-\epsilon$. Hence, the transfer matrix for the batch received at the network node on the next hop can be expressed as

$$\mathbf{H}' = \mathbf{H}\Phi\mathbf{E}.$$

To prove the above lemma, we only need to show that if $\lim_{M\to\infty} \mathbb{E}[\mathrm{rk}(\mathbf{H})]/M \ge 1-\epsilon$, then $\lim_{M\to\infty} \mathbb{E}[\mathrm{rk}(\mathbf{H}')]/M = 1-\epsilon$. First, we have $\mathbb{E}[\mathrm{rk}(\mathbf{H}')] = \mathbb{E}[\mathrm{rk}(\mathbf{H}\Phi\mathbf{E})] \le \mathbb{E}[\mathrm{rk}(\mathbf{E})] = (1-\epsilon)M$. We then prove that for any $\delta > 0$,

$$\lim_{M\to\infty} \Pr\left\{\frac{\mathrm{rk}(\mathbf{H}')}{M} \ge 1-\epsilon-\delta\right\} = 1, \tag{3.1}$$

which implies $\lim_{M\to\infty} \mathbb{E}[\mathrm{rk}(\mathbf{H}')]/M \ge 1-\epsilon$.

It remains to prove (3.1). Toward this end, let $t = 1-\epsilon-\delta$, $\delta > 0$ and consider

$$\begin{aligned}
\Pr\left\{\frac{\mathrm{rk}(\mathbf{H}')}{M} \ge t\right\} &= \Pr\left\{\frac{\mathrm{rk}(\mathbf{H}\Phi\mathbf{E})}{M} \ge t\right\} \\
&\ge \Pr\left\{\frac{\mathrm{rk}(\mathbf{H}\Phi\mathbf{E})}{M} \ge t \middle| \frac{\mathrm{rk}(\mathbf{H})}{M} \ge t+\frac{\delta}{2}, \frac{\mathrm{rk}(\mathbf{E})}{M} \ge t\right\} \\
&\quad \times \Pr\left\{\frac{\mathrm{rk}(\mathbf{H})}{M} \ge t+\delta/2\right\} \Pr\left\{\frac{\mathrm{rk}(\mathbf{E})}{M} \ge t\right\},
\end{aligned} \tag{3.2}$$

where the last two terms on the RHS of (3.2) converge to 1 as $M \to \infty$. Since

$$\begin{aligned}
&\Pr\left\{\frac{\mathrm{rk}(\mathbf{H}\Phi\mathbf{E})}{M} \geq t \middle| \frac{\mathrm{rk}(\mathbf{H})}{M} \geq t + \frac{\delta}{2}, \frac{\mathrm{rk}(\mathbf{E})}{M} \geq t\right\} \\
&\geq \Pr\left\{\mathrm{rk}(\mathbf{H}\Phi\mathbf{E}) = \lceil Mt \rceil \middle| \mathrm{rk}(\mathbf{H}) = \lceil M(t+\delta/2) \rceil, \mathrm{rk}(\mathbf{E}) = \lceil Mt \rceil\right\} \\
&= \zeta_{\lceil Mt \rceil}^{\lceil M(t+\delta/2) \rceil}(q_m),
\end{aligned}$$

where the equality follows from (2.5) and $\zeta_{\lceil Mt \rceil}^{\lceil M(t+\delta/2) \rceil}(q_m) \to 1$ as $M \to \infty$, the RHS of (3.2) converges to 1 as $M \to \infty$.

3.3.4 ACHIEVABLE RATE FOR FIXED BATCH SIZE

We, however, are more interested in the performance for small values of M, which can be characterized numerically. We calculate the normalized expected rank $\sum_r r\pi_l[r]/M$ for $\epsilon = 0.2$ and $q_m = 2$ or 256 in Figure 3.7.

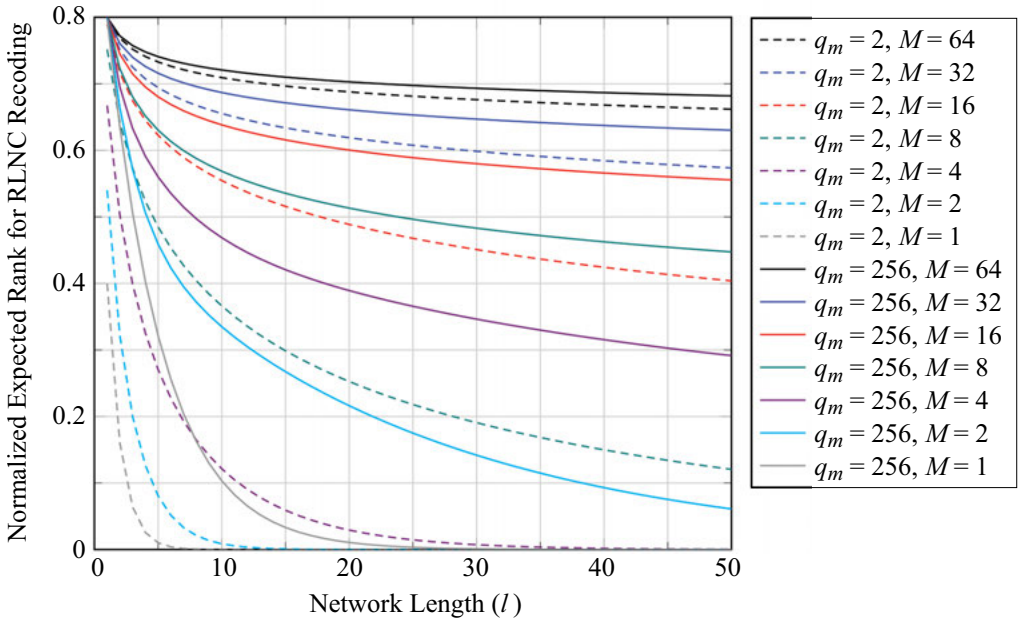


Figure 3.7: Expected ranks of the chunk transfer matrices for line networks with BATS-Pro-0 where $\epsilon = 0.2$ and $l = 1, \ldots, 50$.

When $M = 1$, the outer code becomes an LT/Raptor code, and the BATS-Pro-0 recoding becomes multiplying each received packet by a random coefficient, which is equivalent to forwarding from the performance point of view. Therefore, the effective end-to-end packet loss rate is

$$\pi_l[0] = 1 - (1-\epsilon)(1 - q_m^{-1}(1-\epsilon) - \epsilon)^{l-1}.$$

When q_m is very large, the loss rate is very close to $1-(1-\epsilon)^l$, which is the end-to-end packet loss rate when the packets are forwarded without multiplying by a random coefficient. The achievable rate for the length-l line network when $M=1$ is $(1-\epsilon)^l$, which decreases quickly as the network length increases.

Compared with $M=1$, the normalized expected rank decreases slowly as the network length increases when $M \geq 2$. For a fixed network length, Figure 3.7 illustrates the tradeoff between the batch size and the maximum achievable rates of BATS codes (without considering the coefficient vector overhead). We see that once M is larger than 32, using a larger batch size only brings a marginal rate gain (but increases significantly the computation cost).

We also note that when $M=32$ or 64, the performances of $q_m=2$ and $q_m=256$ are very close to each other. In other words, it is feasible to use the binary field for the inner code when the batch size is not too small. Note that using the binary field for recoding can reduce the coefficient vector overhead and the recoding computation cost.

Now we look at the requirement for the batch size M for different values of the packet loss rate ϵ. Suppose we want to achieve the throughput $0.7(1-\epsilon)$ in a homogenerous line network with 10 hops. From the first two rows of Table 3.1, we see that the higher the loss rate, the larger the batch size is required to achieve the desired throughput, and the increase is faster than a linear function of $1/(1-\epsilon)$ (see the third row in Table 3.1). When $\epsilon=0.9$, for example, a batch size of 220 is required, but a large batch size would result in a large coefficient vector overhead and computation cost. We will discuss how to resolve this issue in Chapter 4.

Table 3.1: The batch size M for achieving a throughput of $0.7(1-\epsilon)$ in a homogenerous line network with 10 hops. Here $q=256$.

ϵ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
M	4	8	12	18	26	39	59	99	220
$M(1-\epsilon)$	3.6	6.4	8.4	10.8	13	15.6	17.7	19.8	22

CHAPTER 4

Advanced Recoding Techniques

In Chapter 3, we discussed a basic recoding scheme called BATS-Pro-0, where the network layer generates M (same as the batch size) recoded packets for each batch using RLNC. Though it is capacity achieving when $M \to \infty$, we need a better scheme for a relative small batch size M.

In this chapter, we study how to improve RLNC at network nodes by (i) generating a number of recoded packets that may be different from M, and (ii) using simpler approaches to generate the recoded packets. The new recoding scheme improves the performance when the recoding field size is small, and reduces the recoding computation cost. Moreover, by increasing the number of recoded packets, a small batch size can be applied even in the case of high packet loss rate.

We use line networks as examples to discuss our recoding techniques, though these techniques also apply to general networks. For a node u in a line network, we use u^+ and u^- to denote the node at the next hop (closer to the destination node) and the node at the previous hop (closer to the source node), respectively. For real numbers x and y, denote their minimum and maximum by $x \wedge y$ and $x \vee y$, respectively.

4.1 PROPER LINEAR RECODING

Linear network coding on packets belonging to the same batch will be referred to as *linear recoding*. With linear recoding at every intermediate node, the end-to-end transformation of a batch is a linear operation. We first give a general characterization of linear recoding. Recall that we use GF(q_m), a subfield of the base field, for linear recoding.

Suppose a network node receives k packets of a batch with the coefficient matrix $\mathbf{H}$, which is an $M \times k$ matrix over GF(q_m) formed by juxtaposing the coefficient vectors of the k packets. The rank of $\mathbf{H}$ is r, $0 \le r \le M$, which is also called the rank of the batch at the node.

A linear recoding scheme takes both the received packets and an integer M' as the input, and generates M' recoded packets with the coefficient matrix

$$\mathbf{H}' = \mathbf{H}\Phi,$$

where Φ is a $k \times M'$ matrix over GF(q_m) called the *recoding generator matrix*. Note that Φ can be deterministic or random. If we can find r linearly independent columns of $\mathbf{H}$, denoted by $\mathbf{H}^*$,

the linear coding scheme can be further written as

$$\mathbf{H}' = \mathbf{H}^* \Phi^*,$$

where Φ^* is an $r \times M'$ matrix over GF(q_m) called the *reduced recoding generator matrix*. The matrices Φ and Φ^* both characterize a recoding scheme. For the RLNC recoding discussed in Chapter 3, Φ_R (or Φ_R^*) is a totally random matrix over GF(q_m).

Same as the last chapter, we use a binary $M' \times M'$ diagonal matrix $\mathbf{E}$ to indicate packet losses for the M' recoded packets when they are transmitted to the node at the next hop, where a "0" in the diagonal indicates a packet loss and a "1" in the diagonal indicates a correctly received packet. We are concerned about the rank of the batch received at the node at the next hop, given by rk($\mathbf{H}'\mathbf{E}$). Since $\mathbf{H}^*$ is of full column rank, we have

$$\text{rk}(\mathbf{H}'\mathbf{E}) = \text{rk}(\mathbf{H}^*\Phi^*\mathbf{E}) = \text{rk}(\Phi^*\mathbf{E}),$$

i.e., the rank of the batch received at the node at the next hop is rk($\Phi^*\mathbf{E}$), a random variable that depends on r, M', and the elements of Φ^* and $\mathbf{E}$. The diagonal elements of $\mathbf{E}$, representing the packet loss pattern of the transmitted packets, can be assumed to be independent of both $\mathbf{H}$ and Φ^*. In all of our recoding approaches, the choice of the rM' elements of Φ^* is independent of $\mathbf{H}$ given its rank.

4.1.1 GENERAL GUIDELINES

We provide some general guidelines about how to design recoding schemes. Consider two batches, where the first batch has the transfer matrix rank equal to r_1 and the number of recoded packets equal to M_1', and the second batch has the transfer matrix rank equal to r_2 and the number of recoded packets equal to M_2'. Let $r_1 = \text{rk}(\mathbf{H}_1)$ and $r_2 = \text{rk}(\mathbf{H}_2)$.

Definition 4.1 A linear recoding scheme is said to be *proper* if it satisfies the following three conditions.

1. When $r_1 > r_2$ and $M_1' = M_2'$, the first batch potentially has a higher rank at the network node at the next hop.
2. When $r_1 = r_2$ and $M_1' = M_2'$, both batches should have the same rank distribution at the network node at the next hop.
3. When $r_1 = r_2$ and $M_1' > M_2'$, the first batch potentially has a higher rank at the network node at the next hop.

Let us explain these three conditions and discuss how to design recoding schemes to satisfy these conditions.

We first consider the case $r_1 > r_2$ and $M_1' = M_2' = M'$. The recoding scheme generates M' recoded packets for both batches using the reduced recoding generator matrices Φ_1^* and Φ_2^*, respectively, where Φ_1^* is $r_1 \times M'$ and Φ_2^* is $r_2 \times M'$. The first condition requires that for any binary $M' \times M'$ diagonal matrix $\mathbf{E}$,

$$\Pr\{\text{rk}(\Phi_1^*\mathbf{E}) \geq j\} \geq \Pr\{\text{rk}(\Phi_2^*\mathbf{E}) \geq j\}, \quad j = 0, 1, \ldots, M. \tag{4.1}$$

We also write (4.1) as

$$\text{rk}(\Phi_1^*\mathbf{E}) \succcurlyeq \text{rk}(\Phi_2^*\mathbf{E}).$$

When Φ_2^* is a submatrix of Φ_1^*, we have $\Pr\{\text{rk}(\Phi_1^*\mathbf{E}) \geq \text{rk}(\Phi_2^*\mathbf{E})\} = 1$, and hence the first condition is satisfied.

We then consider the case $r_1 = r_2$ and $M_1' = M_2'$. To satisfy the second condition, we can use the same reduced recoding generator matrices for both batches.

Now we consider the case that $r_1 = r_2$ and $M_1' > M_2'$. The recoding scheme generates M_1' and M_2' recoded packets using the reduced recoding generator matrices Φ_1^* and Φ_2^*, respectively, where Φ_1^* is $r_1 \times M_1'$ and Φ_2^* is $r_2 \times M_2'$. The third condition says that if $M_1' > M_2'$, then for any binary $M_1' \times M_1'$ diagonal matrix $\mathbf{E}_1$,

$$\text{rk}(\Phi_1^*\mathbf{E}_1) \succcurlyeq \text{rk}(\Phi_2^*\mathbf{E}_2), \tag{4.2}$$

where $\mathbf{E}_2 = \mathbf{E}_1[0 : M_2' - 1, 0 : M_2' - 1]$. When Φ_2^* is the first M_2' columns of Φ_1^*, we have $\Pr\{\text{rk}(\Phi_1^*\mathbf{E}_1) \geq \text{rk}(\Phi_2^*\mathbf{E}_2)\} = 1$, and hence the third condition is satisfied.

From the above analysis, we obtain a general approach to design a proper recoding scheme: First, design a reduced recoding transfer matrix Φ^* with dimension $M \times M'$ for the maximum value of M', and then use the submatrices of Φ^* as the reduced recoding transfer matrices of other dimensions. Since a totally random matrix of a smaller dimension can be regarded as a submatrix of a totally random matrix of a larger dimension (in both row and column), the RLNC recoding discussed in the last chapter is proper. We will introduce another class of proper recoding schemes later in this chapter.

4.1.2 PROPER TRANSITION MATRIX

The batches received at a network node u may have different ranks. Suppose the (empirical) rank distribution of the batches received at a network node is π, which is a vector with $M + 1$ entries.[1] Now we study how a proper recoding scheme affects the rank distribution of the batches received at u^+, the node at the next hop. Assume that M' recoded packets are generated for all the received batches (of different ranks), and let the packet loss rate on (u, u^+) be ϵ.

For a batch of rank r received at node u, its rank is s at node u^+ with probability $\Pr\{\text{rk}(\Phi^*\mathbf{E}) = s\}$, where Φ^* is the $r \times M'$ reduced recoding generator matrix and $\mathbf{E}$ is an

[1]Note that we do not assume that the ranks of all the batches are independent.

$M' \times M'$ random *diagonal* matrix $\mathbf{E}$ with independent components, where a diagonal component is 0 with probability ϵ and is 1 with probability $1-\epsilon$. The transition of the rank distributions between nodes u and u^+ can be given by an $(M+1)\times(M+1)$ probability transition matrix $\mathbf{P}$, called a rank transition matrix, so that the rank distribution at node u^+ is $\pi\mathbf{P}$.

For the RLNC recoding with $M' = M$, the rank transition matrix is characterized in Lemma 3.1. For a general value of M', Lemma 3.1 can be generalized as follows. Denote by $\mathbf{P}_{\mathrm{R}}$ the rank transition matrix between nodes u and u^+ for the RLNC recoding, where M' recoded packets are generated and the packet loss rate on (u, u^+) is ϵ.

Lemma 4.2 *For* $0 \le j \le i \le M$, $\mathbf{P}_{\mathrm{R}}[i,j] = \sum_{k=j}^{M'} \binom{M'}{k}(1-\epsilon)^k \epsilon^{M'-k} \zeta_j^{i,k}(q_m)$.

For two rank distributions π and π', we say that π *dominates* π, denoted by $\pi \succcurlyeq \pi'$, if $\sum_{r=j}^{M} \pi[r] \ge \sum_{r=j}^{M} \pi'[r]$ for all $j = 1, \ldots, M$. If $\pi_1 \succcurlyeq \pi_1'$ and $\pi_2 \succcurlyeq \pi_2'$, then $\alpha\pi_1 + (1-\alpha)\pi_2 \succcurlyeq \alpha\pi_1' + (1-\alpha)\pi_2'$. The dominance relation gives a partial order on the rank distributions.

Definition 4.3 We say a rank distribution transition matrix $\mathbf{P}$ is *proper* if for any rank distributions π and π' with $\pi \succcurlyeq \pi'$, we have $\pi\mathbf{P} \succcurlyeq \pi'\mathbf{P}$.

Lemma 4.4 *An* $(M+1)\times(M+1)$ *transition matrix* $\mathbf{P}$ *is proper if and only if* $\mathbf{P}[i,:] \succcurlyeq \mathbf{P}[i-1,:]$, *for all* $i = 1, \ldots, M$.

Proof. We use $\mathbf{e}_i$ to denote a row $(M+1)$-vector of the form $(0, \ldots, 0, 1, 0, \ldots, 0)$, with the ith entry being 1. Then $\mathbf{e}_i \succcurlyeq \mathbf{e}_{i-1}$. If $\mathbf{P}$ is proper, we have $\mathbf{P}[i,:] = \mathbf{e}_i\mathbf{P} \succcurlyeq \mathbf{e}_{i-1}\mathbf{P} = \mathbf{P}[i-1,:]$. On the other hand, suppose that we have a rank distribution transition matrix $\mathbf{P}$ with $\mathbf{P}[i:] \succcurlyeq \mathbf{P}[i-1,:]$, for all $i = 1, \ldots, M$. For any rank distributions π and π' with $\pi \succcurlyeq \pi'$, construct π_k, $k = 0, 1, \ldots, M$ as follows.

- $\pi_0 = \pi$.
- For $k = 1, \ldots, M$,

$$\begin{aligned} \pi_k[0 : M-k-1] &= \pi_{k-1}[0 : M-k-1] \\ \pi_k[M-k] &= \pi_{k-1}[M-k] + \pi_{k-1}[M-k+1] - \pi'[M-k+1] \\ \pi_k[M-k+1 : M] &= \pi'[M-k+1 : M]. \end{aligned}$$

It can be checked inductively that $\pi_M = \pi'$ and $\pi_{k-1} \succcurlyeq \pi_k$ for $k = 1, \ldots, M$. By $\pi_{k-1} \succcurlyeq \pi'$ for $k = 1, \ldots, M$, we have

$$\begin{aligned} 0 &\le \sum_{r=M-k+1} \pi_{k-1}[r] - \sum_{r=M-k+1} \pi'[r] \\ &= \pi_{k-1}[M-k+1] + \sum_{r=M-k+2} \pi_{k-1}[r] - \sum_{r=M-k+1} \pi'[r] \\ &= \pi_{k-1}[M-k+1] + \sum_{r=M-k+2} \pi'[r] - \sum_{r=M-k+1} \pi'[r] \\ &= \pi_{k-1}[M-k+1] - \pi'[M-k+1]. \end{aligned} \tag{4.3}$$

We can then verify that $\pi_{k-1}\mathbf{P} \succcurlyeq \pi_k\mathbf{P}$ for $k = 1, \ldots, M$ as follows. Write

$$\begin{aligned} \pi_{k-1}\mathbf{P} &= \sum_{i=0}^{M} \pi_{k-1}[i]\mathbf{P}[i,:] \\ &= \sum_{i=0}^{M-k-1} \pi_{k-1}[i]\mathbf{P}[i,:] + \pi_{k-1}[M-k]\mathbf{P}[M-k,:] \\ &\quad + (\pi_{k-1}[M-k+1] - \pi'[M-k+1])\mathbf{P}[M-k+1,:] + \sum_{i=M-k+1}^{M} \pi'[i]\mathbf{P}[i,:] \\ &\succcurlyeq \sum_{i=0}^{M-k-1} \pi_{k-1}[i]\mathbf{P}[i,:] + \pi_{k-1}[M-k]\mathbf{P}[M-k,:] \\ &\quad + (\pi_{k-1}[M-k+1] - \pi'[M-k+1])\mathbf{P}[M-k,:] + \sum_{i=M-k+1}^{M} \pi'[i]\mathbf{P}[i,:] \\ &\succcurlyeq \sum_{i=0}^{M-k-1} \pi_{k-1}[i]\mathbf{P}[i,:] + \pi_{k-1}[M-k]\mathbf{P}[M-k,:] + \sum_{i=M-k+1}^{M} \pi'[i]\mathbf{P}[i,:] \qquad (4.4) \\ &\succcurlyeq \sum_{i=0}^{M} \pi_k[i]\mathbf{P}[i,:] \qquad (4.5) \\ &= \pi_k\mathbf{P}, \end{aligned}$$

where (4.4) follows from (4.3), and (4.5) follows from $\mathbf{P}[M-k+1,:] \succcurlyeq \mathbf{P}[M-k,:]$. □

Consider a proper recoding scheme with a rank transition matrix $\mathbf{P}$, where the same number of recoded packets is generated for all the batches. By checking the first condition of Definition 4.1, we see that $\mathbf{P}[i,:] \succcurlyeq \mathbf{P}[i-1,:]$ for all $i = 1, \ldots, M$, and hence $\mathbf{P}$ is proper by Lemma 4.4.

4.2 SYSTEMATIC RECODING

Following the notations in the last section, k is the number of received packets belonging to a batch and r is the number of packets with linearly independent coefficient vectors among the k

received packets. Suppose we want to generate and transmit M' recoded packets belonging to the batch to the node at the next hop. When using RLNC as the recoding scheme, the computation cost and the recoding delay are as follows.

- Generating 1 new packet using random linear combinations takes Tk finite-field multiplications. Alternatively, we can select r linearly independent packets among the k received packets, and then apply RLNC on these r packets. It takes $O(r^3)$ finite-field operations for determining the r linearly independent packets, and Tr finite-field multiplications for generating 1 recoded packet. RLNC generates totally M' new recoded packets using random linear combinations.
- A relay node using RLNC can start to transmit the first recoded packets of the batch only after all the k packets have been received. If all the packets of the same batch are transmitted consecutively, then the recoding delay of this batch is at least k time slots.

We introduce a new class of recoding technique called *systematic recoding*, which reduces both the recoding computation cost and the recoding delay.

4.2.1 FIRST SYSTEMATIC RECODING SCHEME (SR-1)

The first systematic recoding scheme is called SR-1, where we choose r linearly independent received packets as the recoded packets, and generate $M' - r$ new recoded packets using random linear combination of the r linearly independent received packets. The computation cost of SR-1 for generating the recoded packets is only $1 - r/M'$ of that of RLNC. In addition, it takes $O(r^3)$ finite-field operations for determining the r linearly independent packets.

Let us compare systematic recoding and RLNC by using the expected rank of the batch at the node at the next hop. Let Φ_{S}^* be the reduced recoding generator matrix of SR-1. Recall that Φ_{R}^* is the reduced recoding generator matrix of RLNC. Φ_{S}^* and Φ_{R}^* are $r \times M'$ matrices with the following specifications.

- Φ_{R}^* is a totally random matrix over $\mathrm{GF}(q_m)$.
- The first r columns of Φ_{S}^* is the identity matrix, and the remaining $M' - r$ columns are totally random over $\mathrm{GF}(q_m)$.

In SR-1, the first r recoded packets (uncoded) are called *symmetric packets*, and the last $M' - r$ recoded packets are called *RLNC packets*. Intuitively,

$$\mathrm{rk}(\Phi_{\mathrm{S}}^*\mathbf{E}) \succcurlyeq \mathrm{rk}(\Phi_{\mathrm{R}}^*\mathbf{E}) \tag{4.6}$$

because the first r columns of Φ_S^* are always linearly independent. One technique to show the above inequality is *coupling*: notice that the probabilities on both sides of the inequality do not change if we make the last $M' - r$ columns of Φ_{S}^* and Φ_{R}^* the same. The above inequality then follows from the fact that the subspace spanned by the first r column of Φ_{R}^* is a subset of that of

Φ_{S}^*. The condition (4.6) further implies that $\mathbb{E}[\mathrm{rk}(\Phi_{\mathrm{S}}^*\mathbf{E})] \geq \mathbb{E}[\mathrm{rk}(\Phi_{\mathrm{R}}^*\mathbf{E})]$, i.e., in general, SR-1 is better than RLNC for recoding.

Moreover, since the first r columns of Φ_{R}^* tends to be full rank as $q_m \to \infty$, we have

$$\lim_{q_m\to\infty} \Pr\{\mathrm{rk}(\Phi_{\mathrm{S}}^*\mathbf{E}) \geq a\} = \lim_{q_m\to\infty} \Pr\{\mathrm{rk}(\Phi_{\mathrm{R}}^*\mathbf{E}) \geq a\}, \text{ for all } a = 0, 1, \ldots, M.$$

In other words, when the field size q_m is large (e.g., 256 when $M = 16$), RLNC and SR-1 have almost the same recoding performance.

Denote by $\mathbf{P}_{\mathrm{S}}$ the rank transition matrix between two consecutive nodes for SR-1, where the packet loss rate between these two nodes is ϵ and M' recoded packets are generated for all batches. We see that $\mathbf{P}_{\mathrm{S}}$ is a lower triangular matrix and the distribution of $\mathrm{rk}(\Phi_{\mathrm{S}}^*\mathbf{E})$ is the r-th row of $\mathbf{P}_{\mathrm{S}}$.

Lemma 4.5 *For* $0 \leq j \leq i \leq M$,

$$\mathbf{P}_{\mathrm{S}}[i,j] = \begin{cases} \sum_{v=j}^{M'} \sum_{a=0\vee(v+i-M')}^{j} \binom{M'-i}{v-a}\binom{i}{a}(1-\epsilon)^v \epsilon^{M'-v} \zeta_{j-a}^{i-a,v-a}(q_m) & i < M', \\ \binom{M'}{j}(1-\epsilon)^j \epsilon^{M'-j} & i \geq M'. \end{cases}$$

Proof. Consider that node u receives a batch of rank i and generates M' recoded packets using SR-1. When $i \geq M'$, all the M' recoded packets are linearly independent, and hence the rank of the batch at node u^+ is equal to the number of received packets.

When $i < M'$, among these M' recoded packets, i of them are systematic packets, and $M' - i$ of them are RLNC packets. Since M' packets are transmitted, the probability that v packets are received at u^+ is $(1-\epsilon)^v\epsilon^{M'-v}$. Among the v received packets, the probability that a of them are systematic packets and $v - a$ of them are RLNC packets is $\binom{M'-i}{v-a}\binom{i}{a}$. The a systematic packets are linearly independent, and the probability that the $v - a$ RLNC packets are linearly independent with the a systematic packets and have rank $j - a$ is $\zeta_{j-a}^{i-a,v-a}(q_m)$. Considering all the possible value of v, we prove the lemma. □

4.2.2 SIMPLIFIED SYSTEMATIC RECODING (SR-2)

The recoding computation complexity and the transmission delay of SR-1 can be further reduced by using all the received packets as the recoded packets. We call this scheme SR-2.

1. When $M' \leq r$, M' linearly independent received packets are transmitted as the recoded packets.

2. When $r < M' \leq k$, the r linearly independent received packets and $M' - r$ other received packets are transmitted as the recoded packets.

3. When $M' > k$, the k received packets are used as the recoded packets, and $M' - k$ new recoded packets are generated using random linear combinations of the k received packets.

In the first case ($M' \leq r$), SR-1 and SR-2 both only forward the packets the relay node has received, and do not generate any new packets. In the second case ($r < M' \leq k$), SR-2 just forwards the packets the relay node has received while SR-1 generates $M' - r$ new packets. In the third case ($M' > k$), SR-2 generates $M' - k$ new packets and SR-1 generates $M' - r$ new packets. In other words, SR-2 generates fewer new packets using random linear combination than SR-1 when $r < k$ and $M' > r$.

Note that in SR-2, it is not necessary to select r linearly independent packets among the k received packets. Thus compared with RLNC and SR-1, $O(r^3)$ finite-field operations are saved.

Now we compare the transmission delay of SR-2 and SR-1 when $M' \geq k$, which usually holds for most practical cases (see the next section). In this case, all the received packets can be transmitted in SR-2 and only r linearly independent received packets can be transmitted in SR-1. Therefore, in SR-2, the network node can transmit a packet right after receiving it without any extra transmission delay, while in SR-1, the network node can transmit at most r packets before receiving all the k packets and hence has at least $k - r$ time slots more transmission delay than SR-2.

Next, we show that when all the parameters are the same, SR-1 and SR-2 result in the same rank distribution at the node at the next hop. Consider that the node at the previous hop u^- transmits M' recoded packets for a batch with rank m. Denote by $\mathbf{H}$ the coefficient matrix of these M' packets, where the first m columns of $\mathbf{H}$ are linearly independent and the remaining columns are independently and uniformly distributed on the subspace spanned by the first m columns. Due to packet loss, node u only observes the columns of $\mathbf{H}$ with indices in a subset $\mathcal{K} \subset \{0, 1, \ldots, M' - 1\}$. For a subset $\mathcal{A} \subset \{0, 1, \ldots, M' - 1\}$, we denote by $\mathbf{H}_{\mathcal{A}}$ the submatrix formed by the columns of $\mathbf{H}$ with indices in $\mathcal{A}$. Suppose $\mathrm{rk}(\mathbf{H}_{\mathcal{K}}) = r$.

We first see that for SR-1, the coefficient matrix $\mathbf{H}'_{\text{SR-1}}$ of the M' recoded packets generated at node u has the property that

(1) r out of the M' columns of $\mathbf{H}'_{\text{SR-1}}$ are linearly independent; and

(2) the other $M' - r$ columns of $\mathbf{H}'_{\text{SR-1}}$ are independently and uniformly distributed on the subspace spanned by the r linearly independent columns.

For SR-2, the coefficient matrix $\mathbf{H}'_{\text{SR-2}}$ of the M' recoded packets generated at node u has $\mathbf{H}_{\mathcal{K}}$ as a submatrix. The part of $\mathbf{H}'_{\text{SR-2}}$ that is not in the submatrix $\mathbf{H}_{\mathcal{K}}$ is generated by random linear combination of the columns in $\mathbf{H}_{\mathcal{K}}$. The following lemma implies that $\mathbf{H}'_{\text{SR-2}}$ satisfies the same property (1) and (2) as $\mathbf{H}'_{\text{SR-1}}$.

Lemma 4.6 *There exists a partition $\{\mathcal{I}, \mathcal{J}\}$ of $\mathcal{K}$ such that $|\mathcal{I}| = r$, $rk(\mathbf{H}_{\mathcal{I}}) = r$, and the columns of $\mathbf{H}_{\mathcal{J}}$ are independent and uniformly distributed on the subspace spanned by the columns of $\mathbf{H}_{\mathcal{I}}$.*

Proof. Since the first m columns of $\mathbf{H}$ are linearly independent, we expand $\mathcal{K} \cap \{0, 1, \ldots, m-1\}$ to form $\mathcal{I}$ such that $|\mathcal{I}| = r$ and $\mathrm{rk}(\mathbf{H}_{\mathcal{I}}) = r$, which is feasible since $\mathrm{rk}(\mathbf{H}_{\mathcal{K}}) = r$.

Since $\mathcal{J} \cap \{0, 1, \ldots, m-1\} = \emptyset$, we know that the columns of $\mathbf{H}$ with indices in $\mathcal{J}$ are independent and uniformly distributed on the subspace spanned by the first m columns of $\mathbf{H}$. Under the condition that $\mathrm{rk}(\mathbf{H}_{\mathcal{I}}) = \mathrm{rk}(\mathbf{H}_{\mathcal{I}\cup\mathcal{J}}) = r$, the columns of $\mathbf{H}_{\mathcal{J}}$ are independent and uniformly distributed on the subspace spanned by the columns of $\mathbf{H}_{\mathcal{I}}$. □

The same property (1) and (2) of $\mathbf{H}'_{\text{SR-1}}$ and $\mathbf{H}'_{\text{SR-2}}$ implies that the recoded packets generated by both schemes induce the same rank distribution at the node u^+. Applying the above analysis on all the relay nodes inductively with the source node as the first node, we see that SR-1 and SR-2 have exactly the same recoding performance in line networks.

4.2.3 COMPARISON OF RLNC AND SYSTEMATIC RECODING

We compare the performance of RLNC and systematic recoding numerically. Here we pick $l = 20$, $\epsilon = 0.2$ and $M = M' = 16$. We calculate the rank distributions at the destination node for RLNC and systematic recoding, respectively, for $q_m = 2$ and 256. Table 4.1 lists the expected rank distribution for both RLNC and systematic recoding. We observe clear advantage of systematic recoding for $q_m = 2$. Both RLNC and systematic recoding have almost the same expected ranks for $q_m = 256$.

Table 4.1: RLNC vs. systematic recoding

(a) q_m = 2										
l	1	2	3	4	5	6	7	8	9	10
RLNC	0.7891	0.7166	0.6745	0.6451	0.6226	0.6045	0.5893	0.5762	0.5647	0.5545
SR	0.8	0.7275	0.6859	0.6570	0.6351	0.6174	0.6027	0.5901	0.5789	0.5691
(b) q_m = 256										
l	1	2	3	4	5	6	7	8	9	10
RLNC	0.8000	0.7445	0.7149	0.6951	0.6805	0.6689	0.6594	0.6514	0.6444	0.6383
SR	0.8	0.7445	0.7149	0.6951	0.6805	0.6689	0.6594	0.6514	0.6444	0.6383

4.3 NUMBER OF RECODED PACKETS

Consider a line network of length l ($l \geq 1$), as shown in Figure 3.6, and use R_0 and R_l to denote the source node and the destination node, respectively. We assume that the links in the line network transmit one packet per use, but can have different packet loss rates. Denote by ϵ_k the loss rate on link $(\mathrm{R}_{k-1}, \mathrm{R}_k)$.

Let $M_k, k = 0, 1, \ldots, l-1$ be the number of recoded packets to be generated for a batch at node R_k, and let $\pi_k, k = 0, 1, \ldots, l$ be the rank distribution of a batch at node R_k. A network node may use different methods to generate the recoded packets, e.g., RLNC or systematic recoding. Since a network node uses only its received packets for recoding, the rank of a batch at node $\mathrm{R}_i, i = 0, 1, \ldots, l$ form a Markov chain. That is, for $k > 0$,

$$\pi_k = \pi_{k-1}\mathbf{P}_k,$$

where $\mathbf{P}_k$ is the rank transition matrix from node R_{k-1} to R_k.

The matrix $\mathbf{P}_k$ is determined by the recoding method at node R_{k-1}. When the network node applies RLNC, according to Lemma 4.2, for $0 \le j \le i \le M$,

$$\mathbf{P}_k[i,j] = \sum_{r=j}^{M_{k-1}} \binom{M_{k-1}}{r} (1-\epsilon_k)^r \epsilon_k^{M_{k-1}-r} \zeta_j^{i,r}(q_m),$$

where M_{k-1} is a fixed parameter and ϵ_k can be measured at node R_k. When the network node applies systematic recoding, according to Lemma 4.5, for $0 < j \le i < M_{k-1}$,

$$\mathbf{P}_k[i,j] = \sum_{r=j}^{M_{k-1}} \sum_{a=0\vee(r+i-M_{k-1})}^{j} \binom{M_{k-1}-i}{r-a}\binom{i}{a} (1-\epsilon)^r \epsilon_k^{M_{k-1}-r} \zeta_{j-a}^{i-a,r-a}(q_m),$$

and for $i \ge M_{k-1}$ and $0 < j \le i \le M$, $\mathbf{P}_k[i,j] = \binom{M_{k-1}}{j}(1-\epsilon)^j \epsilon_k^{M_{k-1}-j}$.

4.3.1 GLOBAL OPTIMIZATION

As shown in Table 3.1, using $M_k = M$ for all k in BATS-Pro-0 does not give a good performance when the packet loss rates are high (e.g., 0.9) and M is relatively small (e.g., 16). We now study how to choose $M_k, k = 0, 1, \ldots, l-1$ such that a relatively small batch size can also be applied for used packet loss rate.

Our objective is to maximize the expected rank per network use at the destination node, where the network use $\tilde{M}$ is the maximum number of times that a network link is used for transmitting a batch. In other words, we want to maximize $\mathbb{E}[\pi_l]/\tilde{M}$, such that $M_k \le \tilde{M}$ for all k.

Theorem 4.7 *For a proper recoding scheme, $\mathbb{E}[\pi_l]/\tilde{M}$ with $M_k \le \tilde{M}$ for all k is maximized when $M_k = \tilde{M}, k = 0, 1, \ldots, l-1$.*

Proof. Fix an integer k with $0 \le k \le l-1$ and $M_k = M' < \tilde{M}$. We will show that if we increase M_k by one, which preserves the feasibility, $\mathbb{E}[\pi_l]$ would not be decreased. Repeating the above procedure proves the lemma.

Denote by $\mathbf{P}'_k$ and π'_k the corresponding rank transition matrix and rank distribution when $M_k = M' + 1$. Since only M_k is increased, $\pi_k = \pi'_k$ and $\mathbf{P}_j = \mathbf{P}'_j$, $j = k+2, \ldots, l$. The definition of proper recoding implies that $\mathbf{P}'_{k+1}[i,:] \succcurlyeq \mathbf{P}_{k+1}[i,:]$ for $i = 0, 1, \ldots, M$. Therefore, $\pi'_{k+1} \succcurlyeq \pi_{k+1}$. By Lemma 4.4, we know that $\mathbf{P}_j$, $j = k+2, \ldots, l$ are all proper. Hence, using the properties of a proper rank transition matrix, we have $\pi'_l = \pi'_{k+1}\mathbf{P}'_{k+2}\ldots\mathbf{P}'_l = \pi'_{k+1}\mathbf{P}_{k+2}\ldots\mathbf{P}_l \succcurlyeq \pi_{k+1}\mathbf{P}_{k+2}\ldots\mathbf{P}_l = \pi_l$. □

This lemma says that we only need to let $M_k = \tilde{M}, k = 0, 1, \ldots, l-1$, and then maximize $\mathbb{E}[\pi_l]/\tilde{M}$. In other words, we can find the optimal values of M_k, $k = 1, \ldots, l$ by solving the following optimization problem

$$\begin{aligned} &\max_{\tilde{M}} \mathbb{E}[\pi_l]/\tilde{M} \\ &\text{s.t. } M_k = \tilde{M}, k = 0, 1, \ldots, l-1. \end{aligned} \tag{4.7}$$

Denote by $\bar{R}^*$ the optimal value of (4.7). In general, we know that $\bar{R}^* \leq 1 - \epsilon$. When $l = 1$, the upper bound can be achieved using $\tilde{M} = M$ and systematic recoding.

When $l > 1$, $\bar{R}^*$ can be solved numerically. For each given value of $\tilde{M}$, we can numerically evaluate $\mathbb{E}[\pi_l]/\tilde{M}$ using the formula of $\mathbf{P}_k$ with $M_k = \tilde{M}$. Note that we do not need to consider a very large value of $\tilde{M}$. Suppose we have a design objective of $\bar{R}^* \geq (1-\epsilon)/2$. Since $\mathbb{E}[\pi_l] \leq M$, we have $\mathbb{E}[\pi_l]/\tilde{M} < M/\tilde{M}$. If $M/\tilde{M} \leq (1-\epsilon)/2$, then $\mathbb{E}[\pi_l]/\tilde{M} < (1-\epsilon)/2$, and hence the design objective cannot be satisfied. So, we need $M/\tilde{M} > (1-\epsilon)/2$, or $\tilde{M} < 2M/(1-\epsilon)$. If we cannot find $\mathbb{E}[\pi_l]/\tilde{M} \geq (1-\epsilon)/2$ for all $\tilde{M} < 2M/(1-\epsilon)$, we need to increase the batch size M and run the above optimization process again.

We discuss a special case of the maximization problem when $M = 1$. In this case, the repetition recoding scheme is applied, where if at least one packet is received for a batch, the received packet is transmitted $\tilde{M}$ times. The expected rank of a batch in this case is equal to the probability that a batch has rank 1 at the destination node, so that

$$\frac{\mathbb{E}[\pi_l]}{\tilde{M}} = \frac{(1-\epsilon^{\tilde{M}})^l}{\tilde{M}}, \tag{4.8}$$

which can be optimized explicitly.

Theorem 4.8 *When $M = 1$, $l > 1$ and the repetition recoding scheme is applied, the optimization (4.7) is maximized when $\tilde{M}$ is an integer around $\frac{t^*(l)}{\ln(1/\epsilon)}$, where $\ln l < t^*(l) < 2\ln l$ is the solution of $e^t - 1 - lt = 0, t > 0$, and hence the optimal value $\bar{R}^*$ of (4.7) is $\Theta\left(\frac{\ln(1/\epsilon)}{\ln l}\right)$.*

Proof. To maximize (4.8), we relax $\tilde{M}$ to a real number and solve $\frac{\mathrm{d}\,\mathbb{E}[\pi_l]/\tilde{M}}{\mathrm{d}\,\tilde{M}} = 0$, i.e.,

$$1 - \epsilon^{\tilde{M}} + l\tilde{M}\epsilon^{\tilde{M}}\ln\epsilon = 0, \tag{4.9}$$

or

$$\epsilon^{-\tilde{M}} - 1 + l\tilde{M}\ln\epsilon = 0.$$

Let $t = -\tilde{M}\ln\epsilon$, and denote by $t^*(l)$ the solution of $g(t) \triangleq e^t - 1 - lt = 0, t > 0$. Then the solution of (4.9) is $\tilde{M}^* = t^*(l)/\ln(1/\epsilon)$.

We know that $g(t) < 0$ for $0 < t < t^*(l)$; and $g(t) > 0$ for $t > t^*(l)$. Since $g(\ln l) = l - 1 - l\ln l < 0$ and $g(2\ln l) = l^2 - 1 - 2l\ln l > 0$ when $l > 1$, we have $\ln l < t^*(l) < 2\ln l$ when $l > 1$. Last, using $\epsilon^{\tilde{M}^*} = e^{-t^*(l)}$,

$$0.25 \le (1 - 1/l)^l \le (1 - \epsilon^{\tilde{M}^*})^l \le \left(1 - 1/l^2\right)^l < 1,$$

and hence $\bar{R}^* = (1 - \epsilon^{\tilde{M}^*})^l/\tilde{M}^* = \ln(1/\epsilon)(1 - \epsilon^{\tilde{M}^*})^l/t^*(l) = \Theta(\ln(1/\epsilon)/\ln l)$. □

4.3.2 NUMERICAL EVALUATIONS

In this subsection, we maximize $\mathbb{E}[\pi_l]/\tilde{M}$ for various cases to investigate the performance of systematic recoding in the line network shown in Figure 3.6. We assume that $\epsilon_k = \epsilon$, $k = 1, \ldots, l$. Denote by $\tilde{M}^*$ the value of $\tilde{M}$ maximizing (4.7).

$\tilde{M}^*$ for different network lengths

We evaluate $\tilde{M}^*$ for different network length l. The numerical results are given in Fig. 4.1, where (a) is the zoom-in of (b) for $l = 1, 2, \ldots, 50$. For fixed M and q_m, we see that $\tilde{M}^*$ increases slowly with the network length, and takes constant values piecewisely. Similar to the special case illustrated in Theorem 4.8, $\tilde{M}^*$ is roughly in proportion to $\ln l$. For fixed l and q_m, we see that $\tilde{M}^*$ increases roughly linearly with M.

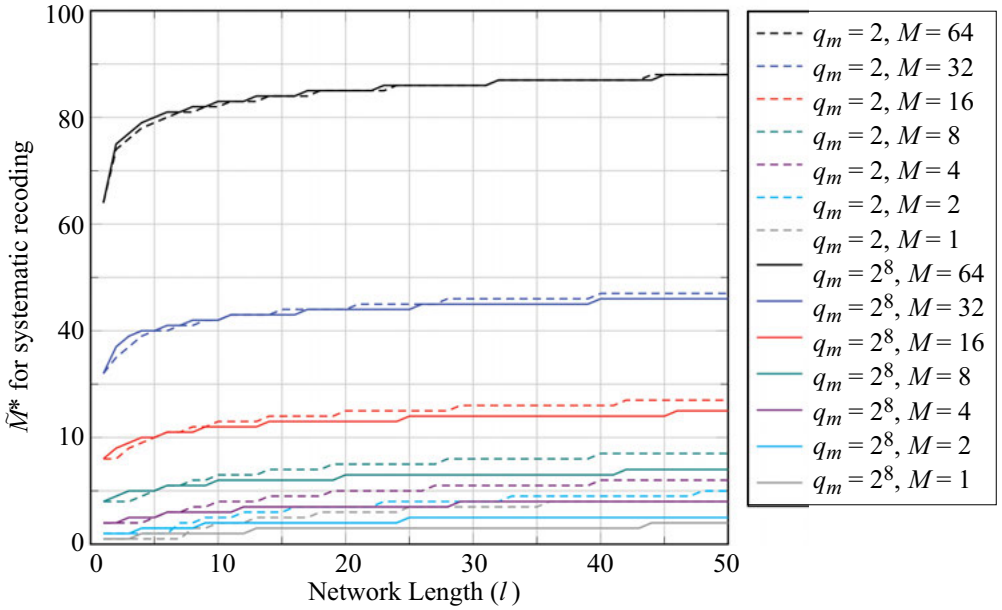
$\bar{R}^*$ for different network lengths

We first evaluate $\bar{R}^*$ with respective to different network length l. The value of $\bar{R}^*$ is an upper bound on the achievable rates of BATS codes. The numerical results are given in Figure 4.2, where Figure 4.2a is the zoom-in of Figure 4.2b for l in the range 1–50.

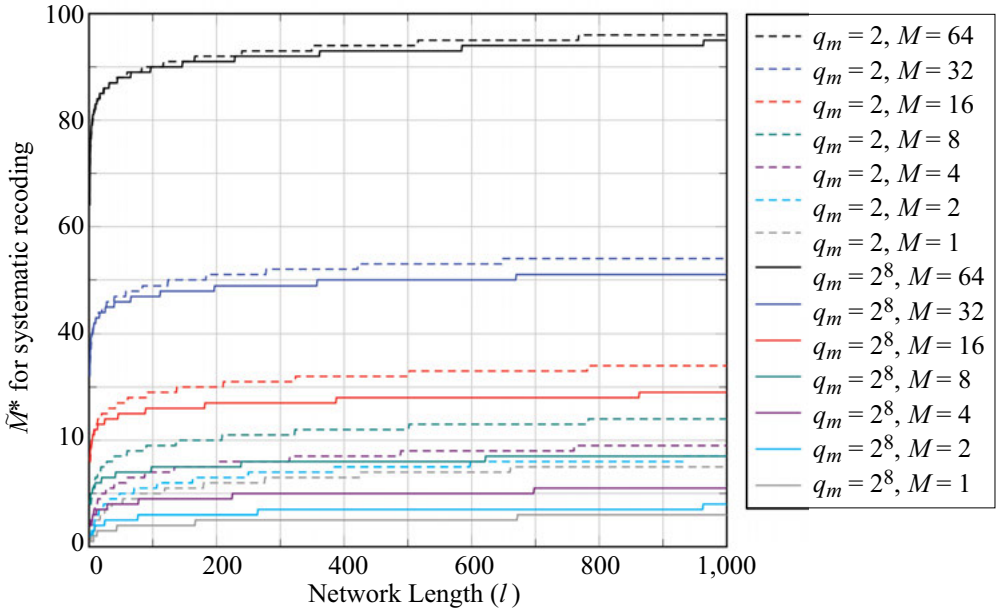
Compared with the performance of BATS-Pro-0 in Figure 3.7, we see that the advanced recoding techniques introduced in this chapter significantly improve the normalized expected rank at the destination node, especially for small q_m and M. For example, when $q_m = 2$, $M = 8$, and $l = 50$, $\bar{R}^*$ is more than 200% better than the performance of BATS-Pro-0.

Similar to the special case illustrated in Theorem 4.8, $\bar{R}^*$ decreases slowly with l, roughly in proportion to $1/\ln l$. For $l = 1{,}000$, $\bar{R}^*$ is larger than 0.4 (half of the network capacity) for many choices of q_m and M (e.g., $q_m = 2$ and $M = 16$). These results show that BATS codes can be used for very long networks.

Now let us explain why the curve for $M = 1$ and $q = 2^8$ is approximately piecewise linear. We have observed that $\tilde{M}^*$ takes constant values piecewisely. Suppose that $\tilde{M}^* = c$ for $l \in [l_0, l_1]$. Since $q = 2^8$ is large, we can approximate $\bar{R}^*$ in this case by (4.8) with c in place of $\tilde{M}$, i.e., $\bar{R}^* \approx (1 - \epsilon^c)^l/c \approx (1 - l\epsilon^c)/c$, where the latter is a linear function for $l \in [l_0, l_1]$.

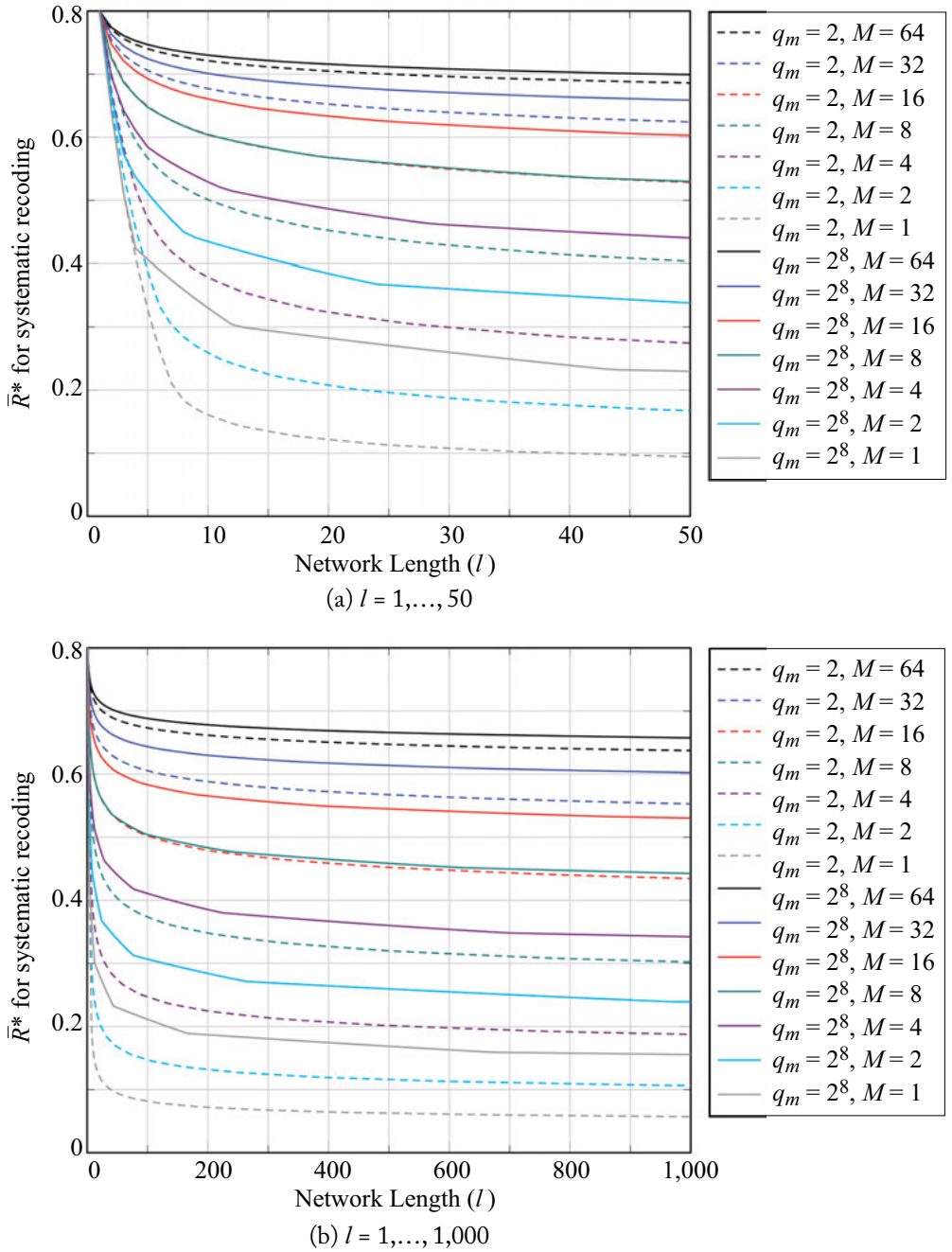


(a) $l = 1, \ldots, 50$



(b) $l = 1, \ldots, 1{,}000$

Figure 4.1: The optimizer of (4.7) with systematic recoding for line networks with $\epsilon = 0.2$.



(a) $l = 1, \ldots, 50$

(b) $l = 1, \ldots, 1{,}000$

Figure 4.2: The optimal value of (4.7) with systematic recoding for line networks with $\epsilon = 0.2$.

$\bar{R}^*$ and $\tilde{M}^*$ for different values of ϵ

We evaluate $\bar{R}^*$ and $\tilde{M}^*$ for different values of ϵ. Here we use $l = 20, 80$, $M = 16, 32$ and $q_m = 2, 256$. For each combination of these parameters, we calculate $\bar{R}^*$ and $\tilde{M}^*$ for $\epsilon = 0.1, 0.2, \ldots, 0.9$ in Table 4.2.

The first column in both tables give the value of $1 - \epsilon$, which is the network capacity without any network storage and computation constraints, and hence an upper bound on $\bar{R}^*$. When $l = 20$, $q_m = 2$, and $M = 32$, $\bar{R}^*$ is at least $0.7(1 - \epsilon)$ for $0.1 \leq \epsilon \leq 0.9$. These results show that BATS codes with the binary field can be used for a wide range of packet loss rate.

Table 4.2: $\bar{R}^*$ and $\tilde{M}^*$ for different values of ϵ

(a) $l = 20$, $M = 16, 32$								
	$q_m = 2$, $M = 16$		$q_m = 2$, $M = 32$		$q_m = 2^8$, $M = 16$		$q_m = 2^8$, $M = 32$	
$1 - \epsilon$	$\bar{R}^*$	$\tilde{M}^*$	$\bar{R}^*$	$\tilde{M}^*$	$\bar{R}^*$	$\tilde{M}^*$	$\bar{R}^*$	$\tilde{M}^*$
0.9	0.6728	21	0.7589	38	0.7576	20	0.8005	38
0.8	0.568	25	0.6524	44	0.6335	23	0.6813	44
0.7	0.4802	29	0.556	52	0.529	27	0.5767	52
0.6	0.3998	35	0.4656	62	0.4367	33	0.4809	61
0.5	0.3246	43	0.38	76	0.3519	41	0.3913	75
0.4	0.2535	55	0.2982	96	0.2733	52	0.3064	95
0.3	0.1859	75	0.2197	130	0.1995	71	0.2253	129
0.2	0.1214	114	0.144	198	0.1298	109	0.1475	197
0.1	0.0595	232	0.0709	402	0.0634	222	0.0725	399
(b) $l = 80$, $M = 16, 32$								
	$q_m = 2$, $M = 16$		$q_m = 2$, $M = 32$		$q_m = 2^8$, $M = 16$		$q_m = 2^8$, $M = 32$	
$1 - \epsilon$	$\bar{R}^*$	$\tilde{M}^*$	$\bar{R}^*$	$\tilde{M}^*$	$\bar{R}^*$	$\tilde{M}^*$	$\bar{R}^*$	$\tilde{M}^*$
0.9	0.6047	24	0.7138	41	0.7152	21	0.77	40
0.8	0.5105	28	0.6113	48	0.5893	25	0.6483	47
0.7	0.4301	33	0.5187	57	0.4877	30	0.5446	55
0.6	0.3571	40	0.4326	68	0.3996	37	0.4514	67
0.5	0.289	50	0.3516	84	0.3201	46	0.3654	82
0.4	0.2251	64	0.275	107	0.2473	59	0.2848	105
0.3	0.1646	87	0.2019	145	0.1797	81	0.2086	143
0.2	0.1072	134	0.132	222	0.1164	125	0.1361	218
0.1	0.0524	273	0.0648	452	0.0567	257	0.0667	444

4.3.3 OPTIMIZATION USING LOCAL INFORMATION

The optimization (4.7) maximizes the normalized expected rank at the destination node, but it requires the knowledge of the packet loss rate of all the network links. So the optimization (4.7) is also called the *global optimization*. In a large distributed wireless networks, e.g., a line network with more than 10 hops, it is difficult to implement this optimization exactly due to the collection of all the packet loss rates and the distribution of the optimization result.

For a network node, we call the status information of its adjacent network nodes and links the *local information*. In other words, only single-hop transmissions are needed for the network node to obtain the local information. Here, we introduce an approach that uses only the local information to optimize the number of batches for recoding, which is called the *local optimization*. Specifically, we assume node R_k knows

- the rank distribution π_k which can be obtained from the batches received at R_k,
- M_{k-1}, the number of recoded packets of a batch transmitted at node R_{k-1}, and
- ϵ_{k+1}, the packet loss rate on link $(\mathrm{R}_k, \mathrm{R}_{k+1})$, which can be known from node R_{k+1}.

We use the convention that $M_{-1} = 0$. The recoding parameters (e.g., l, q_m, and M) are known by all the nodes.

For $0 \leq k < l$, node R_k first solves the following optimization

$$\max_{M_k} \frac{\mathbb{E}[\pi_k (\mathbf{P}_{k+1})^{l-k}]}{M_k}, \tag{4.10}$$

where the nominator of the objective function is the expected rank at the destination node under the assumption that the packet loss rates on all link $(\mathrm{R}_j, \mathrm{R}_{j+1})$, $j = k, k+1, \ldots, l-1$ are ϵ_{k+1}. Node R_k can solve the above optimization using only the local information.

Let M_k^* be the optimal value of (4.10). Instead of using M_k^* as the value of M_k, we set M_k as $\max\{M_k^*, M_{k-1}\}$, which does not change the number of network uses $\max_{k=0}^{l} M_k$, but can potentially increase the expected rank at the destination node.

We use an experiment to compare the performance of the local and global optimizations. For a line network of l hops, we uniformly at random pick ϵ_k, $k = 1, \ldots, l$ in the range $[0, 0.4]$ independently. For these randomly generated packet loss rates, we optimize the number of recoded packets using both the global and the local optimizations.

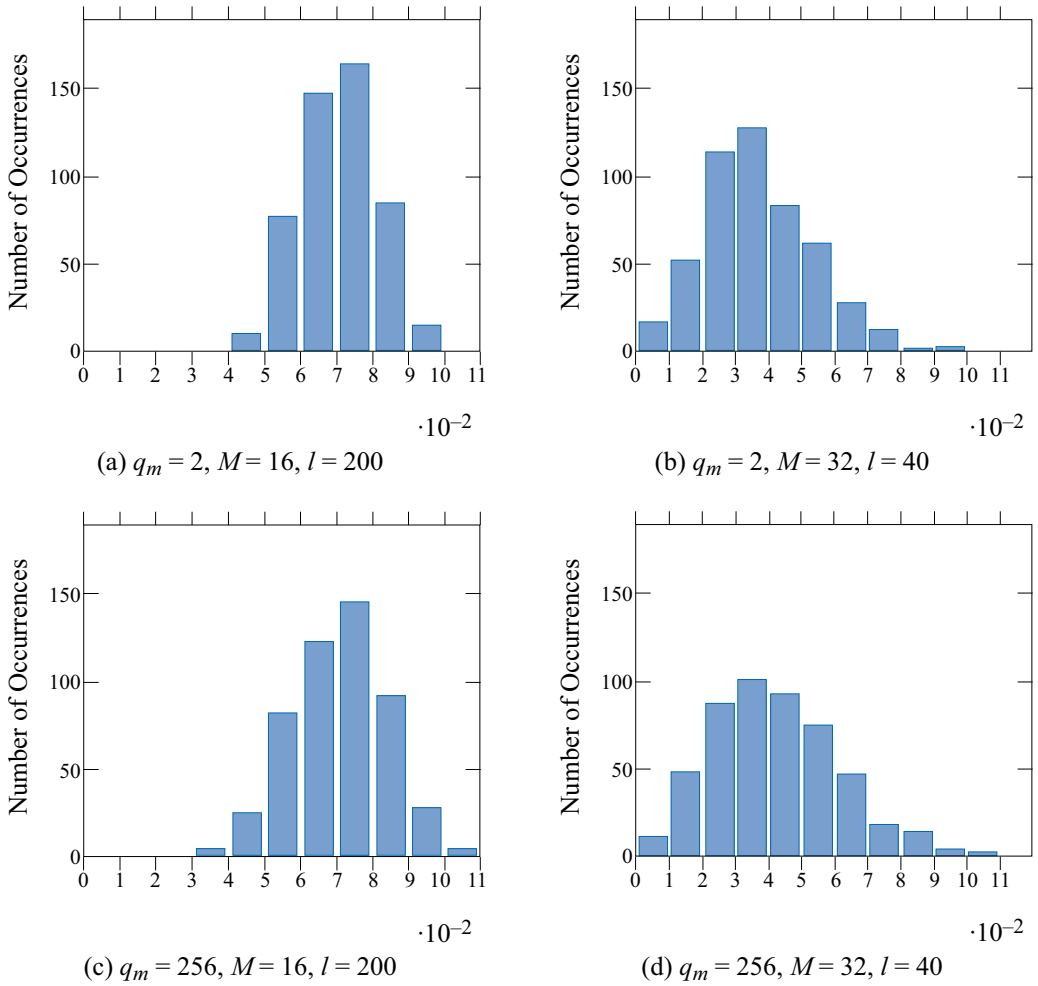
- For the global optimization, we calculate the optimal value $\bar{R}^*$ of (4.7), which is the normalized expected rank at the destination node.
- For the local optimization, we calculate the normalized expected rank at the destination node $\bar{R}^{\text{local}}$ using the following three steps:

  1. Obtain the optimizer M_k^* of (4.10) for $k = 0, 1, \ldots, l-1$.

2. Set $M_0 = M_0^*$, $M_k = \max\{M_k^*, M_{k-1}\}$ for $k = 1, 2, \ldots, l-1$.

3. Calculate the corresponding normalized expected rank at the sink $\bar{R}^{\text{local}}$.

For each set of q_m, M and l values, we repeat the above experiment 500 times and draw the histogram of $1 - (\bar{R}^{\text{local}}/\bar{R}^*)$ in Figure 4.3, where we see that differences between $\bar{R}^{\text{local}}$ and $\bar{R}^*$ are less than 10% for almost all experiments.



(a) $q_m = 2$, $M = 16$, $l = 200$

(b) $q_m = 2$, $M = 32$, $l = 40$

(c) $q_m = 256$, $M = 16$, $l = 200$

(d) $q_m = 256$, $M = 32$, $l = 40$

Figure 4.3: Histograms of $1 - \bar{R}^{\text{local}}/\bar{R}^*$ for 500 instances of the network.

4.4 ADAPTIVE RECODING

In the previous discussion of recoding, the same number of recoded packets is transmitted for all the batches, no matter how many packets are received for a batch. Intuitively, we should transmit more packets for a batch with a higher rank because compared with a batch with a lower rank, the former contains more useful information for decoding. This idea has been justified and implemented in [80, 105], where the technique is called *adaptive recoding*. From the numerical results in [80], we see that adaptive recoding can improve the performance significantly when the batch size is relative small (e.g., $M \leq 16$).

The batches received at a network node u may have different ranks. An adaptive recoding scheme transmits $M'(r)$ recoded packets for a batch when its rank is r. Suppose the (empirical) rank distribution of the batches received at node u is π. The average number of packet transmitted at node u is

$$\bar{M}' = \sum_{r=0}^{M} M'(r)\pi[r],$$

and the rank distribution of the batches received at node u^+ is $\pi\mathbf{P}$, where $\mathbf{P}$ is the rank transition matrix for the adaptive recoding.

The rank transition matrices for adaptive recoding can be characterized similar to these in Lemma 4.2 and Lemma 4.5. Denote by $\mathbf{P}_{\mathrm{RA}}$ and $\mathbf{P}_{\mathrm{SA}}$ the rank transition matrices from node u to node u^+ for RLNC and systematic recoding, respectively, where the recoding scheme transmits $M'(r)$ recoded packets for a batch when its rank is r.

Lemma 4.9 *For* $0 \leq j \leq i \leq M$, $\mathbf{P}_{\mathrm{RA}}[i,j] = \sum_{k=j}^{M'(i)} \binom{M'(i)}{k}(1-\epsilon)^k \epsilon^{M'(i)-k} \zeta_j^{i,k}(q_m)$.

Lemma 4.10 *For* $0 \leq j < i \leq M$,

$$\mathbf{P}_{\mathrm{SA}}[i,j] = \begin{cases} \sum_{r=j}^{M'(i)} \sum_{a=0\vee(r+i-M'(i))}^{j} \binom{M'(i)-i}{r-a}\binom{i}{a}(1-\epsilon)^r \epsilon^{M'(i)-r} \zeta_{j-a}^{i-a,r-a}(q_m) & i < M'(i), \\ \binom{M'(i)}{j}(1-\epsilon)^j \epsilon^{M'(i)-j} & i \geq M'(i). \end{cases}$$

Suppose that node R_k in a line network of length l applies the function $M_k(r)$ to determine the number of recoding packets. We can use the following global optimization approach similar to (4.7) to design $M'(r)$:

$$\begin{aligned} \max_{\tilde{M}, M_k(r), r=0,1,\ldots,M, k=0,1,\ldots,l-1} \quad & \mathbb{E}[\pi_l]/\tilde{M} \\ \text{s.t.} \quad & \sum_{r=0}^{M} M_k(r)\pi_k[r] \leq \tilde{M}, k = 0,1,\ldots,l-1. \end{aligned} \tag{4.11}$$

One approach to solve the above optimization has the following steps.

1. Optimize $\tilde{M}$ by solving (4.7).

2. Fix the optimized $\tilde{M}$ in the last step, and for each k in $\{0, 1, \ldots, l-1\}$, solve the following optimization sequentially

$$\max_{M_k(r), r=0,1,\ldots,M} \frac{\pi_k (\mathbf{P}_{k+1})^{l-k+1}}{\tilde{M}} \quad \text{s.t.} \sum_{r=0}^{M} M_k(r)\pi_k[r] \leq \tilde{M}, \tag{4.12}$$

   where $\mathbf{P}_k$ is the rank transition matrix from node R_{k-1} to R_k using adaptive recoding and $\pi_k = \pi_0 \mathbf{P}_1 \cdots \mathbf{P}_k$ is determined by the optimizer $M_i^*(\cdot)$, $i = 0, 1, \ldots, k-1$ of the first k optimization. We refer readers to [80, 105] for the numerical approaches to solve the above optimization efficiently.

It is also possible to modify the above approach to use only the local information by replacing $\tilde{M}$ in (4.12) with $M_k = \max\{M_k^*, M_{k-1}\}$, where M_k^* is the optimizer of (4.10).

CHAPTER 5

Asymptotic Analysis of BP Decoding

We have discussed a basic BATS protocol and the design of the inner codes in the last two chapters. Starting from this chapter, we move on to the design of the outer codes, which are matrix generalization of fountain codes. In this chapter, we focus on the asymptotic performance of the outer codes when the number of input packets tends to infinity. We obtain a sufficient condition for the BP decoder to recover a given fraction of the input packets, which is called the *intermediate performance* in the literature of fountain codes [28, 50, 65].

5.1 MAIN RESULT

We first present the main analytical result for a generalized BATS code model. Consider a BATS code with K input symbols and n batches of size M, where the i-th batch has degree dg_i, batch generator matrix $\mathbf{G}_i$ and batch transfer matrix $\mathbf{H}_i$. The dg_i input packets involved in the i-th batch, with the index set A_i, are uniformly distributed among all the input packets. The batch generator matrix $\mathbf{G}_i$ is a $\mathrm{dg}_i \times M$ totally random matrix over the base field $\mathbb{F}_q$. The batch transfer matrix has M rows.

Let D be the maximum degree among all batches. For integers d and r with $1 \le d \le D$ and $0 \le r \le M$, let

$$\pi_{d,r} = \frac{|\{i \in \{1, 2, \ldots, n\} : \mathrm{dg}_i = d, \mathrm{rk}(\mathbf{H}_i) = r\}|}{n},$$

the empirical distribution of the batch degree and the transfer matrix rank. We assume that the empirical distribution converges in probability to a probability distribution $\Pi = (\Pi_{d,r}, 1 \le d \le D, 0 \le r \le M)$. Specifically, there exists a function $\lambda(n)$ such that for all sufficiently large n, with probability at least $1 - \lambda(n)$,

$$|\pi_{d,r} - \Pi_{d,r}| = \mathrm{O}(n^{-1/6}), 1 \le d \le D, 0 \le r \le M, \tag{5.1}$$

where $\lim_{n\to\infty} \lambda(n) = 0$.

Compared with BATS codes described in Chapter 2, some assumptions are relaxed here: (i) the batch degrees may not be independently chosen according to a degree distribution; and (ii) the degrees and the transfer matrix ranks can be correlated. For the BATS codes described

in Chapter 2, i.e., all the degrees are independently chosen according to a degree distribution $(\Psi_1, \cdots, \Psi_D)$ and all the batch transfer matrix ranks are i.i.d. following the distribution $(h_0, h_1, \ldots, h_M)$, $\pi_{d,r}$ converges in probability to $\Pi_{d,r} = \Psi_d h_r$. Specifically, the following can be proved by applying Hoeffding's inequality and the union bound:

$$\Pr\left\{|\pi_{d,r} - \Psi_d h_r| < n^{-1/6}, 1 \le d \le D, 0 \le r \le M\right\} \ge 1 - 2MD\exp(-2n^{2/3}).$$

The BP decoding algorithm of BATS codes BP(n) provided in Section 2.2 applies to the generalized version as well. For BP(n), we are interested in the time when the decoding stops, which is equal to the number of input packets that are decoded. For example, if BP(n) stops at time zero, then no input packets are decoded; if BP(n) stops at time K, then all the input packets are decoded. In this chapter, we characterize the ratio of the decoded input packets when K tends to infinity.

For $x \in [0, 1]$, define

$$\Omega(x; \Pi) = \sum_{d=1}^{D}\sum_{r=1}^{M} d\ \Pi_{d,r} \sum_{j=(d-r)^+}^{d-1} \binom{d-1}{j} x^j (1-x)^{d-1-j} \zeta_{d-j}^r, \tag{5.2}$$

where $(d-r)^+ = d - r$ if $d - r \ge 0$ and vanishes otherwise, and ζ_{d-j}^r is defined in (2.3) as

$$\zeta_r^m = \begin{cases} (1-q^{-m})(1-q^{-m+1})\cdots(1-q^{-m+r-1}) & r > 0, \\ 1 & r = 0. \end{cases} \tag{5.3}$$

Lemma 5.1 *We can rewrite*

$$\Omega(x; \Pi) = \sum_{d=1}^{D} d \sum_{r=1}^{M} \mathrm{I}_{d-r,r}(x) \sum_{k=r}^{M} \frac{\zeta_r^k}{q^{k-r}} \Pi_{d,k} \tag{5.4}$$

$$= \sum_{r=1}^{M}\sum_{d=r+1}^{D} d\ \mathrm{I}_{d-r,r}(x) \sum_{k=r}^{M} \frac{\zeta_r^k}{q^{k-r}} \Pi_{d,k} + \sum_{r=1}^{M}\sum_{d=1}^{r} d \sum_{k=r}^{M} \frac{\zeta_r^k}{q^{k-r}} \Pi_{d,k}, \tag{5.5}$$

where

$$\mathrm{I}_{d-r,r}(x) = \sum_{j=(d-r)^+}^{d-1} \binom{d-1}{j} x^j (1-x)^{d-1-j} \tag{5.6}$$

which is called the ***regularized incomplete beta function****.*

Proof. First, we have

$$\sum_{d=1}^{D} d \sum_{r=1}^{M} \mathrm{I}_{d-r,r}(x) \sum_{k=r}^{M} \frac{\zeta_r^k}{q^{k-r}} \Pi_{d,k} = \sum_{d=1}^{D}\sum_{k=1}^{M} d\,\Pi_{d,k} \sum_{r=1}^{k} \mathrm{I}_{d-r,r}(x) \frac{\zeta_r^k}{q^{k-r}}. \tag{5.7}$$

Then, we can write

$$\begin{aligned}\sum_{r=1}^{k} \mathrm{I}_{d-r,r}(x) \frac{\zeta_r^k}{q^{k-r}} &= \sum_{r=1}^{k} \sum_{j=(d-r)^+}^{d-1} \binom{d-1}{j} x^j (1-x)^{d-1-j} \frac{\zeta_r^k}{q^{k-r}} \\ &= \sum_{j=(d-1)^+}^{d-1} \binom{d-1}{j} x^j (1-x)^{d-1-j} \sum_{r=d-j}^{k} \frac{\zeta_r^k}{q^{k-r}} \\ &= \sum_{j=(d-1)^+}^{d-1} \binom{d-1}{j} x^j (1-x)^{d-1-j} \zeta_{d-j}^k, \end{aligned} \tag{5.8}$$

where the last equality is obtained by $\sum_{r=s}^{k} \frac{\zeta_r^k}{q^{k-r}} = \zeta_s^k$. Substituting (5.8) into (5.7), we obtain (5.4). Lastly, (5.5) is obained by noting that when $d \leq r$, $\mathrm{I}_{d-r,r}(x) = 1$. □

Theorem 5.2 *Fix integers $D > 0$, $M > 0$, real numbers $\theta > 0$ and $0 < \eta < 1$. Consider a sequence of BATS codes with K input packets and $n = \lceil K/\theta \rceil$ batches of batch size M, where the empirical distribution of the batch degree and the transfer matrix rank of all batches converges to $\Pi = (\Pi_{d,r}, d = 1, \ldots, D, r = 0, 1, \ldots, M)$ such that*

$$\Omega(x; \Pi) + \theta \ln(1-x) > 0, \ 0 \leq x \leq \eta. \tag{5.9}$$

Then BP(n) *can recover at least ηK input packets with a high probability.*

In the next chapter, we will use (5.9) to study the achievable rates of BP(n) and the design of the outer codes. Readers may skip the remainder of this chapter if the derivation of Theorem 5.2 is not of primary interest.

5.2 ASYMPTOTIC ANALYSIS: DIFFERENTIAL EQUATION APPROACH

In this section, we study the following question: When BP(n) stops, how much input packets would have been decoded in the case that n is sufficiently large? Some existing methods for analyzing BP decoding of erasure codes can be modified to analyze BP decoding of BATS codes. In this section, we adopt the differential equation approach [87] that has been used to analyze Tornado codes [42] (see also [62] for an analysis of LDPC codes over erasure channel). Note that many technical details have to be properly considered in order to apply the differential equation approach. These will be elaborated in this section. This appraoch was published in [97].

5.2.1 RANDOM DECODING GRAPH

The BP decoding process is better described using a random bipartite graph, called the decoding graph. The decoding graph $\mathcal{T}$ has K variable nodes and n check nodes, where the i-th variable node corresponds to the i-th input packet and the j-th check node corresponds to the j-th batch. We equate a variable node with an input packet, and a check node with a batch. Check node j is connects to dg_j variable nodes chosen uniformly at random. Denote by $\mathrm{BATS}(K, n, \Pi)$ the random vector $((\mathrm{dg}_i, \mathbf{G}_i, \mathbf{H}_i)_{i=1}^n, \mathcal{T})$. See an example of $\mathcal{T}$ in Figure 5.1, which is the same as the encoding graph in Figure 2.1 except that the two stages of the encoding are combined together.

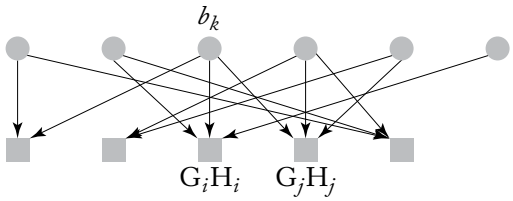


Figure 5.1: A decoding graph. Nodes on the first row are the variable nodes representing the input packets. Nodes on the second row are the check nodes representing the batches.

A check node i is decodable if $\mathrm{dg}_i = \mathrm{rk}(\mathbf{G}_i\mathbf{H}_i)$, where the latter is called the *effective rank* of a batch/check node. In other words, a check node is decodable if its degree and its effective rank are the same. An edge in $\mathcal{T}$ is said to be of degree d and effective rank r if it is connected to a check node with degree d and effective rank r. Let $E_{d,r}$ be the number of edges of degree d and effective rank r. Note that $E_{d,r}/d$ gives the number of check nodes with degree d and effective rank r.

Recall

$$\zeta_r^{d,k} = \frac{\zeta_r^d \zeta_r^k}{\zeta_r^r q^{(d-r)(k-r)}},$$

which is the probability that a totally random $d \times k$ matrix over $\mathbb{F}_q$ has rank r. Define

$$\rho_{d,r} = d \sum_{k=r}^{M} \zeta_r^{d,k} \Pi_{d,k}. \tag{5.10}$$

The following lemma characterizes the initial status of the decoding graph.

Lemma 5.3 *With probability at least* $1 - \lambda(n) - 2M^2 D \exp(-2n^{2/3})$,

$$\left| \frac{E_{d,r}}{n} - \rho_{d,r} \right| = \mathrm{O}(n^{-1/6}), \quad 1 \le r \le M, r \le d \le D. \tag{5.11}$$

Proof. We study the probability of event (5.11) under the condition (5.1), which holds with probability at least $1-\lambda(n)$. With an abuse of notation, we treat $\pi_{d,k}$ as an instance satisfying (5.1) in the rest of this proof, i.e., the decoding graph has $n\pi_{d,k}$ check nodes with degree d and transfer matrix rank k.

Using the property of a totally random matrix and some counting techniques in projective space (see Andrews [4] and Gadouleau and Yan [16]), we have

$$\Pr\{\mathrm{rk}(\mathbf{G}_i\mathbf{H}_i)=r|\mathrm{dg}_i=d,\mathrm{rk}(\mathbf{H}_i)=k\}=\zeta_r^{d,k}. \tag{5.12}$$

In other words, for a batch with degree d and transfer matrix rank k, the batch has an effective rank r ($r\le k, r\le d$) with probability $\zeta_r^{d,k}$. Denote by $X_{d,k,r}$ the number of check nodes with degree d, transfer matrix rank k, and effective rank r. We know that $E_{d,r}=d\sum_{k\ge r}X_{d,k,r}$. Applying Hoeffding's inequality, with probability at least $1-2M^2D\exp(-2n^{2/3})$,

$$\left|\frac{X_{d,k,r}}{n\pi_{d,k}}-\zeta_r^{d,k}\right|<n^{-1/6}, 1\le d\le D, 0\le k\le M, 0\le r\le\min\{d,r\},$$

which implies

$$\left|\frac{E_{d,r}}{n}-d\sum_{k=r}^{M}\pi_{d,k}\zeta_r^{d,k}\right|=\mathrm{O}(n^{-1/6}). \tag{5.13}$$

Moreover,

$$\begin{aligned}\left|\frac{E_{d,r}}{n}-\rho_{d,r}\right| &= \left|\frac{E_{d,r}}{n}-d\sum_{k=r}^{M}\pi_{d,k}\zeta_r^{d,k}+d\sum_{k=r}^{M}\pi_{d,k}\zeta_r^{d,k}-\rho_{d,r}\right| \\ &\le \left|\frac{E_{d,r}}{n}-d\sum_{k=r}^{M}\pi_{d,k}\zeta_r^{d,k}\right|+d\sum_{k=r}^{M}\left|\pi_{d,k}-\Pi_{d,r}\right|\zeta_r^{d,k}.\end{aligned}$$

By (5.13) and (5.1), we have

$$\left|\frac{E_{d,r}}{n}-\rho_{d,r}\right|=\mathrm{O}(n^{-1/6})$$

with probability at least $1-2M^2D\exp(-2n^{2/3})$. The proof is completed by subtracting the probability that condition (5.1) does not hold. □

5.2.2 DENSITY EVOLUTION

Compared with the analysis of fountain codes, a batch has a relatively complex decoding criteria that involves the uniquely decodability of a linear equation system. In addition to the evolution of the degrees of the check nodes, the evolution of the effective ranks of the check nodes also needs to be tracked in the decoding analysis.

Consider the evolution of BATS(K,n,Π) during the decoding process using decoder BP(n), where time t starts at 0 and increases by 1 for each variable node removed by the decoder.

Suppose that in graph $\mathcal{T}$, all the edges connected to the decoded input packets are deleted, and denote the residual graph at time t by $\mathcal{T}^{(t)}$. As we discussed in Section 2.3.1, at time $t \geq 0$ the generator matrix of the i-th batch becomes $\mathbf{G}_i^{(t)}$ and the remaining contributors of the i-th batch form the index set $A_i^{(t)}$. At time $t \geq 0$, a batch is decodable if its degree $|A_i^{(t)}|$ is equal to its effective rank $\text{rk}(\mathbf{G}_i^{(t)}\mathbf{H}_i)$. For $t \geq 0$, $1 \leq r \leq M$ and $r \leq d \leq D$, let $E_{d,r}^{(t)}$ denote the number of edges in $\mathcal{T}^{(t)}$ of degree d and effective rank r, where $E_{d,r}^{(0)} = E_{d,r}$. Note that BP(n) will stop at the first time t such that $\sum_{r=1}^{M} E_{r,r}^{(t)} = 0$.

Upon removing a neighboring variable node of a check node with degree d and rank r, the degree of the check node will change to $d-1$. The effective rank of the check node may remain unchanged or may change to $r-1$. Regarding a degree - effective-rank pair as a state, the state transition of a check node during the decoding process is illustrated in Figure 5.2, where the transition probability is characterized in the following lemma. Define

$$\alpha_{d,r} = \frac{1-q^{-d+r}}{1-q^{-d}},$$

and

$$\bar{\alpha}_{d,r} = 1 - \alpha_{d,r}.$$

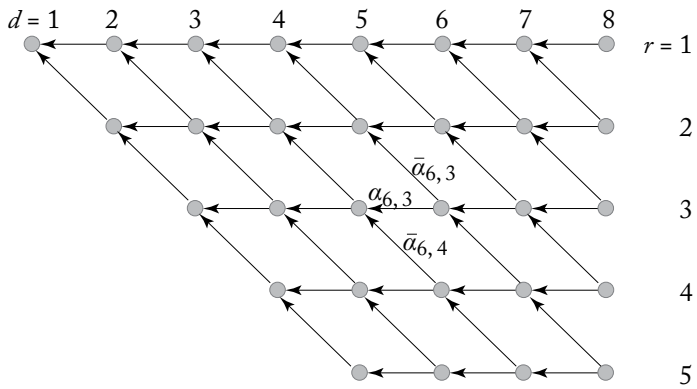


Figure 5.2: The state transition diagram for $M = 5$ and $D = 8$. Each node in the graph represents a degree - effective-rank pair. In each decoding step, if the check node connects to the decoded variable node, its state changes according to the direction of the outgoing edges of its current state. The label on an edge shows the probability that a direction is chosen.

Lemma 5.4 *For any check node i, $1 \leq r \leq M$ and $r \leq d \leq D$,*

$$\begin{aligned}
\Pr\left\{rk(\mathbf{G}_i^{(t+1)}\mathbf{H}_i) = r \,\middle|\, rk(\mathbf{G}_i^{(t)}\mathbf{H}_i) = r, |A_i^{(t+1)}| = d-1, |A_i^{(t)}| = d\right\} &= \alpha_{d,r}, \\
\Pr\left\{rk(\mathbf{G}_i^{(t+1)}\mathbf{H}_i) = r-1 \,\middle|\, rk(\mathbf{G}_i^{(t)}\mathbf{H}_i) = r, |A_i^{(t+1)}| = d-1, |A_i^{(t)}| = d\right\} &= \bar{\alpha}_{d,r}.
\end{aligned}$$

Proof. We omit the index i in the proof to simplify the notation. By (5.12) and the fact that $\mathbf{G}_i$ is totally random, we have for $k \geq r$,

$$\begin{aligned}\Pr\left\{\mathrm{rk}(\mathbf{G}^{(t)}\mathbf{H}) = r \middle| \mathrm{rk}(\mathbf{G}^{(t+1)}\mathbf{H}) = r, |A^{(t+1)}| = d-1, |A^{(t)}| = d, \mathrm{rk}(\mathbf{H}) = k\right\} &= q^{r-k},\\ \Pr\left\{\mathrm{rk}(\mathbf{G}^{(t+1)}\mathbf{H}) = r \middle| |A^{(t+1)}| = d-1, |A^{(t)}| = d, \mathrm{rk}(\mathbf{H}) = k\right\} &= \zeta_r^{d-1,k},\\ \Pr\left\{\mathrm{rk}(\mathbf{G}^{(t)}\mathbf{H}) = r \middle| |A^{(t+1)}| = d-1, |A^{(t)}| = d, \mathrm{rk}(\mathbf{H}) = k\right\} &= \zeta_r^{d,k}.\end{aligned}$$

Hence,

$$\begin{aligned}&\Pr\left\{\mathrm{rk}(\mathbf{G}^{(t+1)}\mathbf{H}) = r \middle| \mathrm{rk}(\mathbf{G}^{(t)}\mathbf{H}) = r, |A^{(t+1)}| = d-1, |A^{(t)}| = d, \mathrm{rk}(\mathbf{H}) = k\right\}\\ &= q^{r-k}\frac{\zeta_r^{d-1,k}}{\zeta_r^{d,k}}\\ &= \frac{1-q^{-d+r}}{1-q^{-d}}.\end{aligned}$$

The proof is completed by multiplying

$$\Pr\left\{\mathrm{rk}(\mathbf{H}) = k \middle| \mathrm{rk}(\mathbf{G}^{(t)}\mathbf{H}) = r, |A^{(t+1)}| = d-1, |A^{(t)}| = d\right\}$$

on both sides of the above equality and taking summation over all $k \geq r$. □

Define $\rho_{d,r}^{(0)} = \rho_{d,r}$ and for $0 \leq i < d$,

$$\rho_{d,r}^{(i+1)} = \alpha_{d-i,r}\rho_{d,r}^{(i)} + \bar{\alpha}_{d-i,r+1}\rho_{d,r+1}^{(i)}. \tag{5.14}$$

The following lemma can readily be proved by induction.

Lemma 5.5 *For* $0 \leq t \leq d$,

$$\rho_{d,r}^{(t)} = d\sum_{k=r}^{M}\zeta_r^{d-t,k}\Pi_{d,k}.$$

5.2.3 EXPECTED DENSITY EVOLUTION

Assume that the decoding process has not stopped. At time t, we have $K-t$ variable nodes left in $\mathcal{T}^{(t)}$, and an edge with degree equal to the rank is uniformly chosen to be removed. Let

$$\bar{E}^{(t)} \triangleq (E_{d,r}^{(t)} : 1 \leq r \leq M, r \leq d \leq D).$$

The random process $\{\bar{E}^{(t)}\}$ is a Markov chain, which suggests a straightforward approach to compute all the transition probabilities in the Markov chain. However, this approach leads to a

complicated formula. Instead of taking this approach, we work out the expected change $E_{d,r}^{(t+1)} - E_{d,r}^{(t)}$ explicitly for all $t \geq 0$. Let

$$E_0^{(t)} = \sum_{r=1}^{M} E_{r,r}^{(t)}.$$

We do not need to study the behavior of $E_{r,r}^{(t)}$ for individual values of r since $E_0^{(t)}$ is sufficient to determine when the decoding process stops. Specifically, the decoding process stops as soon as $E_0^{(t)}$ becomes zero.

We care about when $E_0^{(t)}$ goes to zero for the first time t. The evolution of $E_0^{(t)}$ depends on that of $E_{d,r}^{(t)}$, $1 \leq r \leq M$, and $r < d \leq D$.

Lemma 5.6 *For any constant $c \in (0,1)$, for $t \leq cK$ and $E_0^{(t)} > 0$,*

$$\mathbb{E}\left[E_{d,r}^{(t+1)} - E_{d,r}^{(t)} \middle| \bar{E}^{(t)}\right] = \left(\alpha_{d+1,r} E_{d+1,r}^{(t)} + \bar{\alpha}_{d+1,r+1} E_{d+1,r+1}^{(t)} - E_{d,r}^{(t)}\right) \frac{d}{K-t}, \; 1 \leq r \leq M, r \leq d \leq D, \quad (5.15)$$

and

$$\mathbb{E}\left[E_0^{(t+1)} - E_0^{(t)} \middle| \bar{E}^{(t)}\right] = \frac{\sum_r r\alpha_{r+1,r} E_{r+1,r}^{(t)}}{K-t} - \frac{E_0^{(t)}}{K-t} - 1 + \mathrm{O}(1/K). \quad (5.16)$$

Proof. Fix a time $t \geq 0$. With an abuse of notation, we treat $\bar{E}^{(0)}, \ldots, \bar{E}^{(t)}$ as instances in the proof, i.e., the values of these random vectors are fixed. Let (U, V) be the edge chosen to be removed at time t, where V is the variable node and U is the check node, according decoder BP(n). Note that V is uniformly distributed among all variable nodes and U must be a check node with degree equal to the rank at time t. See an illustration in Figure 5.3.

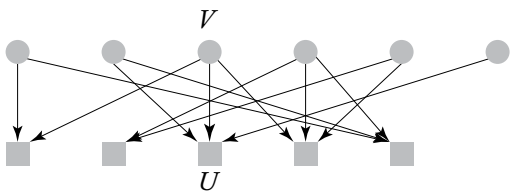


Figure 5.3: A decoding graph. Edge (U, V) is to be removed at time t.

Let $N_{d,r}$ be the number of check nodes which has degree d and effective rank r at time t and has degree $d-1$ at time $t+1$. Let $N_{d,r}^{+}$ (resp. $N_{d,r}^{-}$) be the number of check nodes which

has degree d and effective rank r at time t and has degree $d-1$ and effective rank r (resp. $r-1$) at time $t+1$. Clearly, $N^+_{d,r} + N^-_{d,r} = N_{d,r}$. The difference $E^{(t+1)}_{d,r} - E^{(t)}_{d,r}$ can then be expressed as

$$E^{(t+1)}_{d,r} - E^{(t)}_{d,r} = d(N^+_{d+1,r} + N^-_{d+1,r+1} - N_{d,r}). \tag{5.17}$$

The probability that a check node with degree d and effective rank r, $d > r$, connects to the variable node V at time t is $d/(K-t)$. Therefore, when $d > r$,

$$N_{d,r} \sim \text{Binom}\left(\frac{E^{(t)}_{d,r}}{d}, \frac{d}{K-t}\right).$$

As we characterize in Lemma 5.4, for a check node with degree d and effective rank r connecting to the variable node V at time t, its effective rank will become r (resp. $r-1$) with probability $\alpha_{d,r}$ (resp. $\bar{\alpha}_{d,r}$) at time $t+1$. So when $d > r$,

$$N^+_{d,r} \sim \text{Binom}\left(\frac{E^{(t)}_{d,r}}{d}, \alpha_{d,r}\frac{d}{K-t}\right),$$
$$N^-_{d,r} \sim \text{Binom}\left(\frac{E^{(t)}_{d,r}}{d}, \bar{\alpha}_{d,r}\frac{d}{K-t}\right).$$

The expectation in (5.15) is obtained by taking expectation on (5.17).

To verify (5.16), note that $N^+_{r,r} = 0$ and hence $N^-_{r,r} = N_{r,r}$. Then we have

$$\begin{aligned} E^{(t+1)}_0 - E^{(t)}_0 &= \sum_r \left(E^{(t+1)}_{r,r} - E^{(t)}_{r,r}\right) \\ &= \sum_r r N^+_{r+1,r} - \sum_r N_{r,r}. \end{aligned} \tag{5.18}$$

For a check node with degree r and effective rank r, with probability $r/E^{(t)}_0$ it is U, and hence is connected to V, otherwise, with probability $r/(K-t)$ it is connected to V. Therefore,

$$N_{r,r} \sim \text{Binom}\left(\frac{E^{(t)}_{r,r}}{r}, \frac{r}{E^{(t)}_0} + \left(1 - \frac{r}{E^{(t)}_0}\right)\frac{r}{K-t}\right).$$

Taking expectation on both sides of (5.18), we have

$$\begin{aligned} &\mathbb{E}\left[E^{(t+1)}_0 - E^{(t)}_0 | \bar{E}^{(t)}\right] \\ &= \sum_r r\alpha_{r+1,r}\frac{E^{(t)}_{r+1,r}}{K-t} - \sum_r \left(\frac{E^{(t)}_{r,r}}{E^{(t)}_0} + \left(1 - \frac{r}{E^{(t)}_0}\right)\frac{E^{(t)}_{r,r}}{K-t}\right) \\ &= \sum_r r\alpha_{r+1,r}\frac{E^{(t)}_{r+1,r}}{K-t} - \frac{E^{(t)}_0}{K-t} - 1 + \sum_r \frac{r}{E^{(t)}_0}\frac{E^{(t)}_{r,r}}{K-t}. \end{aligned}$$

The expectation in (5.16) is obtained by $\sum_r \frac{r}{E^{(t)}_0}\frac{E^{(t)}_{r,r}}{K-t} < \frac{M^2}{K(1-c)}$ since $t \le cK$. □

5.2.4 SUFFICIENT AND NECESSARY CONDITIONS

The expected changes in Lemma 5.6 can be approximated by the solution of a differential equation. Consider the system of differential equations

$$\frac{\mathrm{d}\,\rho_{d,r}(\tau)}{\mathrm{d}\,\tau} = \left(\alpha_{d+1,r}\rho_{d+1,r}(\tau) + \bar{\alpha}_{d+1,r+1}\rho_{d+1,r+1}(\tau) - \rho_{d,r}(\tau)\right)\frac{d}{\theta-\tau}, \quad 1 \le r \le M, r < d \le D, \tag{5.19}$$

$$\frac{\mathrm{d}\,\rho_0(\tau)}{\mathrm{d}\,\tau} = \frac{\sum_{r=1}^{D-1} r\alpha_{r+1,r}\rho_{r+1,r}(\tau) - \rho_0(\tau)}{\theta-\tau} - 1 \tag{5.20}$$

with initial values $\rho_{d,r}(0) = \rho_{d,r}$ and $\rho_0(0) = \sum_r \rho_{r,r}$, where $\theta = K/n$ is the design rate of the BATS code.

We can obtain some intuition about how the system of differential equations is related to the problem of our interest. Substitute $E_{d,r}^{(t)}$ and $E_0^{(t)}$ with $n\rho_{d,r}(t/n)$ and $n\rho_0(t/n)$, respectively, in (5.15) and (5.16). Defining $\tau = t/n$ and letting $n \to \infty$, we obtain the system of differential equations in (5.19) and (5.20). The expectation operations are ignored because $\rho_{d,r}(\tau)$ and $\rho_0(\tau)$ are deterministic functions. Intuitively, $E_{d,r}^{(t)}$ and $E_0^{(t)}$, although random, behave like deterministic functions $n\rho_{d,r}(t/n)$ and $n\rho_0(t/n)$ when n is large.

The above intuition can be made rigorous based on a general theorem by [86, 87]. Due to the different problem model, we provide a modified version, Theorem A.1 in Appendix A, which is different from the ones used in [42, 62].

The system of differential equations in (5.19) and (5.20) can be solved explicitly (see Appendix A.3). In particular, the solution for $\rho_0(\tau)$ is

$$\rho_0(\tau) = \left(1-\frac{\tau}{\theta}\right)\left(\sum_{r=1}^{M}\alpha_{r+1,r}\sum_{d=r+1}^{D}\rho_{d,r}^{(d-r-1)}\,\mathrm{I}_{d-r,r}\left(\frac{\tau}{\theta}\right) + \sum_{r=1}^{M}\rho_{r,r} + \theta\ln(1-\tau/\theta)\right),$$

where $\mathrm{I}_{d-r,r}$ is defined in (5.6). The above formula of $\rho_0(\tau)$ can be simplified by substituting the expressions of $\alpha_{r+1,r}$, $\rho_{d,r}^{(d-r-1)}$, and $\rho_{r,r}$ as

$$\rho_0(\tau) = \left(1-\frac{\tau}{\theta}\right)\left(\sum_{r=1}^{M}\sum_{d=1}^{D} d\;\mathrm{I}_{d-r,r}\left(\frac{\tau}{\theta}\right)\sum_{k=r}^{M}\frac{\zeta_r^k}{q^{k-r}}\Pi_{d,k} + \theta\ln(1-\tau/\theta)\right). \tag{5.21}$$

For $\eta \in (0,1)$, the following theorem shows that if $\rho_0(\tau) > 0$ for $\tau \in [0, \eta\theta]$, then the decoding does not stop until $t > \eta K$ with high probability, and $E_{d,r}^{(t)}$ and $E_0^{(t)}$ can be approximated by $n\rho_{d,r}(t/n)$ and $n\rho_0(t/n)$, respectively.

Theorem 5.7 *Consider a sequence of decoding graphs BATS(K, n, Ψ, h), $n = 1, 2, \ldots$ with fixed $\theta = K/n$. For $\eta \in (0,1)$,*

(i) if $\rho_0(\tau) > 0$ *for* $\tau \in [0, \eta\theta]$, *then for sufficiently large* K, *with probability* $1 - \mathrm{O}(n^{7/24}\exp(-n^{1/8})) - \lambda(n)$, *the decoding terminates with at least* ηK *variable nodes decoded, and*

$$|E_{d,r}^{(t)} - n\rho_{d,r}(t/n)| = \mathrm{O}(n^{5/6}),\ 1 \le r \le M, r < d \le M,$$
$$|E_0^{(t)} - n\rho_0(t/n)| = \mathrm{O}(n^{5/6})$$

uniformly for $t \in [0, \eta K]$*; and*

(ii) if $\rho_0(\tau) < 0$ *for some* $\tau \in [0, \eta\theta]$, *then for sufficiently large* K, *with probability* $1 - \mathrm{O}(n^{7/24}\exp(-n^{1/8})) - \lambda(n)$, *the decoding terminates before* ηK *variable nodes are decoded.*

See Appendix A.2 for the proof of the above theorem.

5.3 TREE ANALYSIS OF BATS CODES

We provide an analysis of the BP decoding performance of the generalized BATS codes using a *tree-based approach*, which extends the tree analysis proposed in [41]. Compared with the differential equation approach, the tree-based approach is more intuitive: the "strange" formula obtained previously by solving a differential equation now has a simple interpretation. This analysis approach was published in [98].

5.3.1 AN EXTENSION OF AND-OR TREE ANALYSIS

We first discuss an extension of the And-Or tree analysis [41]. For an integer $\ell \ge 0$, define a random tree Γ_ℓ formed by variable nodes and check nodes with $\ell + 1$ levels as follows: The root of the tree is at level 0 (the highest) and the leaves of tree is at level ℓ (the lowest). Each node at an even level is a variable node, and each node at an odd level is a check node. For $i \ge 1$, a non-leaf variable node has $i - 1$ children at the next lower level with probability α_i, i.e., $\alpha_i \ge 0$ and $\sum_{i=1}^{\infty} \alpha_i = 1$. For $d \ge 1$ and $r \ge 0$, a check node has degree d and transfer matrix rank r with probability $\beta_{d,r}$, i.e., $\beta_{d,r} \ge 0$ and $\sum_{d=1}^{\infty}\sum_{r=0}^{\infty} \beta_{d,r} = 1$. A non-leaf check node with degree d has $d - 1$ children at the next lower level. Nodes at level ℓ have no children. An instance of Γ_3 is illustrated in Figure 5.4.

Same as the check nodes in the decoding graph $\mathcal{T}$ of a BATS code, a check node in Γ_ℓ also associates with the variables of a batch. The difference is that the adjacent variable nodes of a check node are not chosen uniformly at random among all the variable nodes. The BP decoding algorithm of BATS codes can be applied on Γ_ℓ as well. Note that Γ_ℓ does not degenerate to the And-Or tree discussed in [41] even when all the batches are of size 1. We are interested in

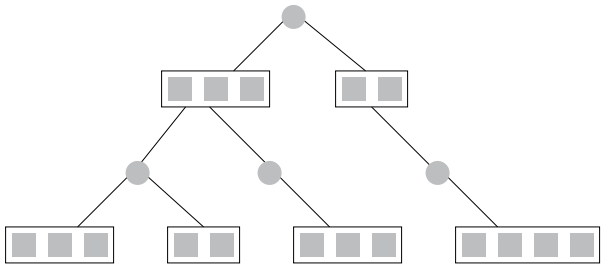


Figure 5.4: An instance of Γ_3 with $\alpha_2 = 0.5, \alpha_3 = 0.5, \beta_{2,2} = 0.25, \beta_{3,2} = 0.5$, and $\beta_{4,3} = 0.25$. The transfer matrix ranks are not denoted in the figure.

the probability that the root of Γ_ℓ can be recovered when the BP decoding of BATS codes is applied.

Lemma 5.8 *For $\ell \geq 1$, let x_ℓ be the probability that the root of $\Gamma_{2\ell-1}$ can be recovered by applying the BP decoding of BATS codes. Then for $i = 1, \ldots, \ell$*

$$y_i = \sum_{d=1}^{\infty}\sum_{r=1}^{\infty}\beta_{d,r}\sum_{s=(d-r)^+}^{d-1}\binom{d-1}{s}x_{i-1}^s(1-x_{i-1})^{d-1-s}\zeta_{d-s}^r,$$
$$x_i = 1 - \sum_{k=1}^{\infty}\alpha_k(1-y_i)^{k-1},$$

where $x_0 = 0$ and ζ_{d-s}^r is defined in (5.3).

Proof. We claim that for $i = 1, \ldots, \ell$,

1. x_i is the probability that a variable node v at level $2(\ell - i)$ can be recovered by applying the BP decoding of BATS codes on the subtree of $\Gamma_{2\ell-1}$ formed by v and all its descendants; and

2. y_i is the probability that a check node c at level $2(\ell - i) + 1$ is decodable by applying the BP decoding of BATS codes on the subtree of $\Gamma_{2\ell-1}$ formed by c and all its descendants.

We prove the claim by induction on i. First, consider a check node c at level $2\ell - 1$ with degree d and transfer matrix rank r. Since c does not have a child, it is decodable with probability ζ_d^r when $1 \leq d \leq r$ and zero otherwise. Thus, the claim for y_1 is proved.

Assume that the claim holds for y_i for some $1 \leq i \leq \ell$. Consider the subtree formed by a variable node v at level $2(\ell - i)$ and all its descendants. Node v can be recovered if at least one of its children check node is decodable. By the induction hypothesis, each children of v is decodable with probability y_i independently. Hence, the claim for x_i holds.

Assume that the claim holds for y_i for some $1 \leq i < \ell$. Fix a check node c at level $2(\ell - i) - 1$ with degree $d \geq 1$ and transfer matrix rank $r \geq 1$. Consider the BP decoding on the subtree formed by node c and all its descendants. Note that if $r = 0$, the node c cannot be decodable since $d \geq 1$. Consider $r \geq 1$. The probability that node c is decodable when s of its $d - 1$ children are recovered is ζ^r_{d-s}, where $(d - r)^+ \leq s \leq d - 1$. Since each child of v is recovered with probability x_i independently, the probability that node c is decodable is $\sum_{s=(d-r)^+}^{d-1} \binom{d-1}{s} x_i^s (1 - x_i)^{d-1-s} \zeta^r_{d-s}$. The claim for y_{i+1} is proved by considering the probability of node c which has degree d and transfer matrix rank r. □

5.3.2 TREE ANALYSIS OF BP DECODING

For the convenience of the tree-based analysis, we give a different version of the BP(n) decoder, called the BP$'$(n) decoder. The input of BP$'$(n) is the sequence $(\mathbf{Y}_i, \mathbf{G}_i\mathbf{H}_i)$, $i = 1 \ldots, n$. The decoder also knows the index set of the input packets involved in each batch. The BP$'$(n) decoder includes multiple iterations. In the first iteration, all the decodable batches are decoded (by solving the associated linear system of equations (2.1)), and the input packets involved in these decodable batches are recovered. In each of the following iterations, undecoded batches are first updated: For each input packet involved in the i-th batch, if this input packet has already been recovered, its value is substituted into (2.1) and the degree of the batch is reduced by one. Then the batches which become decodable after the updating are decoded, and the input packets involved in these decodable batches are recovered. As we have discussed in Section 2.3.5, BP$'$(n) and BP(n) must stop with the same subset of input packets recovered.

Theorem 5.9 *Fix an integer $D > 0$, real numbers $\theta > 0$ and $0 < \eta < 1$. Consider a sequence of BATS codes with K input packets and $n = \lceil K/\theta \rceil$ batches, where the empirical distribution of the batch degree and the transfer matrix rank of all batches converges to Π such that*

$$\Omega(x; \Pi) + \theta \ln(1 - x) > 0, \ 0 \leq x \leq \eta. \tag{5.22}$$

Then, for all sufficiently large K, BP$'$(n) can recover at least ηK input packets with probability at least $1 - \exp(-cK)$, where c has a value independent of K.

We now prove Theorem 5.9. The structure of the proof is similar to the structure of tree analysis in the existing literature. First, we define a random tree and prove that the subgraph expanded from each variable node in the decoding graph $\mathcal{T}$ including all the nodes within its ℓ neighborhood is a tree with a sufficiently large probability. Then, we bound the probability that a variable node is recoverable after ℓ iterations of the BP decoding, in terms of the probability that the root of the corresponding random tree is decodable. Lastly, we bound the number of variable node that can be recovered using a martingale argument. Since some techniques are similar to the ones used for LT codes [50] and LDPC codes [43], we omit these details and refer to the corresponding papers.

Let $\bar{\Pi}_{\deg} = \sum_{d,r} d\,\Pi_{d,r}$ and $\lambda = \bar{\Pi}_{\deg}/R$. For $\ell \geq 0$, define a random tree T_ℓ^* in the same way as Γ_ℓ with the distributions

$$\beta_{d,r} = \begin{cases} \frac{d\,\Pi_{d,r}}{\bar{\Pi}_{\deg}}, & 1 \leq d \leq D, 0 \leq r \leq M, \\ 0, & \text{otherwise}, \end{cases} \tag{5.23}$$

and

$$\alpha_k = \begin{cases} \frac{1}{1-\epsilon_{\bar{k}}} \frac{\lambda^{k-1} e^{-\lambda}}{(k-1)!}, & k = 1, \ldots, \bar{k}, \\ 0, & \text{otherwise}, \end{cases} \tag{5.24}$$

where $\bar{k} > 0$ is an integer and $\epsilon_{\bar{k}} = \sum_{k=\bar{k}}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!}$. The meanings of these two distributions defined above will be made clear in the proof of the following lemma. Let $\mathcal{T}_\ell^*(\bar{k})$ be the set of possible instances of T_ℓ^*. Note that the elements in $\mathcal{T}_\ell^*(\bar{k})$ are trees.

Fix a variable node v in the decoding graph $\mathcal{T}$ representing the BATS code with K variable nodes and n check nodes. For $\ell \geq 0$, the ℓ-neighborhood of variable node v, denoted by $T_\ell(v)$, is the subgraph of G that includes all the nodes with distance less than or equal to ℓ from node v. After ℓ iterations of the BP decoding, whether or not variable node v is recovered is determined by $T_{2\ell-1}(v)$. Since $T_\ell(v)$ has the same distribution for all variable nodes v, we may also write $T_\ell(v)$ as T_ℓ when v is implied.

For a fixed $\ell > 0$, the next lemma bounds the probability that T_ℓ is a tree of bounded variable and check node degrees when K is sufficiently large.

Lemma 5.10 *Fix integers $\bar{k} > 0$ and $\ell \geq 0$. There exists $c_{\ell,\bar{k}}$ with $0 \leq c_{\ell,\bar{k}} \leq \lceil \ell/2 \rceil (D\bar{k})^{\lfloor \frac{\ell-1}{2} \rfloor} + \ell$ such that when K is sufficiently large, for any $G_\ell \in \mathcal{T}_\ell(\bar{k})$,*

$$\Pr\{T_\ell = G_\ell\} \geq (1 - c_{\ell,\bar{k}} \epsilon_{\bar{k}}) \Pr\{T_\ell^* = G_\ell\}.$$

Proof. We prove the lemma by induction on ℓ. Both T_0^* and T_0 include only one variable node, so the lemma holds for $\ell = 0$. Suppose the lemma holds for some $\ell \geq 0$. Fix $G_{\ell+1} \in \mathcal{T}_{\ell+1}(\bar{k})$, which by definition is a tree. We have

$$\Pr\{T_{\ell+1} = G_{\ell+1}\} = \Pr\{T_{\ell+1} = G_{\ell+1} | T_\ell = G_\ell\} \Pr\{T_\ell = G_\ell\},$$

where G_ℓ is the subgraph of $G_{\ell+1}$ obtained by removing the nodes at level $\ell + 1$. By the induction hypothesis, we have

$$\Pr\{T_\ell = G_\ell\} \geq \Pr\{T_\ell^* = G_\ell\}(1 - c_{\ell,\bar{k}} \epsilon_{\bar{k}}).$$

To complete the proof, we bound $\Pr\{T_{\ell+1} = G_{\ell+1} | T_\ell = G_\ell\}$ by considering two cases: ℓ is even and ℓ is odd.

First, consider that ℓ is odd. Suppose G_ℓ has N *check* nodes at level ℓ, which are also at level ℓ of $G_{\ell+1}$. Denote by $\deg(i)$ and $\mathrm{tmr}(i)$ the degree and the transfer matrix rank of the i-th check node at level ℓ in $G_{\ell+1}$, respectively. Let E_0 be the event $\{T_\ell = G_\ell\}$. For $i = 1, \ldots, N$, let E_i be the event that the i-th check node at level ℓ of $T_{\ell+1}$ has degree $\deg(i)$ and transfer matrix rank $\mathrm{tmr}(i)$, and the $\deg(i) - 1$ children are not shared by the first $i-1$ check nodes at level ℓ of $T_{\ell+1}$. Then

$$\begin{aligned}\Pr\{T_{\ell+1} = G_{\ell+1} | T_\ell = G_\ell\} &= \Pr\{E_i, i = 1, \ldots, N | E_0\} \\ &= \prod_{i=1}^{N} \Pr\{E_i | E_0, \ldots, E_{i-1}\}. \end{aligned} \tag{5.25}$$

Note that N and the number of nodes in G_ℓ is upper bounded by $(D\bar{k})^{\ell/2}$, which does not change with K. Since the maximum degree of a batch D is a constant, the probability that the i-th check nodes at level ℓ in $T_{\ell+1}$ have a child in common with the first $i-1$ check nodes is bounded by $O(1/K)$, which converges to 0 as $K \to \infty$. Similar to the discussion for LT codes [50], as $K \to \infty$, $\Pr\{E_i | E_0, \ldots, E_{i-1}\}$ converges to $\deg(i)\Pi_{\deg(i),\mathrm{tmr}(i)}/\bar{\Pi}_{\deg}$, which is equal to $\beta_{\deg(i),\mathrm{tmr}(i)}$ by definition. Hence, as $K \to \infty$,

$$\begin{aligned}\Pr\{T_{\ell+1} = G_{\ell+1} | T_\ell = G_\ell\} &\to \prod_{i=1}^{N} \beta_{\deg(i),\mathrm{tmr}(i)} \\ &= \Pr\{T^*_{\ell+1} = G_{\ell+1} | T^*_\ell = G_\ell\},\end{aligned}$$

where the equality follows from the definition of $T^*_{\ell+1}$ with the degree distributions in (5.23) and (5.24). Thus, for a sufficiently large K, $\Pr\{T_{\ell+1} = G_{\ell+1} | T_\ell = G_\ell\} \geq \Pr\{T^*_{\ell+1} = G_{\ell+1} | T^*_\ell = G_\ell\}(1 - \epsilon_{\bar{k}})$.

Second, consider that ℓ is even. Suppose G_ℓ has N *variable* nodes at level ℓ, which are also at level ℓ of $G_{\ell+1}$. Denote by k_i the number of children check nodes of the i-th variable node at level ℓ of $G_{\ell+1}$. We have $k_i < \bar{k}$. Similar to the case that ℓ is odd, let $E_0 = \{T_\ell = G_\ell\}$, and for $i = 1, \ldots, N$, let E_i be the event that the i-th variable node at level ℓ of $T_{\ell+1}$ has k_i children, which are not the children of the first $i-1$ variable nodes at level ℓ of $T_{\ell+1}$. With these events, we have the expression of $\Pr\{T_{\ell+1} = G_{\ell+1} | T_\ell = G_\ell\}$ as in (5.25). Similar to LT codes [50], under the condition that a bounded number of edges in G is fixed, the probability that the degree of a variable node v is d converges to $\frac{\lambda^{d-a} e^{-\lambda}}{(d-a)!}$ as $K \to \infty$, where a is the number of fixed edges incident to v. Therefore, $\Pr\{E_i | E_0, \ldots, E_{i-1}\} \to \frac{\lambda^{k_i} e^{-\lambda}}{k_i!}$ as $K \to \infty$, and hence

$$\Pr\{T_{\ell+1} = G_{\ell+1} | T_\ell = G_\ell\} \to \prod_{i=1}^{N} \frac{\lambda^{k_i} e^{-\lambda}}{k_i!} \quad \text{as } K \to \infty.$$

On the other hand, we have

$$\Pr\{T^*_{\ell+1} = G_{\ell+1} | T^*_\ell = G_\ell\} = \frac{1}{(1 - \epsilon_{\bar{k}})^N} \prod_{i=1}^{N} \frac{\lambda^{k_i} e^{-\lambda}}{k_i!}.$$

Hence, for a sufficiently large K,

$$\begin{aligned}\Pr\{T_{\ell+1} = G_{\ell+1} | T_\ell = G_\ell\} &\geq (1-\epsilon_{\bar{k}})^N \Pr\{T^*_{\ell+1} = G_{\ell+1} | T^*_\ell = G_\ell\}(1-\epsilon_{\bar{k}}) \\ &\geq (1-(N+1)\epsilon_{\bar{k}}) \Pr\{T^*_{\ell+1} = G_{\ell+1} | T^*_\ell = G_\ell\}.\end{aligned}$$

The proof is completed by noting that $N \leq (D\bar{k})^{\ell/2}$. □

5.3.3 RECOVERABLE PROBABILITY OF A VARIABLE NODE

Fix $\ell > 0$ and a sufficiently small $\epsilon > 0$. We say $T^*_{2\ell-1}$ or a tree in $\mathcal{T}_{2\ell-1}(\bar{k})$ is decodable if its root can be recovered by the BP decoding algorithm. Note that $\epsilon_{\bar{k}}$ decreases faster than any polynomial function of $\bar{k}$. For a sufficiently large $\bar{k}$, $c_{\ell,\bar{k}}\epsilon_{\bar{k}} < \epsilon/4$. By Lemma 5.10, for all sufficiently large K,

$$\Pr\{T_{2\ell-1} = G\} \geq \Pr\{T^*_{2\ell-1} = G\}(1-\epsilon/4),$$

and hence

$$\begin{aligned}&\Pr\{T_{2\ell-1} \in \mathcal{T}_{2\ell-1}(\bar{k}) \text{ and is decodable}\} \\ &\geq \sum_{G \in \mathcal{T}_{2\ell-1}(\bar{k})} \Pr\{G \text{ is decodable}\} \Pr\{T_{2\ell-1} = G\} \\ &\geq \sum_{G \in \mathcal{T}_{2\ell-1}(\bar{k})} \Pr\{G \text{ is decodable}\} \Pr\{T^*_{2\ell-1} = G\}\,(1-\epsilon/4) \\ &\geq \Pr\{T^*_{2\ell-1} \text{ is decodable}\} - \epsilon/4. \end{aligned} \tag{5.26}$$

Let $x_l = \Pr\{T^*_{2\ell-1} \text{ is decodable}\}$, which can be characterized by Lemma 5.10. For $i = 1, 2, \ldots$, let

$$\tilde{x}_i = 1 - \exp(-\lambda \tilde{y}_i) \text{ and } \tilde{y}_i = \Omega(\tilde{x}_{i-1}; \Pi)/\bar{\Pi}_{\deg},$$

where $\tilde{x}_0 = 0$. We see that $x_\ell \to \tilde{x}_\ell$ as $\bar{k} \to \infty$. Thus, for a sufficiently large $\bar{k}$, $x_\ell \geq \tilde{x}_\ell - \epsilon/4$. Together with (5.26), we conclude that a variable node v is recoverable and has $T_{2\ell-1}(v) \in \mathcal{T}_{2\ell-1}(\bar{k})$ with probability at least $\tilde{x}_\ell - \epsilon/2$ for all sufficiently large K.

Let A be the number of variable nodes v that can be recovered by ℓ iterations of the BP decoding and have $T_{2\ell-1}(v) \in \mathcal{T}_{2\ell-1}(\bar{k})$, where ℓ has a fixed value (to be determined later) that does not change with K. We have $\mathbb{E}[A] \geq (\tilde{x}_\ell - \epsilon/2)K$ for all sufficiently large K. We use a standard exposure martingale argument to show that $A > \eta K$ with high probability. Specifically, for $i = 1, \ldots, n$, let Z_i be the subgraph formed by the i-th check node and the adjacent variable nodes. Define $\mathbf{X}_i = \mathbb{E}[A|Z_1, \ldots, Z_i]$. By definition, $\mathbf{X}_i$ is a martingale with $\mathbf{X_0} = \mathbb{E}[A]$ and $\mathbf{X}_n = A$. Since the exposure of a check node will affect the degrees of a constant number of variable nodes v with $T_{2\ell-1}(v) \in \mathcal{T}_{2\ell-1}(\bar{k})$, we have $|\mathbf{X}_i - \mathbf{X}_{i-1}| \leq c'$, a constant that does not depend on K. Applying the Azuma-Hoeffding Inequality, we have

$$\Pr\{A \leq \mathbb{E}[A] - \epsilon/2K\} \leq \exp\left(-\frac{\epsilon^2 K}{8c'^2}\right).$$

Hence,

$$\Pr\{A > (\tilde{x}_\ell - \epsilon)K\} > 1 - \exp\left(-\frac{\epsilon^2 K}{8c'^2}\right).$$

Define $f(x) = 1 - \exp(-\Omega(x;\Pi)/R)$. Then $\tilde{x}_i = f(\tilde{x}_{i-1})$. We know that $f(x)$ is an increasing function for $x \in [0,1]$ and by (5.22), $f(x) > x$ for $x \in [0,\eta]$. Hence, the sequence $\tilde{x}_i$, $i = 0, 1, \ldots$ is increasing and converges to a value η' larger than η. Therefore, by letting $\epsilon = (\eta' - \eta)/2$ for a sufficiently large ℓ (independent of K), we have $\tilde{x}_l - \epsilon \geq \eta$ and hence $A \geq \eta K$ with high probability. This completes the proof of Theorem 5.9.

CHAPTER 6

Asymptotic Degree Distribution Optimizations

Following the asymptotic analysis of BP decoding, we now study the achievable rates of BATS codes and the design of degree distributions based on the sufficient condition in Theorem 5.2. We assume that the batch degrees and the transfer matrix ranks are "asymptotically independent." Specifically, there exist a degree distribution $\mathbf{\Psi} = (\Psi_1, \cdots, \Psi_D)$ and a rank distribution $(h_0, h_1, \ldots, h_M)$ such that $\Pi_{d,r} = \Psi_d h_r$, which implies that the empirical batch degree distribution converges in probability to $\mathbf{\Psi}$ and the empirical transfer matrix rank distribution converges in probability to $(h_0, h_1, \ldots, h_M)$. The BATS code model described in Chapter 2 satisfies the above assumption.

6.1 OPTIMIZATION FOR SINGLE RANK DISTRIBUTION

We start with a single rank distribution $(h_0, h_1, \ldots, h_M)$. Since $h_0 = 1 - \sum_{i=1}^{M} h_M$, we can alternatively represent a rank distribution $(h_0, h_1, \ldots, h_M)$ as

$$\mathbf{h} = (h_1, \ldots, h_M),$$

where h_0 is omitted. With the above notations and assumption, we rewrite $\Omega(x; \Pi)$ defined in (5.2) as

$$\Omega(x; \mathbf{h}, \mathbf{\Psi}) = \sum_{d=1}^{D} d\,\Psi_d \sum_{r=1}^{M} h_r \sum_{j=(d-r)^+}^{d-1} \binom{d-1}{j} x^j (1-x)^{d-1-j} \zeta_{d-j}^r. \tag{6.1}$$

Following Lemma 5.1, we can equivalently write $\Omega(x; \mathbf{h}, \mathbf{\Psi})$ as

$$\Omega(x; \mathbf{h}, \mathbf{\Psi}) = \sum_{d=1}^{D} d\,\Psi_d \sum_{r=1}^{M} \hbar_r \, \mathrm{I}_{d-r,r}(x) \tag{6.2}$$

$$= \sum_{r=1}^{M} \hbar_r \sum_{d=r+1}^{D} d\,\Psi_d \, \mathrm{I}_{d-r,r}(x) + \sum_{r=1}^{M} \hbar_r \sum_{d=1}^{r} d\,\Psi_d, \tag{6.3}$$

where $\mathrm{I}_{d-r,r}$ is defined in (5.6), and

$$\hbar_r = \sum_{k=r}^{M} \frac{\zeta_r^k}{q^{k-r}} h_k.$$

See the meaning of $\hbar_r$ in Section 2.6.

For $\eta \in [0, 1)$, we say a rate R is *η-achievable by BATS codes using BP decoding* if for every $\epsilon > 0$ and every sufficiently large K, there exists a BATS code with K input packets such that for any $n \leq \eta K/(R-\epsilon)$ received batches, BP decoding can recover at least ηK input packets with probability at least $1-\epsilon$.

For a given rank distribution $\mathbf{h}$, the following optimization problem maximizes the η-achievable rate with the degree distribution as the variable:

$$\begin{aligned} \max_{\mathbf{\Psi},\theta} \ & \theta \\ \text{s.t. } & \Omega(x;\mathbf{h},\mathbf{\Psi}) + \theta \ln(1-x) \geq 0, \quad 0 \leq x \leq \eta, \\ & \sum_d \Psi_d = 1 \ \text{ and } \ \Psi_d \geq 0, \ d = 1, \cdots, D. \end{aligned} \tag{6.P1}$$

When the context is clear, we also write $\Omega(x;\mathbf{\Psi})$, $\Omega(x;\mathbf{h})$ or $\Omega(x)$ to simplify the notation. Let $\hat{\theta}_{\mathbf{h}}$, or $\hat{\theta}$ when $\mathbf{h}$ is clear from the context, be the optimal value in (6.P1). For a fixed rank distribution $\mathbf{h}$ and a degree distribution $\mathbf{\Psi}$, let $\hat{\theta}_{\mathbf{h},\mathbf{\Psi}}$ be the largest θ such that

$$\Omega(x;\mathbf{h},\mathbf{\Psi}) + \theta \ln(1-x) \geq 0, \ 0 \leq x \leq \eta.$$

Lemma 6.1 *When the empirical rank distribution of the transfer matrices converges in probability to $(h_0, h_1, \ldots, h_M)$ (in the sense of (5.1)), any rate less than or equal to $\eta\hat{\theta}_{\mathbf{h},\mathbf{\Psi}}$ is η-achievable by BATS codes with degree distribution $\mathbf{\Psi}$ using BP decoding.*

Proof. By (5.21), we can write

$$\rho_0(\tau) = (1-\tau/\theta)\left(\Omega(\tau/\theta) + \theta \ln(1-\tau/\theta)\right). \tag{6.4}$$

To show that $\eta\hat{\theta}_{\mathbf{h},\mathbf{\Psi}}$ is η-achievable, by Theorem 5.7, we only need to show that there exists a degree distribution such that for any $\epsilon > 0$,

$$\Omega(x;\mathbf{\Psi}) + (\hat{\theta}_{\mathbf{h},\mathbf{\Psi}} - \epsilon)\ln(1-x) > 0, \quad 0 \leq x \leq \eta. \tag{6.5}$$

Since the proof for $\eta = 0$ is trivial, assume that $\eta > 0$. By the definition of $\hat{\theta}_{\mathbf{h},\mathbf{\Psi}}$, we have

$$\Omega(x;\mathbf{\Psi}) + \hat{\theta}_{\mathbf{h},\mathbf{\Psi}} \ln(1-x) \geq 0, \quad 0 \leq x \leq \eta.$$

Multiplying by $\frac{\hat{\theta}_{\mathbf{h},\mathbf{\Psi}}-\epsilon}{\hat{\theta}_{\mathbf{h},\mathbf{\Psi}}}$ on both sides, we have

$$\frac{\hat{\theta}_{\mathbf{h},\mathbf{\Psi}} - \epsilon}{\hat{\theta}_{\mathbf{h},\mathbf{\Psi}}}\Omega(x;\mathbf{\Psi}) + (\hat{\theta}_{\mathbf{h},\mathbf{\Psi}} - \epsilon)\ln(1-x) \geq 0, \quad 0 \leq x \leq \eta. \tag{6.6}$$

Since $\Omega(x; \boldsymbol{\Psi}) > 0$ for $x > 0$, (6.6) implies that $\boldsymbol{\Psi}$ satisfies (6.5) except possibly for $x = 0$. If $\Omega(0; \boldsymbol{\Psi}) > 0$, which implies $\boldsymbol{\Psi}$ satisfies (6.5), we are done.

In the following, we consider the case with $\Omega(0; \boldsymbol{\Psi}) = 0$. By the definition of Ω in (6.2), we have

$$\Omega(0; \boldsymbol{\Psi}) = \sum_{r=1}^{M} r \Psi_r \sum_{s=r}^{M} \hbar_s.$$

Let r^* be the largest integer r such that $h_r > 0$. Since $\Omega(0; \boldsymbol{\Psi}) = 0$, we know that $\sum_{d \le r^*} \Psi_d = 0$. Define a new degree distribution Ψ' by $\Psi'_d = \Psi_d \frac{\hat{\theta}_{\mathbf{h},\boldsymbol{\Psi}} - \epsilon}{\hat{\theta}_{\mathbf{h},\boldsymbol{\Psi}}}$ for $d > r^*$ and $\Psi'_d = \Delta$ for certain $d \le r^*$, where $\Delta > 0$ can be determined by the constraint $\sum_d \Psi'_d = 1$. Then we can check that Ψ' satisfies (6.5). □

Since $\hat{\theta}_{\mathbf{h}} = \max_{\boldsymbol{\Psi}} \hat{\theta}_{\mathbf{h},\boldsymbol{\Psi}}$, the above lemma imples that $\eta\hat{\theta}_{\mathbf{h}}$ is η-achievable by BATS codes using BP decoding. The converse of Lemma 6.1 is that "a rate larger than $\eta\hat{\theta}_{\mathbf{h}}$ is not η-achievable." Intuitively, for any $\epsilon > 0$, we cannot have a degree distribution $\boldsymbol{\Psi}$ such that

$$\Omega(x; \mathbf{h}, \boldsymbol{\Psi}) + (\hat{\theta}_{\mathbf{h}} + \epsilon) \ln(1 - x) \ge 0, \quad 0 \le x \le \eta.$$

Thus, with $\hat{\theta}_{\mathbf{h}} + \epsilon$ in place of θ in the expression of ρ_0 in (6.4), for any degree distribution we have $\rho_0(\tau) < 0$ for some $\tau \in [0, \eta(\hat{\theta}_{\mathbf{h}} + \epsilon)]$. By Theorem 5.7, for any degree distribution there exists K_0 such that when the number of input packets $K \ge K_0$, with probability approaching 1 the BATS code cannot recover ηK input packets. To prove this converse, however, we need a uniform bound K_0 for all degree distributions such that the second part of Theorem 5.7 holds, which is difficult to obtain. Instead, we will demonstrate that $\hat{\theta}_{\mathbf{h}}$ is close to $\mathbb{E}[\mathbf{h}]$, the capacity of the underlying linear operator channel (see Section 2.5).

Before analyzing the achievable rate, we determine the maximum degree D, which affects the encoding/decoding complexity. The next theorem shows that it is optimal to choose $D = \lceil M/\bar{\eta} \rceil - 1$, where $\bar{\eta} = 1 - \eta$, which implies that the average batch degree is $O(M)$.

Theorem 6.2 *Using $D > \lceil M/\bar{\eta} \rceil - 1$ does not give a better optimal value in (6.P1), where $\bar{\eta} = 1 - \eta$.*

Proof. Consider an integer Δ such that $\bar{\eta} \ge \frac{M}{\Delta+1}$. Let Ψ be a degree distribution with $\sum_{d>\Delta} \Psi_d > 0$. Construct a new degree distribution $\tilde{\Psi}$ with

$$\tilde{\Psi}_d = \begin{cases} \Psi_d & \text{if } d < \Delta, \\ \sum_{k \ge \Delta} \Psi_k & \text{if } d = \Delta, \\ 0 & \text{if } d > \Delta. \end{cases}$$

Write

$$\Omega(x;\tilde{\Psi}) - \Omega(x;\Psi) = \sum_{d=\Delta+1}^{\infty} \Psi_d \sum_{r=1}^{M} \hbar_r (\Delta \, \mathrm{I}_{\Delta-r,r}(x) - d \, \mathrm{I}_{d-r,r}(x)).$$

For $d \geq \Delta + 1$,

$$\frac{r-1}{d-r} \leq \frac{M-1}{d-M} < \frac{M}{\Delta - M + 1} \leq \frac{\bar{\eta}}{1-\bar{\eta}}.$$

So we can apply the properties of the incomplete beta function (Lemma B.2 in Appendix B) to show that, for any x with $0 < x \leq \eta$,

$$\begin{aligned} \frac{d \, \mathrm{I}_{d-r,r}(x)}{(d-1)\,\mathrm{I}_{d-1-r,r}(x)} &< \frac{d}{d-1}\left(1 - \frac{\bar{\eta}}{r}\right) \\ &\leq \frac{d}{d-1}\left(1 - \frac{\bar{\eta}}{M}\right) \\ &\leq \frac{\Delta+1}{\Delta}\left(1 - \frac{1}{\Delta+1}\right) \\ &= 1, \end{aligned}$$

which gives $\Omega(x;\tilde{\Psi}) > \Omega(x;\Psi)$ for $0 < x \leq \eta$. This means that using only degree distributions Ψ with $\sum_{d>\Delta} \Psi_d = 0$, we can obtain the same optimal value as using all degree distributions. Therefore, it is sufficient to take the maximum degree $D \leq \min_{\bar{\eta} \geq \frac{M}{\Delta+1}} \Delta = \lceil M/\bar{\eta} \rceil - 1$. □

To solve (6.P1) numerically, we can relax it as a linear programming by only considering x in a linearly sampled set of values between 0 and η. Let $x_i = \eta \frac{i}{N}$ for some integer N. We relax (6.P1) by considering only $x = x_i, i = 1, \ldots, N$, where N can be chosen to be 100 or even smaller.

For many cases, we can directly use the degree distribution Ψ obtained by solving (6.P1). But it is possible that $\Omega(0;\Psi) = 0$, so that the degree distribution Ψ does not guarantee that decoding can start. We can then modify Ψ as we do in the proof of Lemma 6.1 by increasing the probability masses Ψ_d, $d \leq M$ by a small amount to make sure that decoding can start.

Remark 6.3 To compare with the degree distribution optimization of LT/Raptor codes, we see that when $M = 1$, $\Omega(x;\mathbf{h},\boldsymbol{\Psi}) = \hbar_1 \sum_{d=1}^{D} d\,\Psi_d x^{d-1}$. The optimization (6.P1) becomes the optimization of Raptor codes when replacing $\hbar_1$ by h_1, which is the case when the generator matrices with all 1's are used. Moreover, we only need to consider the maximum degree $\lceil 1/\bar{\eta} \rceil - 1$ to recover η fraction of the input packets using LT codes.

6.2 ACHIEVABLE RATES

The first upper bound on the optimal value $\hat{\theta}$ of (6.P1) is given by the capacity of LOCs with receiver side channel state information. When the empirical rank distribution of the transfer matrices converging to $(h_0, \ldots, h_M)$, the capacity is $\sum_r r h_r$ packets per batch. The BP decoding algorithm recovers at least a fraction η of all the input packets with high probability. So asymptotically BATS codes under BP decoding can recover at least a fraction $\eta\hat{\theta}$ of the input packets. Thus, we have $\eta\hat{\theta} \le \sum_r r h_r$.

A tighter upper bound can be obtained by analyzing (6.P1) directly. We first rewrite (6.3) as

$$\Omega(x; \mathbf{h}, \mathbf{\Psi}) = \sum_{r=1}^{M} \hbar_r S_r(x; \mathbf{\Psi}), \tag{6.7}$$

where

$$S_r(x; \mathbf{\Psi}) = S_r(x) \triangleq \sum_{d=r+1}^{D} d\,\Psi_d\, \mathrm{I}_{d-r,r}(x) + \sum_{d=1}^{r} d\,\Psi_d. \tag{6.8}$$

This form of $\Omega(x; \mathbf{h}, \mathbf{\Psi})$ will be used in the subsequent proofs.

Theorem 6.4 *The optimal value $\hat{\theta}$ of (6.P1) satisfies*

$$\eta\hat{\theta} \le \sum_{r=1}^{M} r\hbar_r.$$

Remark 6.5 The theorem says that the achievable rates of BP decoding are upper bounded by $\sum_{r=1}^{M} r\hbar_r$. We note, however, that $\hat{\theta}$ can be larger than $\sum_{r=1}^{M} r\hbar_r$.

Proof. Fix a degree distribution that achieves the optimal value of (6.P1). Using (B.2) in Appendix B, we have

$$\begin{aligned}
\int_0^1 S_r(x)\,\mathrm{d}\,x &= \sum_{d=r+1}^{D} d\,\Psi_d \int_0^1 \mathrm{I}_{d-r,r}(x)\,\mathrm{d}\,x + \sum_{d=1}^{r} d\,\Psi_d \\
&= \sum_{d=r+1}^{D} r\,\Psi_d + \sum_{d=1}^{r} d\,\Psi_d \\
&\le r \sum_{d=1}^{D} \Psi_d \\
&= r.
\end{aligned}$$

Hence,

$$\int_0^1 \Omega(x)\,\mathrm{d}\,x = \int_0^1 \sum_{r=1}^{M} \hbar_r S_r(x)\,\mathrm{d}\,x \le \sum_{r=1}^{M} r\hbar_r. \tag{6.9}$$

Since $\Omega(x)$ is an increasing function,

$$\int_\eta^1 \Omega(x)\,\mathrm{d}\,x \ge \bar{\eta}\Omega(\eta) \ge -\bar{\eta}\hat{\theta}\ln\bar{\eta}. \tag{6.10}$$

Since $\Omega(x) + \hat{\theta}\ln(1-x) \ge 0$ for $0 < x \le \eta$,

$$\begin{aligned}&\int_0^\eta \Omega(x)\,\mathrm{d}\,x - \hat{\theta}(\bar{\eta}\ln\bar{\eta} + \eta)\\ &\quad= \int_0^\eta \Omega(x)\,\mathrm{d}\,x + \hat{\theta}\int_0^\eta \ln(1-x)\,\mathrm{d}\,x \ge 0.\end{aligned} \tag{6.11}$$

Therefore, by (6.9)–(6.11), we have

$$\begin{aligned}\sum_{r=1}^{M} r\hbar_r &\ge \int_0^1 \Omega(x)\,\mathrm{d}\,x\\ &= \int_0^\eta \Omega(x)\,\mathrm{d}\,x + \int_\eta^1 \Omega(x)\,\mathrm{d}\,x\\ &\ge \hat{\theta}(\bar{\eta}\ln\bar{\eta} + \eta) - \bar{\eta}\hat{\theta}\ln\bar{\eta}\\ &= \hat{\theta}\eta.\end{aligned}$$

The proof is completed. □

Since $\sum_{k=r}^{M} \hbar_k = \sum_{i=r}^{M} h_i\zeta_r^i \le \sum_{k=r}^{M} h_k$, where the last inequality follows from $\zeta_r^i < 1$, we have

$$\begin{aligned}\sum_r r\hbar_r &= \sum_{r=1}^{M}\sum_{k=r}^{M} \hbar_k\\ &\le \sum_{r=1}^{M}\sum_{k=r}^{M} h_k\\ &= \sum_r r h_r.\end{aligned}$$

Therefore, Theorem 6.4 gives an improved upper bound on $\hat{\theta}$ compared with $\sum_r r h_r$. When $q \to \infty$, $\sum_r r\hbar_r \to \sum_r r h_r$. For finite fields with $q \ge 16$, $\sum_r r\hbar_r$ and $\sum_r r h_r$ are very close, but for smaller finite fields, the difference can be significant.

We prove for a special case and demonstrate by simulation for general cases that the optimal value $\hat{\theta}$ of (6.P1) is very close to $\sum_r r\hbar_r$.

Theorem 6.6 *For $D = \lceil M/\bar{\eta}\rceil - 1$, the optimal value $\hat{\theta}$ of (6.P1) satisfies*

$$\hat{\theta} \geq \max_{r=1,2,\cdots,M} r \sum_{i=r}^{M} \hbar_i.$$

Proof. Define a degree distribution $\mathbf{\Psi}^r$ by

$$\Psi_d^r = \begin{cases} 0 & \text{if } d \leq r, \\ \frac{r}{d(d-1)} & \text{if } d = r+1, \cdots, D-1, \\ \frac{r}{D-1} & \text{if } d = D. \end{cases} \tag{6.12}$$

Recall the definition of $S_r(x;\mathbf{\Psi})$ in (6.8). For $M \geq r' \geq r$, we will show that

$$S_{r'}(x;\mathbf{\Psi}^r) + r\ln(1-x) > 0, \quad 0 \leq x \leq \eta. \tag{6.13}$$

By Lemma B.3 in Appendix B,

$$-r\ln(1-x) = r\sum_{d=r'+1}^{\infty} \frac{1}{d-1} \mathrm{I}_{d-r',r'}(x).$$

By (6.8) and (6.12),

$$\begin{aligned} S_{r'}(x;\mathbf{\Psi}^r) + r\ln(1-x) &\geq \sum_{d=r'+1}^{D} d\,\Psi_d^r\, \mathrm{I}_{d-r',r'}(x) - r\sum_{d=r'+1}^{\infty} \frac{1}{d-1} \mathrm{I}_{d-r',r'}(x) \\ &= r\frac{D}{D-1} \mathrm{I}_{D-r',r'}(x) - r\sum_{d=D}^{\infty} \frac{1}{d-1} \mathrm{I}_{d-r',r'}(x) \\ &= r\,\mathrm{I}_{D-r',r'}(x) - r\sum_{d=D+1}^{\infty} \frac{1}{d-1} \mathrm{I}_{d-r',r'}(x). \end{aligned}$$

The expression above is strictly positive for $x \in [0,\eta]$ if and only if

$$\sum_{d=D+1}^{\infty} \frac{1}{d-1} \frac{\mathrm{I}_{d-r',r'}(x)}{\mathrm{I}_{D-r',r'}(x)} < 1 \quad \text{for } x \in [0,\eta]. \tag{6.14}$$

By Lemma B.1 in Appendix B, $\frac{\mathrm{I}_{d-r',r'}(x)}{\mathrm{I}_{D-r',r'}(x)}$ is monotonically increasing, so we only need to prove the above inequality for $x = \eta$. By Lemma B.2 in Appendix B, $\frac{\mathrm{I}_{d-r',r'}(\eta)}{\mathrm{I}_{D-r',r'}(\eta)} < (1-\frac{\bar{\eta}}{M})^{d-D}$. Therefore,

$$\begin{aligned}\sum_{d=D+1}^{\infty} \frac{1}{d-1}\frac{\mathrm{I}_{d-r',r'}(x)}{\mathrm{I}_{D-r',r'}(x)} &\leq \frac{1}{D}\sum_{d=D+1}^{\infty}\frac{\mathrm{I}_{d-r',r'}(\eta)}{\mathrm{I}_{D-r',r'}(\eta)} \\ &< \frac{1}{D}\sum_{d=D+1}^{\infty}\left(1-\frac{\bar{\eta}}{M}\right)^{d-D} \\ &= \frac{M-\bar{\eta}}{D\bar{\eta}} \\ &\leq 1,\end{aligned}$$

where the last inequality follows from $D = \lceil M/\bar{\eta}\rceil - 1$. So we have established (6.14) and hence (6.13).

Finally, by (6.7) and (6.13), we have for $0 \leq x \leq \eta$,

$$\begin{aligned}\Omega(x;\boldsymbol{\Psi}^r) &\geq \sum_{r'\geq r}\hbar_{r'}S_{r'}(x;\Psi^r) \\ &> -\ln(1-x)r\sum_{r'\geq r}\hbar_{r'},\end{aligned}$$

or

$$\Omega(x;\boldsymbol{\Psi}^r) + \left(r\sum_{r'\geq r}\hbar_{r'}\right)\ln(1-x) > 0.$$

Hence, we conclude that $\hat{\theta} \geq r\sum_{r'\geq r}\hbar_{r'}$. The proof is completed by considering all $r = 1, 2, \cdots, M$. □

Though in general the lower bound in Theorem 6.6 is not tight, we can show for a special case that it converges asymptotically to the upper bound in Theorem 6.4. Consider a rank distribution $(h_0, h_1, \ldots, h_M)$ with $h_\kappa = 1$ for some $1 \leq \kappa \leq M$. Theorem 6.6 implies that $\hat{\theta} \geq \kappa\hbar_\kappa$. On the other hand, Theorem 6.4 says that $\eta\hat{\theta} \leq \sum_r r\hbar_r = \kappa\hbar_\kappa + \sum_{r<\kappa} r\hbar_r$. Note that η can be arbitrarily close to 1, and $\sum_{r<\kappa} r\hbar_r \to 0$ and $\hbar_\kappa \to h_\kappa$ when the field size goes to infinity. Thus, both the upper bound in Theorem 6.4 and the lower bound in Theorem 6.6 converge to κh_κ, the capacity of the LOC with empirical rank distribution converging to h.

We can compute the achievable rates of BATS codes numerically by solving (6.P1). Set $M = 16$ and $q = 2^8$. Totally 4×10^4 rank distributions are tested.[1] For each rank distribution

[1]A rank distribution is randomly generated as follows. First, select $x_1, x_2, \ldots, x_{M-1}$ independently and uniformly at random in $[0, 1]$. Next, sort $\{x_i\}$ so that $x_1 \leq x_2 \leq \cdots \leq x_{M-1}$. Then, the rank distribution is given by $h_0 = 0$, and for $1 \leq r \leq M$, $h_r = x_r - x_{r-1}$, where $x_0 = 1$ and $x_M = 1$. This gives an almost uniform sampling among all the rank

h we solve (6.P1) for $\bar{\eta} = 0.98, 0.99$, and 0.995. The empirical distributions of $\tilde{\theta} \triangleq \eta\hat{\theta}/\sum_r r\hbar_r$ are shown in Figure 6.1. By Theorem 6.4, $\tilde{\theta} \leq 1$. The results show that when $\eta = 0.995$, for more than 99.1% of the rank distributions, $\tilde{\theta}$ is larger than 0.96; for all the rank distributions the smallest $\tilde{\theta}$ is 0.9057. The figures in Figure 6.1 clearly show the trend that when η becomes smaller, $\hat{\theta}$ becomes larger for the same rank distribution. Note that for these rank distributions, the ratio $\sum_r r\hbar_r/\sum_r rh_r$ are all larger than 0.999, and therefore, the achievable rate of BATS code (i.e., $\eta\hat{\theta}$) is very close to the capacity $\sum_r rh_r$ for all the rank distributions evaluated.

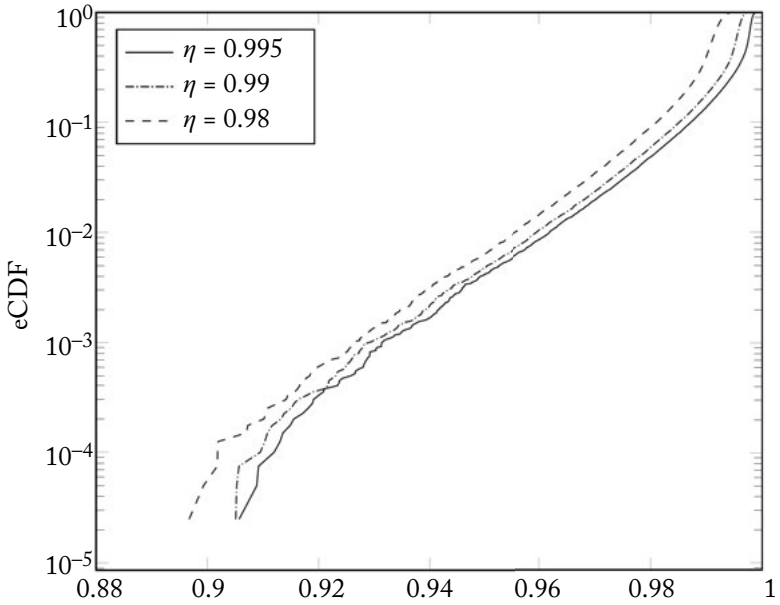


Figure 6.1: The empirical cumulative distribution function (eCDF) of $\tilde{\theta} = \eta\hat{\theta}/\sum_r r\hbar_r$ for 4×10^4 rank distributions. Here $q = 2^8$ and $M = 16$.

Remark 6.7 For LT codes, we can apply Theorem 6.6 with $M = 1$ and $\hbar_i = h_i$, and obtain that $\hat{\theta} \geq h_1$. Therefore, the achievable rate of LT code is at least ηh_1, which tends to h_1 as $\eta \to 1$. In other words, LT codes achieve the capacity of a channel with packet loss rate $1 - h_1$.

distributions with $\sum_{i=1}^{M} h_i = 1$ according to [75]. The reason that we choose $h_0 = 0$ is as follows. For a rank distribution $(h_0, \ldots, h_M)$ with $h_0 > 0$, we obtain a new rank distribution $(h'_0 = 0, h'_i = h_i/(1 - h_0) : i = 1, \ldots, M)$. Optimization (6.P1) is equivalent for these two rank distributions except that the objective function is scaled by $1 - h_0$. Thus, the values of $\tilde{\theta} = \eta\hat{\theta}/\sum_r r\hbar_r$ for both $(h_0, \ldots, h_M)$ and $(h'_0, \ldots, h'_M)$ are the same.

6.3 OPTIMIZATIONS FOR MULTIPLE RANK DISTRIBUTIONS

In the previous part of this chapter, we consider how to find an optimal degree distribution for a single rank distribution. For many scenarios, however, we need a degree distribution that is good for multiple rank distributions. In a general multicast problem, the rank distributions observed by the destination nodes can be different. Even for a single destination node, the empirical rank distribution may not always converge to the same value.

Let $\mathcal{H}$ be a set of rank distributions. Note that for a rank distribution $(h_1, h_2, \ldots, h_M)$ in $\mathcal{H}$, it is implied that $h_0 = 1 - \sum_{i=1}^{M} h_i$. Consider a degree distribution $\mathbf{\Psi}$, $\mathbf{h} \in \mathcal{H}$, and $\theta_{\mathbf{h}}$ satisfying the following set of constraints:

$$\Omega(x; \mathbf{h}, \mathbf{\Psi}) + \theta_{\mathbf{h}} \ln(1-x) \geq 0, \ \forall x \in [0, \eta], \ \forall \mathbf{h} \in \mathcal{H}. \tag{6.15}$$

Then for a destination node with the empirical rank distribution converges in probability to $\mathbf{h} \in \mathcal{H}$, rate $\eta\theta_{\mathbf{h}}$ is η-achievable by the BATS code with degree distribution $\mathbf{\Psi}$.

To illustrate the discussion, we extend the three-node network in Figure 1.1 with two more destination nodes as shown in Figure 6.2. In this network, node R transmits the same packets on its three outgoing links, but these links have different loss rates. Fixing $M = 16$, $q = 256$ and a certain inner code at node R (RLNC with $q_m = 256$ and M recoded packets), we obtain the rank distributions $\mathbf{h}^i$ for node Dst_i, $i = 1, 2, 3$ in Table 6.1. For this example, the maximum η-achievable rates are evaluated and listed in Table 6.2. From the rows labeled by $\mathbb{E}[\mathbf{h}]$, $\mathbf{\Psi}^1$, $\mathbf{\Psi}^2$, and $\mathbf{\Psi}^3$, we observe that that the degree distribution optimized for one rank distribution may not give a good performance for the other rank distributions. In particular, the degree distributions optimized for destination nodes Dst_1 and Dst_2, namely $\mathbf{\Psi}^1$ and $\mathbf{\Psi}^2$, respectively, give a poor performance for destination node Dst_3.

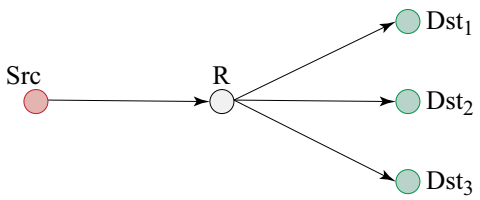


Figure 6.2: In this network, node Src is the source node. Node $\mathrm{Dst}_1, \mathrm{Dst}_2$, and Dst_3 are the destination nodes. Node R is the intermediate node that does not demand the file. All links are capable of transmitting one packet per use. The link $(\mathrm{Src}, \mathrm{R})$ has packet loss rate 0.2. The links $(\mathrm{R}, \mathrm{Dst}_i), i = 1, 2, 3$ have packet loss rates 0.1, 0.2, and 0.3, respectively.

Table 6.1: The rank distributions for the three destination nodes in Figure 6.2

Rank	$\mathbf{h}^1$	$\mathbf{h}^2$	$\mathbf{h}^3$
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0.0002
5	0	0.0001	0.0013
6	0.0002	0.0004	0.0058
7	0.0012	0.0025	0.0197
8	0.0056	0.011	0.0537
9	0.0201	0.0387	0.1165
10	0.0576	0.1041	0.1969
11	0.1306	0.2062	0.2468
12	0.2276	0.2795	0.2121
13	0.2796	0.2339	0.112
14	0.2052	0.1039	0.0312
15	0.0671	0.019	0.0036
16	0.0052	0.0008	0.0001

Table 6.2: The achievable rates for different pairs of rank distributions and degree distributions. For each rank distribution in the first row and each degree distribution in the first column, we evaluate the maximum 0.99-achievable rate in the table. For $i = 1, 2, 3$, $\mathbf{\Psi}^i$ is obtained by solving (6.P1) with $\mathbf{h}^i$ in place of $\mathbf{h}$. $\mathbf{\Psi}^3$ can also be obtained by solving (6.P2) with $\{\mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3\}$ in place of $\mathcal{H}$. $\mathbf{\Psi}^{\text{fair}}$ is obtain by solving (6.P3) with $\{\mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3\}$ in place of $\mathcal{H}$.

	$\mathbf{h}^1$	$\mathbf{h}^2$	$\mathbf{h}^3$
$\mathbb{E}[\mathrm{h}]$	12.58	11.91	10.84
Ψ^1	12.55	6.10	1.77
Ψ^2	11.96	11.89	4.79
Ψ^3	10.99	10.95	10.81
Ψ^{fair}	11.94	11.35	10.28

6.3.1 OPTIMIZATION PROBLEMS

There are different criteria to optimize the degree distribution for a set of rank distributions. Here we discuss a few examples.

Multicast

One performance metric of interest is the multicast rate, which is a rate achievable by all the rank distributions using the same degree distribution. We can find the maximum multicast rate for a set of rank distributions $\mathcal{H}$ by solving the following optimization problem:

$$\begin{aligned} \max_{\boldsymbol{\Psi},\theta} \ & \theta \\ \text{s.t.} \ & \Omega(x;\mathbf{h},\boldsymbol{\Psi}) + \theta \ln(1-x) \geq 0, \forall x \in [0,\eta], \ \forall \mathbf{h} \in \mathcal{H}, \\ & \sum_{d} \Psi_d = 1 \ \text{ and } \ \Psi_d \geq 0, \ d = 1, \cdots, D. \end{aligned} \tag{6.P2}$$

Denote by $\hat{\theta}_{\mathcal{H}}$ the maximum of (6.P2) for $\mathcal{H}$. Then

$$\hat{\theta}_{\mathcal{H}} = \max_{\boldsymbol{\Psi}} \min_{\mathbf{h}\in\mathcal{H}} \hat{\theta}_{\mathbf{h},\boldsymbol{\Psi}} \leq \min_{\mathbf{h}\in\mathcal{H}} \hat{\theta}_{\mathbf{h}}. \tag{6.16}$$

For the example that $\mathcal{H} = \{\mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3\}$, the optimal degree distribution of (6.P2) is exactly $\boldsymbol{\Psi}^3$ and $\eta \min_{\mathbf{h}\in\mathcal{H}} \hat{\theta}_{\mathbf{h}} = 10.81 = \eta\theta_{\mathbf{h}^3}$. Since nodes t_1 and t_2 can emulate the packet loss rate of node t_3, the multicast rate $\eta\hat{\theta}_{\mathbf{h}_3}$ is achievable. So in this case, BATS codes can achieve a multicast rate equal to $\eta \min_{\mathbf{h}\in\mathcal{H}} \hat{\theta}_{\mathbf{h}}$. In general, however, $\hat{\theta}_{\mathcal{H}}$ may not be very close to $\min_{\mathbf{h}\in\mathcal{H}} \theta_{\mathbf{h}}$.

Fair Multicast

The degree distribution obtained using (6.P2) may not be fair for all the destination nodes. In the previous example of three destination nodes, for the degree distribution $\boldsymbol{\Psi}^3$ optimized using (6.P2), nodes Dst_1 and Dst_2 do not achieve a rate much higher than node Dst_3 though they have much lower packet loss rates than node Dst_3 (see Table 6.2). For a single rank distribution $\mathbf{h}$, we know that the achievable rate of BATS codes is upper bounded by and is very close to $\mathbb{E}[\mathbf{h}]$. To ensure that a destination node with a lower packet loss rate has a higher multicast rate, we can find the percentage of $\mathbb{E}[\mathbf{h}]$ that is achievable for all the rank distributions $\mathbf{h}$ in $\mathcal{H}$ using the following optimization:

$$\begin{aligned} \max_{\boldsymbol{\Psi},\alpha} \ & \alpha \\ \text{s.t.} \ & \Omega(x;\mathbf{h},\boldsymbol{\Psi}) + \alpha\, \mathbb{E}[\mathbf{h}] \ln(1-x) \geq 0, \forall x \in [0,\eta], \ \forall \mathbf{h} \in \mathcal{H}, \\ & \sum_{d} \Psi_d = 1 \ \text{ and } \ \Psi_d \geq 0, \ d = 1, \cdots, D. \end{aligned} \tag{6.P3}$$

Denote by $\hat{\alpha}_{\mathcal{H}}$ the maximum of (6.P3). We know that for each rank distribution $\mathbf{h}$ in $\mathcal{H}$, $\eta\hat{\alpha}_{\mathcal{H}}\, \mathbb{E}[\mathbf{h}]$ is an achievable rate, and hence $\eta\hat{\alpha}_{\mathcal{H}}$ is the percentage of $\mathbb{E}[\mathbf{h}]$ that is achievable for all the rank distributions in $\mathcal{H}$. Then by letting

$$\hat{\alpha}_{\mathbf{h},\boldsymbol{\Psi}} = \frac{\hat{\theta}_{\mathbf{h},\boldsymbol{\Psi}}}{\mathbb{E}[\mathbf{h}]} \tag{6.17}$$

and

$$\hat{\alpha}_{\mathbf{h}} = \frac{\hat{\theta}_{\mathbf{h}}}{\mathbb{E}[\mathbf{h}]}, \tag{6.18}$$

we have

$$\hat{\alpha}_{\mathcal{H}} = \max_{\mathbf{\Psi}} \min_{\mathbf{h}\in\mathcal{H}} \hat{\alpha}_{\mathbf{h},\mathbf{\Psi}} \le \min_{\mathbf{h}\in\mathcal{H}} \hat{\alpha}_{\mathbf{h}}. \tag{6.19}$$

When $\mathcal{H} = \{\mathbf{h}^1, \mathbf{h}^2, \mathbf{h}^3\}$ and $\eta = 0.99$, the percentage is 94.9 (the optimal value of (6.P3) multiplied by 100η). The performance of the optimal degree distribution of (6.P3) is shown in the last row of Table 6.2. A BATS codes with this degree distribution can achieve 95.0, 95.3, and 94.8 percentage of $\mathbb{E}[\mathbf{h}^i]$ for $i = 1, 2, 3$, respectively.

We now build a connection between these two optimizations by showing that the optimization (6.P3) can be equivalently converted to the optimization (6.P2). Define for $0 < \mu \le M$,

$$\mathcal{D}_\mu = \left\{\mathbf{h} \in \mathcal{P}^M : \mathbb{E}[\mathbf{h}] = \mu\right\}. \tag{6.20}$$

Lemma 6.8 *Optimization* (6.P3) *for* $\mathcal{H} \subset \mathcal{D}_\mu$, $0 < \mu \le M$, *has the same optimal degree distribution as optimization* (6.P2) *for* $\mathcal{H}$, *and* $\hat{\alpha}_{\mathcal{H}} = \hat{\theta}_{\mathcal{H}}/\mu$.

Proof. Since for any $\mathbf{h} \in \mathcal{H}$, $\mathbb{E}[\mathbf{h}] = \mu$, the lemma follows directly from (6.19). □

Using different objective functions and constraints, other optimization problems can be formulated to optimize a degree distribution for a set of rank distributions. For example, we can optimize the average rate and average completion time of all the destination nodes.

6.3.2 SIMPLIFICATIONS

When $\mathcal{H}$ has a small cardinality, the optimizations (6.P2) and (6.P3) are easy to solve. We study how to simplify the optimizations when $|\mathcal{H}|$ is large or infinite. Let $\mathbb{R}^+$ be the set of non-negative real numbers, and let

$$\mathcal{P}_M = \left\{(h_1, \ldots, h_M) \in (\mathbb{R}^+)^M : \sum_{i=1}^{M} h_i \le 1\right\},$$

i.e., $\mathcal{P}_M$ be the set of all rank distributions with the maximum rank equal to M.

Linearity and Dominance

We first prove some properties about the degree distribution optimization of BATS codes.

Theorem 6.9 *For any* $\beta, \beta' \ge 0$, $\mathbf{h}, \mathbf{h}' \in (\mathbb{R}^+)^M$, *and any degree distribution* $\mathbf{\Psi}$,

(1) $\hat{\theta}_{\beta\mathbf{h}+\beta'\mathbf{h}',\mathbf{\Psi}} \ge \beta\hat{\theta}_{\mathbf{h},\mathbf{\Psi}} + \beta'\hat{\theta}_{\mathbf{h}',\mathbf{\Psi}}$;

(2) $\hat{\theta}_{\beta\mathbf{h}+\beta'\mathbf{h}',\mathbf{\Psi}} \geq \min\{\hat{\theta}_{\mathbf{h},\mathbf{\Psi}}, \hat{\theta}_{\mathbf{h}',\mathbf{\Psi}}\}$ *when* $\beta + \beta' = 1$; *and*

(3) $\hat{\alpha}_{\beta\mathbf{h}+\beta'\mathbf{h}',\mathbf{\Psi}} \geq \min\{\hat{\alpha}_{\mathbf{h},\mathbf{\Psi}}, \hat{\alpha}_{\mathbf{h}',\mathbf{\Psi}}\}$

Proof. For any $x \in [0, \eta]$, $\Omega(x; \mathbf{h}, \mathbf{\Psi}) + \theta \ln(1-x)$ is a linear function of $(\theta, \mathbf{h})$. Therefore, for $x \in [0, 1]$, if

$$\Omega(x; \mathbf{h}, \mathbf{\Psi}) + \theta \ln(1-x) \geq 0,$$

and

$$\Omega(x; \mathbf{h}', \mathbf{\Psi}) + \theta' \ln(1-x) \geq 0,$$

then for any $\beta, \beta' \geq 0$, we have

$$\Omega(x; \beta\mathbf{h} + \beta'\mathbf{h}', \mathbf{\Psi}) + (\beta\theta + \beta'\theta') \ln(1-x) \geq 0,$$

which implies (1). (2) is a consequence of (1) with $\beta + \beta' = 1$. Following (6.17), (6.18), and (1), we have

$$\begin{aligned}
\hat{\alpha}_{\beta\mathbf{h}+\beta'\mathbf{h}',\mathbf{\Psi}} \, \mathbb{E}[\beta\mathbf{h} + \beta'\mathbf{h}'] &= \hat{\theta}_{\beta\mathbf{h}+\beta'\mathbf{h}',\mathbf{\Psi}} \\
&\geq \beta\hat{\theta}_{\mathbf{h},\mathbf{\Psi}} + \beta'\hat{\theta}_{\mathbf{h}',\mathbf{\Psi}} \\
&= \beta\hat{\alpha}_{\mathbf{h},\mathbf{\Psi}} \, \mathbb{E}[\mathbf{h}] + \beta'\hat{\alpha}_{\mathbf{h}',\mathbf{\Psi}} \, \mathbb{E}[\mathbf{h}'] \\
&\geq \min\{\hat{\theta}_{\mathbf{h},\mathbf{\Psi}}, \hat{\theta}_{\mathbf{h}',\mathbf{\Psi}}\} \left(\beta \, \mathbb{E}[\mathbf{h}] + \beta' \, \mathbb{E}[\mathbf{h}']\right).
\end{aligned}$$

Since $\mathbb{E}[\beta\mathbf{h} + \beta'\mathbf{h}'] = \beta \, \mathbb{E}[\mathbf{h}] + \beta' \, \mathbb{E}[\mathbf{h}']$, we obtain (3). The theorem is proved. □

Consider the case that $\mathcal{H}$ in the optimization (6.P2) is a ray generated by a rank distribution $\mathbf{h} = (h_1, \ldots, h_M)$ with $\sum_{i=1}^{M} h_i > 0$ as

$$\mathcal{H} = \langle \mathbf{h} \rangle \triangleq \left\{ \alpha\mathbf{h} : \alpha \geq 0, \alpha \sum_{i=1}^{M} h_i \leq 1 \right\}.$$

Then it follows from 1) in Theorem 6.9 that there exists a degree distribution which is optimal for all the rank distributions in $\langle \mathbf{h} \rangle$.

Remark 6.10 Following Remark 6.7, h_1 is achievable by LT codes. When $M = 1$, $\langle h_1 \rangle$ for any $0 < h_1 \leq 1$ includes all possible rank distributions, and hence there exists a degree distribution that is optimal for all the rank distributions for $M = 1$. In other words, there exists a universally optimal degree distribution when $M = 1$.

The partial order (dominance relation) defined on rank distributions in Section 4.1.2 can be extended for vectors in $(\mathbb{R}^+)^M$. For $\mathbf{h} = (h_1, \ldots, h_M)$ and $\mathbf{h}' = (h'_1, \ldots, h'_M)$ in $(\mathbb{R}^+)^M$, we say that $\mathbf{h}$ dominates $\mathbf{h}'$, denoted by $\mathbf{h} \succcurlyeq \mathbf{h}'$, if $\sum_{i \geq k} h_i \geq \sum_{i \geq k} h'_i$ for all $k = 1, \ldots, M$.

Lemma 6.11 *For two vectors* $\mathbf{h} = (h_1, \ldots, h_M)$ *and* $\mathbf{h}' = (h'_1, \ldots, h'_M)$ *in* $(\mathbb{R}^+)^M$ *with* $\mathbf{h} \succcurlyeq \mathbf{h}'$, *and* $0 \leq v_1 \leq \ldots \leq v_M$, *we have* $\sum_{i=k}^{M} v_i h_i \geq \sum_{i=k}^{M} v_i h'_i$, *where the equality holds when* $\mathbf{h} = \mathbf{h}'$.

Proof. Fix $1 \le k \le M$. Let $v'_k = v_k$ and $v'_i = v_i - v_{i-1}$ for $i = k+1, \dots, M$. Since $v'_i \ge 0$,

$$\sum_{i=k}^{M} v_i h_i = \sum_{i=k}^{M} v'_i \sum_{j \ge i} h_j \ge \sum_{i=k}^{M} v'_i \sum_{j \ge i} h'_j = \sum_{i=k}^{M} v_i h'_i,$$

where the inequality is an equality if and only if $\mathbf{h} = \mathbf{h}'$. □

Now we study another property of $\Omega(x; \mathbf{h}, \boldsymbol{\Psi})$ related to the dominance relation.

Lemma 6.12 *Fix $x \in [0, 1)$, and two rank distributions $\mathbf{h} \succcurlyeq \mathbf{h}'$. If*

$$\Omega(x; \mathbf{h}', \boldsymbol{\Psi}) + \theta \ln(1 - x) \ge 0,$$

then

$$\Omega(x; \mathbf{h}, \boldsymbol{\Psi}) + \theta \ln(1 - x) \ge 0.$$

Proof. Since $\zeta_k^M \ge \zeta_k^{M-1} \ge \cdots \ge \zeta_k^k \ge 0$, we have for $k = 1, \dots, M$,

$$\begin{aligned} \sum_{i \ge k} \hbar_i(\mathbf{h}) &= \sum_{i \ge k} \zeta_s^k h_k \\ &\ge \sum_{i \ge k} \zeta_s^k h'_k \\ &= \sum_{i \ge k} \hbar_i(\mathbf{h}'), \end{aligned}$$

where the inequality follows from Lemma 6.11 and the two equalities follow from (2.11), i.e,

$$(\hbar_1(\mathbf{h}), \dots, \hbar_M(\mathbf{h})) \succcurlyeq (\hbar_1(\mathbf{h}'), \dots, \hbar_M(\mathbf{h}')).$$

Since

$$S_{r+1}(x) - S_r(x) = \sum_{d=r+1}^{D} d\,\Psi_d \binom{d-1}{r} x^{d-r-1}(1-x)^r \ge 0,$$

again by Lemma 6.11, we have

$$\begin{aligned} \Omega(x; \mathbf{h}, \boldsymbol{\Psi}) &= \sum_{r=1}^{M} \hbar_r(\mathbf{h}) S_r(x) \\ &\ge \sum_{r=1}^{M} \hbar_r(\mathbf{h}') S_r(x) \\ &= \Omega(x; \mathbf{h}', \boldsymbol{\Psi}), \end{aligned}$$

where the inequality follows from Lemma 6.11 and the two equalities follow from (6.7). □

Lemma 6.12 says that for two rank distributions $\mathbf{h} \succcurlyeq \mathbf{h}'$, not only $\hat{\theta}_{\mathbf{h}} \geq \hat{\theta}_{\mathbf{h}'}$ (the achievable rate of BATS codes for $\mathbf{h}$ is higher than that for $\mathbf{h}'$), but also for any degree distribution $\boldsymbol{\Psi}$, $\hat{\theta}_{\mathbf{h},\boldsymbol{\Psi}} \geq \hat{\theta}_{\mathbf{h}',\boldsymbol{\Psi}}$. For example, the three rank distributions for Figure 6.2 satisfies $\mathbf{h}^1 \succcurlyeq \mathbf{h}^2 \succcurlyeq \mathbf{h}^3$. We can see from Table 6.1 that for each degree distribution evaluated, the achievable rate for $\mathbf{h}^1$ (resp. $\mathbf{h}^2$) is larger than that of $\mathbf{h}^2$ (resp. $\mathbf{h}^3$).

Theorem 6.13 *Consider the optimization* (6.P2) *for a set of rank distributions $\mathcal{H}$. Let $\mathcal{B}$ be a subset of $\mathcal{H}$ such that for any $\mathbf{h} \in \mathcal{H}$ there exists $\mathbf{h}' \in \mathcal{B}$ such that $\mathbf{h} \succcurlyeq \mathbf{h}'$. Then, the optimization* (6.P2) *is equivalent if we replace $\mathcal{H}$ by $\mathcal{B}$.*

Proof. Let $\boldsymbol{\Psi}$ be a degree distribution that achieves the optimal value $\hat{\theta}_{\mathcal{B}}$ of (6.P2) with $\mathcal{B}$ in place of $\mathcal{H}$. Since $\mathcal{B} \subset \mathcal{H}$, we have $\hat{\theta}_{\mathcal{B}} \geq \hat{\theta}_{\mathcal{H}}$. For any $\mathbf{h} \in \mathcal{H} \setminus \mathcal{B}$, find $\mathbf{h}' \in \mathcal{B}$ such that $\mathbf{h} \succcurlyeq \mathbf{h}'$. By Lemma 6.12, $\hat{\theta}_{\mathbf{h},\boldsymbol{\Psi}} \geq \hat{\theta}_{\mathbf{h}',\boldsymbol{\Psi}} \geq \hat{\theta}_{\mathcal{B}}$, which implies $\hat{\theta}_{\mathcal{H}} \geq \hat{\theta}_{\mathcal{B}}$. Hence, $\hat{\theta}_{\mathcal{B}} = \hat{\theta}_{\mathcal{H}}$, which is achieved by $\boldsymbol{\Psi}$. □

This theorem provides a general approach to simplifying (6.P2) by replacing $\mathcal{H}$ with a minimal subset $\mathcal{H}^*$ where for any $\mathbf{h} \in \mathcal{H}$ there exists $\mathbf{h}' \in \mathcal{H}^*$ such that $\mathbf{h} \succcurlyeq \mathbf{h}'$. Since $\mathcal{H}^*$ is minimal, for any $\mathbf{h} \neq \mathbf{h}' \in \mathcal{H}^*$, neither $\mathbf{h} \succcurlyeq \mathbf{h}'$ nor $\mathbf{h}' \succcurlyeq \mathbf{h}$.

Convex Hull and Cone

Let $\text{hull}(\mathcal{A})$ be the convex hull of $\mathcal{A} \subset (\mathbb{R}^+)^{\boldsymbol{M}}$, i.e., $\text{hull}(\mathcal{A})$ is the set of all convex combinations of the vectors in $\mathcal{A}$.

Theorem 6.14 *Consider the optimization* (6.P2) *for a set of rank distributions $\mathcal{H}$. Let $\mathcal{A}$ be a set of rank distributions such that* $\text{hull}(\mathcal{A}) = \text{hull}(\mathcal{H})$. *Then, the optimization* (6.P2) *is equivalent if we replace $\mathcal{H}$ by $\mathcal{A}$.*

Proof. Let $\boldsymbol{\Psi}$ be a degree distribution that achieves the optimal value $\hat{\theta}_{\mathcal{H}}$ of (6.P2) for the set of rank distributions $\mathcal{H}$. Consider any $\mathbf{h} \in \text{hull}(\mathcal{H})$. Since $\mathbf{h}$ is a convex combination of the vectors in $\mathcal{H}$, by (2) in Theorem 6.9 and (6.16), $\hat{\theta}_{\mathbf{h},\boldsymbol{\Psi}} \geq \hat{\theta}_{\mathcal{H}}$, and hence $\hat{\theta}_{\text{hull}(\mathcal{H})} \geq \hat{\theta}_{\mathcal{H}}$. On the other hand, $\hat{\theta}_{\text{hull}(\mathcal{H})} \leq \hat{\theta}_{\mathcal{H}}$ since $\mathcal{H} \subset \text{hull}(\mathcal{H})$. Therefore, $\hat{\theta}_{\text{hull}(\mathcal{H})} = \hat{\theta}_{\mathcal{H}}$, which can be achieved by $\boldsymbol{\Psi}$.

Applying the same argument for $\mathcal{A}$ in place of $\mathcal{H}$, we obtain $\hat{\theta}_{\mathcal{A}} = \hat{\theta}_{\text{hull}(\mathcal{A})} = \hat{\theta}_{\text{hull}(\mathcal{H})} = \hat{\theta}_{\mathcal{H}}$, which can be achieved by $\boldsymbol{\Psi}$. □

This theorem provides further simplifications of (6.P2), i.e., instead of replacing $\mathcal{H}$ by a minimal subset $\mathcal{H}^*$, we can further replace $\mathcal{H}$ by the vertices of $\text{hull}(\mathcal{H}^*)$ if it is a polytope. When $\text{hull}(\mathcal{H}^*)$ is not a polytope, we may approximate it by a polytope to simplify the problem.

For $\mathcal{A} \subset (\mathbb{R}^+)^{\boldsymbol{M}}$, let $\text{cone}(\mathcal{A})$ be the smallest convex cone that includes $\mathcal{A}$ as a subset. In other words,

$$\text{cone}(\mathcal{A}) = \{\alpha z : \alpha \geq 0, z \in \text{hull}(\mathcal{A})\}.$$

Theorem 6.15 *Consider the optimization* (6.P3) *for a set* $\mathcal{H}$ *of rank distributions. Let* $\mathcal{C}$ *be a set of rank distributions such that* $\text{cone}(\mathcal{C}) = \text{cone}(\mathcal{H})$. *Then the optimization* (6.P3) *is equivalent if we replace* $\mathcal{H}$ *by* $\mathcal{C}$.

Proof. Let $\boldsymbol{\Psi}$ be a degree distribution that achieves the optimal value $\hat{\alpha}_{\mathcal{H}}$ in (6.P3) for $\mathcal{H}$. Any $\mathbf{h} \in \text{cone}(\mathcal{H}) \cap \mathcal{P}_{\boldsymbol{M}}$ is a conical combination of vectors in $\mathcal{H}$, and by (3) in Theorem 6.9), $\hat{\alpha}_{\mathbf{h},\boldsymbol{\Psi}} \geq \hat{\alpha}_{\mathcal{H}}$. So $\hat{\alpha}_{\text{cone}(\mathcal{H})\cap\mathcal{P}_{\boldsymbol{M}}} \geq \hat{\alpha}_{\mathcal{H}}$. On the other hand, $\hat{\alpha}_{\text{cone}(\mathcal{H})\cap\mathcal{P}_{\boldsymbol{M}}} \leq \hat{\alpha}_{\mathcal{H}}$ since $\mathcal{H} \subset \text{cone}(\mathcal{H}) \cap \mathcal{P}_{\boldsymbol{M}}$. Hence, $\hat{\alpha}_{\text{cone}(\mathcal{H})\cap\mathcal{P}_{\boldsymbol{M}}} = \hat{\alpha}_{\mathcal{H}}$, which can be achieved by $\boldsymbol{\Psi}$.

Applying the same argument for $\mathcal{C}$ in place of $\mathcal{H}$, we obtain $\hat{\alpha}_{\mathcal{C}} = \hat{\alpha}_{\text{hull}(\mathcal{C})\cap\mathcal{P}_{\boldsymbol{M}}} = \hat{\alpha}_{\text{hull}(\mathcal{H})\cap\mathcal{P}_{\boldsymbol{M}}} = \hat{\alpha}_{\mathcal{H}}$, which can be achieved by $\boldsymbol{\Psi}$. □

Theorem 6.16 *For* $\mathcal{H} \subset \mathcal{P}^{\boldsymbol{M}}$ *with* $\mathcal{H} \neq \{(0,\ldots,0)\}$, *optimization* (6.P3) *for* $\mathcal{H}$ *is equivalent to optimization* (6.P2) *for* $\text{cone}(\mathcal{H}) \cap \mathcal{D}_1$.

Proof. To save notation, we write $\sum_{i=1}^{\boldsymbol{M}}$ as $\sum_i$ throughout the proof. Let

$$\mu^*(\mathcal{H}) = \min\left\{\mathbb{E}[\mathbf{h}] : \mathbf{h} \in \text{cone}(\mathcal{H}), \sum_i h_i = 1\right\}.$$

Since $\mathcal{H} \neq \{(0,\ldots,0)\}$, we know that there exists $(h_1,\ldots,h_{\boldsymbol{M}}) \in \text{cone}(\mathcal{H})$ with $\sum_i h_i = 1$. We now show that $\mu^*(\mathcal{H}) \geq 1$. For any $\mathbf{h} = (h_1,\ldots,h_{\boldsymbol{M}}) \in \text{cone}(\mathcal{H})$ with $\sum_i h_i = 1$, since $h_0 = 0$, it is evident that $\mathbb{E}[\mathbf{h}] \geq 1$. Therefore, $\mu^*(\mathcal{H}) \geq 1$. Fix $0 < \mu \leq \mu^*(\mathcal{H})$. As we will see, we only need to consider $\mu = 1$ to prove the theorem.

Next, we show that

$$\text{cone}(\text{cone}(\mathcal{H}) \cap \mathcal{D}_\mu) = \text{cone}(\mathcal{H}). \tag{6.21}$$

Since $\text{cone}(\mathcal{H}) \cap \mathcal{D}_\mu \subset \text{cone}(\mathcal{H})$, we have $\text{cone}(\text{cone}(\mathcal{H}) \cap \mathcal{D}_\mu) \subset \text{cone}(\mathcal{H})$. Fix $\mathbf{h} = (h_1,\ldots,h_{\boldsymbol{M}}) \in \text{cone}(\mathcal{H})$ with $\mathbf{h} \neq (0,\ldots,0)$. Let

$$\mathbf{h}' = (h'_1,\ldots,h'_{\boldsymbol{M}}) = \frac{\mathbf{h}}{\sum_i h_i}$$

and

$$\mathbf{h}'' = (h''_1,\ldots,h''_{\boldsymbol{M}}) = \mu\frac{\mathbf{h}'}{\mathbb{E}[\mathbf{h}']}.$$

We know that $\mathbf{h}', \mathbf{h}'' \in \text{cone}(\mathcal{H})$ and $\sum_i h'_i = 1$. Since

$$\sum_i h''_i = \frac{\mu}{\mathbb{E}[\mathbf{h}']} \leq \frac{\mu^*(\mathcal{H})}{\mathbb{E}[\mathbf{h}']} \leq 1$$

and

$$\mathbb{E}[\mathbf{h}''] = \mu,$$

we have $\mathbf{h}'' \in \text{cone}(\mathcal{H}) \cap \mathcal{D}_\mu$ and hence $\mathbf{h} \in \text{cone}(\text{cone}(\mathcal{H}) \cap \mathcal{D}_\mu)$. Thus, we have $\text{cone}(\mathcal{H}) \subset \text{cone}(\text{cone}(\mathcal{H}) \cap \mathcal{D}_\mu)$.

Due to (6.21), Theorem 6.15 says (6.P3) for $\mathcal{H}$ is equivalent to (6.P3) for $\text{cone}(\mathcal{H}) \cap \mathcal{D}_\mu$, while the latter is the same as (6.P2) for $\text{cone}(\mathcal{H}) \cap \mathcal{D}_\mu$ with the objective function scaled by $1/\mu$. The proof is completed by considering $\mu = 1$. □

6.4 GUARANTEED RATES AND UNIVERSALITY

Using the techniques in the last section, we provide some numerical evaluation examples that demonstrate the performance of BATS codes for multiple rank distributions.

6.4.1 GUARANTEED MULTICAST RATES

We first evaluate the guaranteed multicast rates of BATS codes using BP decoding. For a rank distribution $\mathbf{h} = (h_1, \ldots, h_M)$, we know that the maximum achievable rate of BATS codes is upper bounded by $\mathbb{E}[\mathbf{h}] = \sum_{i=1}^{M} i h_i$. For a real number $0 \le \mu \le M$, define

$$\mathcal{B}_\mu = \{\mathbf{h} \in \mathcal{P}_M : \mathbb{E}[\mathbf{h}] \ge \mu\}.$$

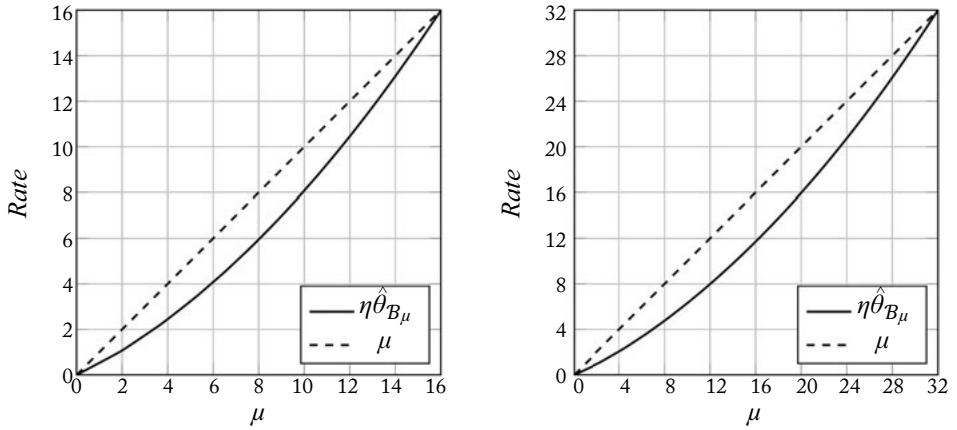
The set $\mathcal{B}_\mu$ includes all the rank distributions that can potentially support rate μ. For any set of rank distribution $\mathcal{H} \subset \mathcal{B}_\mu$, $\hat{\theta}_\mathcal{H} \ge \hat{\theta}_{\mathcal{B}_\mu}$.

For any set of rank distributions $\mathcal{H}$ with $\min_{\mathbf{h}\in\mathcal{H}} \mathbb{E}[\mathbf{h}] = \mu$, $\eta\hat{\theta}_{\mathcal{B}_\mu}$ is a guaranteed multicast rate for $\mathcal{H}$. Note that any multicast rate higher than μ is not achievable by $\mathcal{H}$. Therefore, by comparing $\eta\hat{\theta}_{\mathcal{B}_\mu}$ with μ, we can gain some insight into the achievable multicast rate of BATS codes. Directly solving (6.P2) for $\mathcal{B}_\mu$ is difficult since $\mathcal{B}_\mu$ contains infinitely many of rank distributions, but we can simplify the problem using Theorem 6.14 as prescribed in the following lemma. Recall the convex polytope $\mathcal{D}_\mu$ defined in (6.20).

Lemma 6.17 *The optimization* (6.P2) *for* $\mathcal{B}_\mu$ *is equivalent to the optimization* (6.P2) *for the set of vertices of* $\mathcal{D}_\mu$.

Proof. We see that $\mathcal{D}_\mu \subset \mathcal{B}_\mu$. For each $\mathbf{h} \in \mathcal{B}_\mu \setminus \mathcal{D}_\mu$, define $\mathbf{h}' = \mu\mathbf{h}/\mathbb{E}[\mathbf{h}]$. Then $\mathbf{h}' \in \mathcal{D}_\mu$ and $\mathbf{h} \succcurlyeq \mathbf{h}'$. Thus by Theorem 6.13, (6.P2) for $\mathcal{B}_\mu$ is equivalent to (6.P2) for $\mathcal{D}_\mu$, which by Theorem 6.14 is also equivalent to (6.P2) for the set of vertices of $\mathcal{D}_\mu$. □

We discuss briefly in Appendix C how to find the vertices of $\mathcal{D}_\mu$. See Figure 6.3 for $\eta\hat{\theta}_{\mathcal{B}_\mu}$ when $M = 16$ or 32, $q = 256$, and $\eta = 0.99$. For example, $\eta\hat{\theta}_{\mathcal{B}(10)} = 8.10$ when $M = 16$.



(a) $M = 16$ (b) $M = 32$

Figure 6.3: The optimal value of (6.P2) for $\mathcal{B}_\mu$, where $q = 2^8$ and $\eta = 0.99$.

6.4.2 UNIVERSALITY

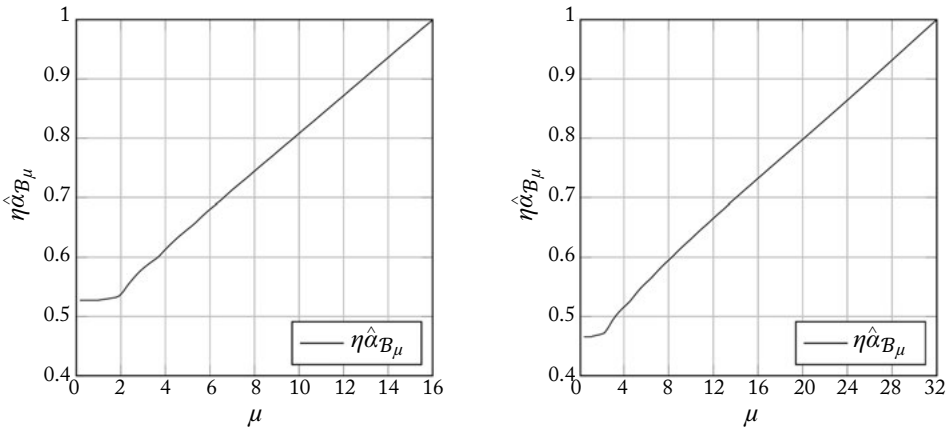
In general, BATS codes are not universal. There does not exist a degree distribution that can achieve rates close to $\sum_i i\hbar_i$ for all rank distributions for a given batch size M, except for $M = 1$, the case of LT/Raptor codes. The universality of BATS codes for a set of degree distributions $\mathcal{H}$ can be measured by $\eta\hat{\alpha}_{\mathcal{H}}$: the more $\eta\hat{\alpha}_{\mathcal{H}}$ is close to 1, the more BATS codes are close to being universal for $\mathcal{H}$. To see the universality of BATS codes, we evaluate (6.P3) for $\mathcal{B}_\mu$, $\mu \in [0, M]$, which can simplified as follows.

Lemma 6.18 *For $0 < \mu \leq M$, optimization (6.P3) for $\mathcal{B}_\mu$ is equivalent to (6.P3) for $\mathcal{D}_\mu$. Moreover, $\hat{\alpha}_{\mathcal{D}_\mu} = \hat{\theta}_{\mathcal{D}_\mu}/\mu$ and for $\mu \in (0, 1]$, $\hat{\alpha}_{\mathcal{D}_\mu} = \hat{\alpha}_{\mathcal{D}_1}$.*

Proof. First $\text{cone}(\mathcal{D}_\mu) \subset \text{cone}(\mathcal{B}_\mu)$ since $\mathcal{D}_\mu \subset \mathcal{B}_\mu$. For any $\mathbf{h} \in \mathcal{B}_\mu \setminus \mathcal{D}_\mu$, since $\mu\mathbf{h}/\mathbb{E}[\mathbf{h}] \in \mathcal{D}_\mu$, we have $\mathcal{B}_\mu \subset \text{cone}(\mathcal{D}_\mu)$, and hence $\text{cone}(\mathcal{B}_\mu) \subset \text{cone}(\mathcal{D}_\mu)$. Therefore, $\text{cone}(\mathcal{B}_\mu) = \text{cone}(\mathcal{D}_\mu)$. By Theorem 6.15, (6.P3) for $\mathcal{B}_\mu$ is equivalent to (6.P3) for $\mathcal{D}_\mu$. Finally, $\hat{\alpha}_{\mathcal{D}_\mu} = \hat{\theta}_{\mathcal{D}_\mu}/\mu$ follows from Lemma 6.8. □

We plot $\eta\hat{\alpha}_{\mathcal{B}_\mu}$ in Figure 6.4 for $\mu \in (0, M]$, where we see that the universality roughly increases linearly with μ with the minimum value at $\mu \in (0, 1]$. The universality of BATS codes for $\mathcal{P}_M$ is given by $\eta\hat{\alpha}_{\mathcal{B}_1}$. In Table 6.3, we give $\eta\hat{\alpha}_{\mathcal{P}_M}$ for $M = 1, 2, 4, \ldots, 64$. Take $M = 16$ as an example. The value $\eta\hat{\alpha} = 0.5274$ implies a worst guaranteed rate for an arbitrary number of destination nodes with arbitrary empirical rank distributions: A destination node can decode the original file with high probability after receiving n batches such that $0.5274 \sum_{i=1}^{n} \text{rk}(\mathbf{H}_i)$

is larger than the number of original input packets, where $\mathbf{H}_i$ is the transfer matrix of the ith batch. When the possible empirical rank distributions are in a smaller set, the optimal value of (6.P3) can be much larger, as in the network with three destination nodes.



(a) $M = 16$ (b) $M = 32$

Figure 6.4: The optimal value of (6.P3) for $\mathcal{B}_\mu$ with $q = 2^8$ and $\eta = 0.99$.

Table 6.3: The maximum value $\hat{\alpha}$ of (6.P3) when $\mathcal{H}$ is the set of all rank distributions for a given batch size, i.e., $\mathcal{P}_M$. Here, $\eta = 0.99$.

M	1	2	4	8	16	32
$\eta\hat{\alpha}$	0.9942	0.8383	0.7068	0.6060	0.5274	0.4657

CHAPTER 7

Finite-Length Analysis of BP Decoding

The asymptotic performance of BATS codes with belief propagation (BP) decoding has been analyzed, and a sufficient condition for the BP decoder to recover a given fraction of the input symbols with high probability was obtained in Chapter 5. This sufficient condition enables us to design BATS codes with good performance for a large number of input symbols (e.g., tens of thousands). It has been verified theoretically for certain special cases and demonstrated numerically for general cases that BATS codes can achieve rates very close to optimality for a given rank distribution of the transfer matrices.

The performance of BATS codes for a relatively small number of input symbols is of important practical interest. For such codes, however, the error bound obtained in the asymptotic analysis is rather loose (if valid), and the degree distribution optimized asymptotically does not give a good performance. Toward designing better BATS codes for a relatively small number of input symbols (e.g., a few hundreds), we analyze in this chapter BATS codes with a finite number of input symbols for BP decoding. Parts of the chapter were published in [54, 96, 102]. More results about finite-length analysis and degree distribution optimizations can be found in [110, 111].

Karp et al. [29] provided a recursive formula to compute the error probability of LT codes for a given number of input symbols. Maneva and Shokrollahi [49] used a random model of the number of coded symbols and obtained a simpler formula for BP decoding. Our finite length analysis results for BATS codes are more than generalizations of the previous results; in particular, Sections 7.2 and 7.3 provide new analytical tools for LT codes.

7.1 STOPPING TIME OF BP DECODING

We follow the discussion of BATS codes in Chapter 2, and asume that the batch degrees $\mathrm{dg}_i, i = 1, \ldots$ are i.i.d. random variables with a given distribution $\mathbf{\Psi} = (\Psi_1, \ldots, \Psi_K)$, the batch transfer matrices $\mathbf{H}_i, i = 1, 2, \ldots$ are independent and follow the same distribution, and $\mathbf{H}_i, i = 1, 2, \ldots$ are also independent of the encoding process. We characterize the distribution of the stopping time of BP(n) for finite values of n (see Section 2.3.1 for the description of BP(n)).

7.1.1 BASIC RECURSIVE FORMULA

Let $R_n^{(t)}$ be the number of decodable input symbols at time t (which is also called the *input ripple size* in the literature of LT codes). Recall that BP decoding stops when there are no decodable input symbols. The probability that BP(n) stops at time t is

$$P_{\text{stop}}(t|n) \triangleq \Pr\left\{R_n^{(t)} = 0, R_n^{(\tau)} > 0, \tau < t\right\}.$$

Let $C_n^{(t)}$ be the number of undecodable batches at time t. Define an $(n+1) \times (K-t+1)$ matrix $\mathbf{\Lambda}_n^{(t)}$ as

$$\mathbf{\Lambda}_n^{(t)}[c, r] \triangleq \Pr\left\{C_n^{(t)} = c, R_n^{(t)} = r, R_n^{(\tau)} > 0, \tau < t\right\}, \tag{7.1}$$

where $c = 0, 1, \ldots, n$ and $r = 0, 1, \ldots, K-t$. With the definition of $\mathbf{\Lambda}_n^{(t)}[c, r]$, we have

$$P_{\text{stop}}(t|n) = \sum_{c=0}^{n} \mathbf{\Lambda}_n^{(t)}[c, 0]. \tag{7.2}$$

We will express $\mathbf{\Lambda}_n^{(t)}$ in terms of $\mathbf{\Lambda}_n^{(t-1)}$, so that we can calculate $\mathbf{\Lambda}_n^{(t)}$ recursively for $t = 0, \ldots, K$, which together with (7.2) gives a formula to calculate $P_{\text{stop}}(t|n)$.

Let

$$\text{Bi}(k; n, p) \triangleq \binom{n}{k} p^k (1-p)^{n-k}$$

and

$$\text{hyge}(k; n, i, j) \triangleq \begin{cases} \frac{\binom{i}{k}\binom{n-i}{j-k}}{\binom{n}{j}} & \max\{0, i+j-n\} \le k \le \min\{i, j\} \\ 0 & o.w. \end{cases}$$

be the p.m.f. of the binomial distribution and the hypergeometric distribution, respectively.

Theorem 7.1 *Consider a BATS code with K input symbols, n batches, degree distribution $\mathbf{\Psi}$, rank distribution $\mathbf{h}$ of the transfer matrices, and batch size M. Define*

$$p_{t,s} = \begin{cases} \Psi_s \hbar'_s & t = 0, \\ \hbar_s \sum_{d=s+1}^{s+t} \Psi_d \frac{d}{K} \text{hyge}(d-s-1; K-1, d-1, t-1) & t > 0, s+t \le K, \\ 0 & t > 0, s+t > K, \end{cases}$$

and

$$p_t = \sum_{s=0}^{M} p_{t,s}.$$

We have

$$\mathbf{\Lambda}_n^{(0)}[c,:] = \mathrm{Bi}(c;n,1-\rho_0)\mathbf{e}_0\mathbf{Q}_0^{n-c}, \tag{7.3}$$

and for $t > 0$,

$$\mathbf{\Lambda}_n^{(t)}[c,:] = \sum_{c'=c}^{n} \mathrm{Bi}(c;c',1-\rho_t)\mathbf{\Lambda}_n^{(t-1)}[c',1:]\mathbf{Q}_t^{c'-c} \tag{7.4}$$

where ρ_t *and* $\mathbf{Q}_t$ *are defined as follows:*

1. $\rho_0 = p_0$.
2. *For* $t > 0$,
$$\rho_t = \frac{p_t}{1-\sum_{\tau=0}^{t-1} p_\tau}.$$
3. *For* $t = 0, 1, \ldots, K$, $\mathbf{Q}_t$ *is a* $(K-t+1)\times(K-t+1)$ *matrix with*
$$\mathbf{Q}_t[i,j] = \sum_{s=j-i}^{j\wedge M} \frac{p_{t,s}}{p_t}\mathrm{hyge}(i+s-j;K-t,i,s) \tag{7.5}$$
for $0\vee(j-M) \le i \le j \le K-t$, *and* $\mathbf{Q}_t[i,j] = 0$ *otherwise,*

where $x \vee y$ *is the maximum of* x *and* y.

Proof. The proof is left to Appendix D.1. The idea is to characterize the corresponding probability transition matrix between two consecutive decoding times. □

Remark 7.2 In the above theorem, when $p_t = 0$, $\mathbf{Q}_t$ involves undefined entries 0/0. We use the following convention for the iterative formula in the above theorem:

1. $0\mathbf{Q}_t^k = \mathbf{0}$, $k \ge 1$; and
2. $\mathbf{Q}_t^0$ is the identity matrix.

With these convention, we have that when $p_0 = 0$,

$$\begin{aligned}&\mathbf{\Lambda}_n^{(0)}[c,:] = \mathbf{0}, \quad \text{for } c = 0,1,\ldots,n-1,\\ &\mathbf{\Lambda}_n^{(0)}[n,:] = \mathbf{e}_0.\end{aligned}$$

When $p_t = 0$, $t > 0$,

$$\mathbf{\Lambda}_n^{(t)}[c,:] = \mathbf{\Lambda}_n^{(t-1)}[c,1:].$$

We will use the same convention to simplify the formulae in this and the next chapter without further elaborations.

7.1.2 EXPLANATIONS OF SOME NOTATIONS

The notations defined in Theorem 7.1 deserve some explanations. First, $p_{t,s}$ is the probability that a batch is decodable for the first time at time t and has degree s at time t. Then p_t is the probability that a batch is decodable for the first time at time t. Note that $p_0 = 0$ is equivalent to $\sum_{d=1}^{M} \Psi_d \hbar'_d = 0$, i.e., the probability that a batch is decodable at time 0 is 0. In this case, we have $P_{\text{stop}}(0|n) = \Pr\{R_n^0 = 0\} = 1$, i.e., the decoding stops at time 0 with probability 1. When the probability that a batch is decodable at time 0 is positive, the following lemma (proved in Appendix D.2) implies that $p_t > 0$ for $t = 0, 1, \ldots, K$.

The probability that a batch is decodable at time 0 is equal to $\sum_{d=1}^{M} \Psi_d \hbar'_d$. Define r_{BP} as the smallest integer d such that $\Psi_d \hbar'_d > 0$. Under the assumption that $\sum_{d=1}^{M} \Psi_d \hbar'_d > 0$, i.e., *the probability that a batch is decodable at time* 0 *is positive*, r_{BP} is well defined.

Lemma 7.3 *When a batch is decodable at time* 0 *with positive probability,*

$$p_{t,s} \begin{cases} = 0, & \text{for } t + s < r_{\text{BP}}, \\ > 0, & \text{for } t = 0 \text{ and } s = r_{\text{BP}}, \\ > 0, & \text{for } t \geq 1, t + s \geq r_{\text{BP}} \text{ and } s < r_{\text{BP}}. \end{cases}$$

We also note that ρ_t $(t > 0)$ is the probability that a batch is decodable at time t under the condition that it is not decodable at time $t - 1$. The following properties about ρ_t and p_t are straightforward and they are proved in Appendix D.2.

Lemma 7.4

1. *For* $0 \leq t \leq K$,
$$\prod_{\tau=0}^{t} (1 - \rho_\tau) = 1 - \sum_{\tau=0}^{t} p_\tau.$$
2. *For* $0 < t \leq K$,
$$\rho_t \prod_{\tau=0}^{t-1} (1 - \rho_\tau) = p_t.$$

When defined, the matrix $\mathbf{Q}_t$ can be regarded as a transition matrix. Suppose k batches become decodable at time t and we generate new decodable input symbols from these k batches one batch after another. Define random variable $Z_0 = R_n^{(t-1)} - 1$ for $t > 0$ or $Z_0 \equiv 0$ for $t = 0$, and for $i = 1, \ldots, k$ define Z_i as the total number of decodable input symbols after having

generated new decodable input symbols from the first i decodable batches. Note that $Z_k = R_n^{(t)}$. Then $Z_0, \ldots, Z_k$ forms a homogeneous Markov chain with the transition matrix $\mathbf{Q}_t$.

To evaluate the formulae in Theorem 7.1, we first calculate $p_{t,s}$ for $t = 0, 1, \ldots, K$ and $s = 0, 1, \ldots, M$, which takes $O(K^2M)$ real number operations. We then calculate ρ_t and $\mathbf{Q}_t$ for $t = 0, 1, \ldots, K$ using $O(KM)$ and $O(K^2M^2)$ real number operations, respectively. Thus, it totally takes $O(K^2M^2)$ real number operations to calculate ρ_t and $\mathbf{Q}_t$. Note that $p_{t,s}$, ρ_t, and $\mathbf{Q}_t$ do not depend on n, and are determined by K, $\mathbf{\Psi}$ and $\mathbf{h}$ only. Once they are calculated, we can use them in the evaluation of $\mathbf{\Lambda}_n^t$ for different values of n. Note that the matrix $\mathbf{Q}_t$ has at most $M + 1$ non-zero entries in each column, so the vector-matrix multiplication takes $O(KM)$ real number operations. Since a total of $O(Kn^2)$ such vector-matrix multiplications are used in the formulae, the complexity for computing $P_{\text{stop}}(t|n)$ using Theorem 7.1 is $O(K^2M^2 + K^2n^2M)$ real number operations.

7.1.3 SPECIAL CASES

Let us examine a few special cases of the formulae in Theorem 7.1.

Example 7.5 ($\Psi_1 = 1$) Consider a BATS code with $\Psi_1 = 1$. In this special case, every batch has degree one and a packet in the batch is generated by the input packet involved in the batch multiplied by a scalar. The decoding becomes the *coupon collector's problem*, where we can treat the input packets as the coupons and the batches as the boxes.[1] In this case, when a batch is decodable at time 0, it can recover one input packet; otherwise, the batch must have rank zero at time 0.

It can be calculated that $p_{0,1} = \hbar'_1$ and $p_{0,s} = 0$ for $s \neq 1$, i.e., a batch is decodable at time 0 with rank 1 and probability $\hbar'_1$. All the components of $\mathbf{Q}_0$ are zero except that $\mathbf{Q}_0[i, i] = i/K$ for $i = 1, \ldots, K$ and $\mathbf{Q}_0[i, i+1] = 1 - i/K$ for $i = 0, \ldots, K-1$. When $t > 0$, we have $p_{t,0} = \hbar_0/K$ and $p_{t,s} = 0$ for $s > 0$, i.e., a batch is decodable with rank 0 and probability $\hbar_0/K$. When $t > 0$, $\mathbf{Q}_t$ is the identity matrix.

Example 7.6 ($t = K$) In this example, we consider a general degree distribution and a general rank distribution, and look at the decoding status when $t = K$. At time K, all the input symbols are decoded so that all the batches have degree 0 at time K (after substitution). We have $p_{K,0} = \hbar_0 \sum_{d=1}^{K} \Psi_d \frac{d}{K}$, $p_{K,s} = 0$ for $s > 0$, $\rho_K = 1$, and $\mathbf{Q}_K = [1]$.

Example 7.7 (LT Codes) Letting $M = 1, \hbar_0 = h_0, \hbar'_0 = 1$ and $\hbar_1 = \hbar'_1 = h_1$ in Theorem 7.1, we obtain

$$p_{0,1} = \Psi_1 h_1 \text{ and } p_{0,0} = 0,$$

[1] In the most discussed form of the coupon collector's problem, each box contains one of the coupons. But in our case, it is possible that a box contains nothing (i.e., the batch is not decodable at time 0).

and for $t > 0$,

$$p_{t,0} = h_0 \sum_{d=1}^{t} \Psi_d \frac{d}{K} \frac{\binom{K-d}{t-d}}{\binom{K-1}{t-1}} = h_0 \sum_{d=1}^{t} \Psi_d \frac{\binom{t-1}{d-1}}{\binom{K}{d}},$$

$$p_{t,1} = \begin{cases} h_1 \sum_{d=2}^{t+1} \Psi_d \frac{d(d-1)}{K} \frac{\binom{K-d}{t-d+1}}{\binom{K-1}{t-1}} = h_1 \sum_{d=2}^{t+1} \Psi_d (K-t) \frac{\binom{t-1}{d-2}}{\binom{K}{d}} & t < K, \\ 0 & t = K. \end{cases}$$

The matrix $\mathbf{Q}_t$, $t = 0, 1, \ldots, K-1$ has the following expression: for $i = 0, \ldots, K-t$,

$$\mathbf{Q}_t[i, i] = \frac{p_{t,0}}{p_t} + \frac{p_{t,1}}{p_t} \frac{i}{K-t},$$

for $i = 0, \ldots, K-t-1$,

$$\mathbf{Q}_t[i, i+1] = \frac{p_{t,1}}{p_t} \left(1 - \frac{i}{K-t}\right),$$

and $\mathbf{Q}_t[i, j] = 0$ otherwise.

Karp et al. [29] has given a formula for LT codes to recursively calculate the joint distribution of the number of decodable received symbols (called *output ripple size*) and the number of undecodable received symbols at each decoding step. Note that the distribution of output ripple size determines the distribution of the input ripple size. Their formula is given in a polynomial form and has an evaluation bit-complexity $O(n^3 \log^2(n) \log\log(n))$ based on polynomial evaluation and interpolation.

Note that it is possible to extend the approach in [29] for $M > 1$, i.e., recursively calculating the joint distribution of the number of decodable batches and the number of undecodable batches. When $M > 1$, decodable batches with different degrees must be considered separately and M recursive formulae must be provided for each positive degree value of the decodable batches. The evaluation complexity of this extension increases exponentially with M (see an outline of this extension in [55, Appendix]). Our approach [54, 102], which instead tracks the number of decodable input symbols and the number of undecodable batches at each step, gives a formula with complexity equal to a quadratic function of M. Further, our formula is given in a matrix form, which facilitates certain analyses as we will demonstrate in this monograph.

7.2 FURTHER RESULTS ON BP DECODING

In this section, we study the following performance measues for BP decoding:

1. the distribution of the stopping time of BP(n) for a sequence of n;
2. the decrease rate of the error probability of BP(n) when n increases;

3. the distribution of the number of batches consumed by BP* (defined in Section 2.3.2); and

4. the expected number of batches consumed by BP*.

7.2.1 STOPPING TIME DISTRIBUTION

For a given number n, $\mathbf{\Lambda}_n^{(t)}$ can be calculated recursively for $t = 0, \ldots, K$ using Theorem 7.1 and hence the stopping time distribution $P_{\text{stop}}(\cdot|n)$ can be calculated using (7.2). But for applications that will be discussed later in this section, we may want to calculate $P_{\text{stop}}(\cdot|n')$ for $n' = 0, 1, \ldots, n$, where $n > 0$ is a given integer. Using the formula in Theorem 7.1, we have to run the program for each value of n'. In Theorem 7.8, we will propose a new formula that can simplify the calculation of $P_{\text{stop}}(\cdot|n)$ for a range of n.

Theorem 7.8 *For $n \geq 0$ and $t \geq 0$,*

$$P_{\text{stop}}(t|n) = \sum_{c=0}^{n} \binom{n}{c} \left(1 - \sum_{\tau=0}^{t} p_\tau\right)^c \mathbf{\Lambda}_{n-c}^{(t)}[0,0], \tag{7.6}$$

where the first row of the matrices $\mathbf{\Lambda}_{n'}^{(t)}$, $n' = 0, 1, \ldots, n$ can be computed by the following recursion: For $n' = 0, 1, \ldots, n$,

$$\mathbf{\Lambda}_{n'}^{(0)}[0,:] = (p_0 \mathbf{Q}_0)^{n'}[0,:], \tag{7.7}$$

and for $t > 0$

$$\mathbf{\Lambda}_{n'}^{(t)}[0,:] = \sum_{c=0}^{n'} \binom{n'}{c} \mathbf{\Lambda}_{n'-c}^{(t-1)}[0,1:](p_t \mathbf{Q}_t)^c. \tag{7.8}$$

Proof. The formula in Theorem 7.1 implies that

$$\mathbf{\Lambda}_n^{(t)}[c,:] = \binom{n}{c} \prod_{i=0}^{t} (1-\rho_i)^c \mathbf{\Lambda}_{n-c}^{(t)}[0,:].$$

The theorem can be derived by substituting the above equality into (7.2) and (7.4). See the details in Appendix D.3. □

Remark 7.9 The above theorem only involves the 0-th row of $\mathbf{\Lambda}_{n'}^{(t)}$, which has a particular meaning. According to the definition, we know that

$$\mathbf{\Lambda}_{n'}^{(t)}[0,r] = \Pr\left\{C_{n'}^{(t)} = 0, R_{n'}^{(t)} = r, R_{n'}^{(\tau)} > 0, \tau < t\right\}.$$

In other words, $\mathbf{\Lambda}_{n'}^{(t)}[0,r]$ is the probability that the input ripple size is r and all the batches are decodable for a BATS code with n' batches at time t.

For a given number $n > 0$, the above theorem provides us a new representation of $P_{\text{stop}}(\cdot|n)$ in terms of $\mathbf{\Lambda}_{n'}^{(t)}[0,0]$ for $n' = 0, 1, \ldots, n$, and a recursive formula given by (7.7) and (7.8) to calculate $\mathbf{\Lambda}_{n'}^{(t)}[0,:]$ for $t = 0, 1, \ldots, K$ and $n' = 1, \ldots, n$. To evaluate the formulae in the above theorem, we first use (7.7) to calculate $\mathbf{\Lambda}_{i}^{(0)}[0,:]$ for $i = 0, 1, \ldots, n$. For $t > 0$, we use the following recursive formulae induced by (7.8) to calculate $\mathbf{\Lambda}_{i}^{(t)}[0,:]$ for $i = 0, 1, \ldots, n$:

$$\begin{aligned}
\mathbf{\Lambda}_0^{(t)}[0,:] &= \binom{0}{0}\mathbf{\Lambda}_0^{(t-1)}[0,1:](p_t\mathbf{Q}_t)^0 \\
\mathbf{\Lambda}_1^{(t)}[0,:] &= \binom{1}{0}\mathbf{\Lambda}_1^{(t-1)}[0,1:](p_t\mathbf{Q}_t)^0 + \binom{1}{1}\mathbf{\Lambda}_0^{(t-1)}[0,1:](p_t\mathbf{Q}_t)^1 \\
\mathbf{\Lambda}_2^{(t)}[0,:] &= \binom{2}{0}\mathbf{\Lambda}_2^{(t-1)}[0,1:](p_t\mathbf{Q}_t)^0 + \binom{2}{1}\mathbf{\Lambda}_1^{(t-1)}[0,1:](p_t\mathbf{Q}_t)^1 + \binom{2}{2}\mathbf{\Lambda}_0^{(t-1)}[0,1:](p_t\mathbf{Q}_t)^2 \\
&\vdots \\
\mathbf{\Lambda}_n^{(t)}[0,:] &= \binom{n}{0}\mathbf{\Lambda}_n^{(t-1)}[0,1:](p_t\mathbf{Q}_t)^0 + \binom{n}{1}\mathbf{\Lambda}_{n-1}^{(t-1)}[0,1:](p_t\mathbf{Q}_t)^1 + \ldots + \binom{n}{n}\mathbf{\Lambda}_0^{(t-1)}[0,1:](p_t\mathbf{Q}_t)^n.
\end{aligned}$$

This theorem is more convenient to use when we want to calculate $P_{\text{stop}}(\cdot|n')$ for $n' = 1, \ldots, n$, which has the same complexity $O(K^2M^2 + K^2n^2M)$ as calculating $P_{\text{stop}}(\cdot|n)$ only using Theorem 7.1.

7.2.2 POWER-SUM FORMULA

The matrix $\mathbf{Q}_t$ defined in Theorem 7.1 is upper-triangular. The following lemma, proved in Appendix D.2, shows that $\mathbf{Q}_t$ is also diagonalizable.

Lemma 7.10 *The matrix $\mathbf{Q}_t$ is diagonalizable, i.e.,*

$$\mathbf{Q}_t = \mathbf{U}_t\mathbf{D}_t\mathbf{U}_t^{-1},$$

where $\mathbf{D}_t$ is a diagonal matrix with $\mathbf{D}_t[i,i] = \mathbf{Q}_t[i,i]$, $\mathbf{U}_t$ is an upper-triangular matrix with $\mathbf{U}_t[i,j] = \binom{K-t-i}{j-i}$ for $i \le j$, and $\mathbf{U}_t^{-1}$ is an upper-triangular matrix with $\mathbf{U}_t^{-1}[i,j] = (-1)^{j-i}\binom{K-t-i}{j-i}$ for $i \le j$.

In the above decomposition, the degree and rank distributions only affect $\mathbf{D}_t$, i.e., the eigenvalues of $\mathbf{Q}_t$. The matrix $\mathbf{U}_t$ depends only on K and t. We also notice that $\mathbf{U}_t[1:,1:] =$

$\mathbf{U}_{t+1}$ and $\mathbf{U}_t^{-1}[1:,1:] = \mathbf{U}_{t+1}^{-1}$. Substituting the above decomposition of $\mathbf{Q}_t$ into Theorem 7.8, we obtain another formula for $P_{\text{stop}}(t|n)$ with a power-sum form.

Theorem 7.11 *For $n \geq 0$ and $t \geq 0$,*

$$P_{\text{stop}}(t|n) = \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0] \left(1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0]\right)^n,$$

where row vector $\mathbf{V}_{t,i}$ and diagonal matrix $\mathbf{\Delta}_{t,i}$ are defined as follows:

1. $\mathbf{V}_{0,0} \triangleq \mathbf{U}_0[0,:]$ *and* $\mathbf{\Delta}_{0,0} \triangleq p_0\mathbf{D}_0$,
2. *For $t \geq 0$ and $i = 0, 1, \ldots, 2^t - 1$,*

$$\begin{aligned} \mathbf{V}_{t+1,i} &= \mathbf{V}_{t,i}[1:], \\ \mathbf{\Delta}_{t+1,i} &= \mathbf{\Delta}_{t,i}[1:,1:] + p_{t+1}\mathbf{D}_{t+1}, \\ \mathbf{V}_{t+1,2^t+i} &= -\mathbf{V}_{t,i}[0]\mathbf{U}_t[0,1:], \\ \mathbf{\Delta}_{t+1,2^t+i} &= \mathbf{\Delta}_{t,i}[0,0]\mathbf{I} + p_{t+1}\mathbf{D}_{t+1}. \end{aligned} \tag{7.9}$$

Proof. This theorem can be proved by substituting the diagonal decomposition of $\mathbf{Q}_t$ in Lemma 7.10 into Theorem 7.8. The details can be found in Section D.3. □

The formula in Theorem 7.11 is a linear combination of 2^t n-th powers, where the number of batches n appears only in the power, but in neither $\mathbf{V}_{t,i}$ nor $\mathbf{\Delta}_{t,i}$. It is now easy to see that $P_{\text{stop}}(t|n)$ decreases exponentially with n, which will be made explicit in the next subsection. Note that $\mathbf{V}_{t,i}[0]$ are integers determined by K, t, and i, but not n, and can be both positive and negative. According to the definition, we also know that for $t = 0, 1, \ldots, K-1$,

$$0 < \sum_{\tau=0}^{t} p_\tau - \mathbf{\Delta}_{t,i}[0,0] < 1.$$

We prefer Theorem 7.8 to Theorem 7.11 for numerical evaluation for two reasons. First, because of the 2^t n-th power for $t = 0, 1, \ldots, K$ in the formula for $P_{\text{stop}}(t|n)$ in Theorem 7.11, the computation complexity increases exponentially with K. Second, the absolute value of $\mathbf{V}_{t,i}[0]$ can be very large, so that the accuracy of the numerical evaluation is difficult to guarantee if we use a fixed number of significant digits.

7.2.3 ERROR PROBABILITY AND ERROR EXPONENT

For BP(n), we say a decoding error occurs if the decoder cannot recover all the K input symbols, i.e., the decoder stops before time K. Hence, the corresponding *error probability* is

$$P_{\text{err}}(n) = \sum_{t=0}^{K-1} P_{\text{stop}}(t|n) = 1 - P_{\text{stop}}(K|n).$$

Using Theorem 7.8, we can calculate $P_{\text{err}}(n)$ efficiently.

The asymptotic decrease rate of the error probability of BP(n) with respect to n can be characterized using the *BP error exponent* of BATS codes defined as

$$\text{EE}_{\text{BP}} = \lim_{n\to\infty} \frac{-\log(P_{\text{err}}(n))}{n}.$$

For $0 \le t \le K$, define

$$q_t = 1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,0}[0,0]. \tag{7.10}$$

Referring to the power-sum formula for $P_{\text{stop}}(t|n)$ in Theorem 7.11, the first summand on the RHS is $\mathbf{V}_{t,0}[0]q_t^n$. Applying (7.9) iteratively, we obtain

$$\mathbf{\Delta}_{t,0}[0,0] = \sum_{\tau=0}^{t} p_\tau \mathbf{D}_\tau[t-\tau, t-\tau].$$

Here, $\mathbf{D}_\tau[t-\tau, t-\tau]$ is the probability that a batch decodable for the first time at time τ does not increase the input ripple size when the input ripple size is $t-\tau$ at time τ.

Recall the definition of r_{BP} above Lemma 7.3. The following theorem enables us to characterize the BP error exponent.

Theorem 7.12 *Suppose that a batch is decodable at time* 0 *with positive probability. We have*

1. $P_{\text{stop}}(0|n) = q_0^n$;
2. *for* $1 \le t < r_{\text{BP}}$, $P_{\text{stop}}(t|n) = 0$ *for all* $n \ge 1$; *and*
3. *for* $t \ge r_{\text{BP}}$,

$$\lim_{n\to\infty} \frac{-\log P_{\text{stop}}(t|n)}{n} = -\log q_t.$$

Proof. This theorem is derived using Theorem 7.11. See the details in Appendix D.3. □

Remark 7.13 The above theorem says that $\mathbf{V}_{t,0}q_t^n$ is the dominating term of $P_{\text{stop}}(t|n)$ when n is large.

Corollary 7.14 *When a batch is decodable at time* 0 *with positive probability, the BP error exponent of BATS codes satisfies*

$$\text{EE}_{\text{BP}} = -\log q^*,$$

where $q^* \triangleq q_0 \vee (\vee_{t=r_{\text{BP}}}^{K-1} q_t) = \vee_{t=0}^{K-1} q_t$.

Proof. The corollary follows the above theorem and $P_{\text{err}}(n) = \sum_{t=0}^{K-1} P_{\text{stop}}(t|n)$. The equality $q_0 \vee (\vee_{t=r_{\text{BP}}}^{K-1} q_t) = \vee_{t=0}^{K-1} q_t$ follows by $q_0 \geq q_t$ for $t < r_{\text{BP}}$. (By checking the proof of Theorem 7.12, we know $q_t = 1 - \sum_{\tau=0}^{t} p_\tau$ for $t < r_{\text{BP}}$.) □

Recall that q_t, $t = 0, 1, \ldots, K-1$, are all functions of $\mathbf{\Psi}$. Since $\max_{\mathbf{\Psi}} \text{EE}_{\text{BP}} = -\log \min_{\mathbf{\Psi}} q^*$, we can obtain the maximum BP error exponent by solving $\min_{\mathbf{\Psi}} q^*$ as the following linear program for given K and the rank distribution:

$$\begin{aligned} &\min_{\mathbf{\Psi},x} x \\ &\text{s.t. } q_t \leq x, \quad t = 0, 1, \ldots, K-1. \end{aligned} \tag{7.11}$$

The variables in the above optimization are the degree distribution and x.

7.2.4 NUMBER OF BATCHES CONSUMED

We now consider the decoder BP* described in Section 2.3.2. We are interested in the number of batches consumed when BP* decodes all the input symbols, which is denoted by N_{BP^*}. We assume that a batch is decodable at time 0 with positive probability in this subsection, since otherwise BP* does not stop.

It is possible to characterize the distribution of N_{BP^*} using the error probability of BP(n). The event $N_{\text{BP}^*} \geq n$ is the same as the event that BP($n-1$) stops with less than K input symbols decoded. So we have for $n \geq 1$,

$$\Pr\{N_{\text{BP}^*} \geq n\} = P_{\text{err}}(n-1). \tag{7.12}$$

By (7.12), we can write

$$\mathbb{E}[N_{\text{BP}^*}] = \sum_{n=1}^{\infty} n \Pr\{N_{\text{BP}^*} = n\} = \sum_{n=1}^{\infty} \Pr\{N_{\text{BP}^*} \geq n\} = \sum_{n=0}^{\infty} P_{\text{err}}(n). \tag{7.13}$$

Using the fact that $P_{\text{err}}(n)$ decreases exponentially fast as n is large (see Corollary 7.14), we know that $\mathbb{E}[N_{\text{BP}^*}] < \infty$.

The *coding overhead* of a BATS code is defined as

$$\mathrm{CO} = \sum_{i=1}^{N_{\mathrm{BP}^*}} \mathrm{rk}(\mathbf{H}_i) - K.$$

We are interested in the expected coding overhead

$$\mathbb{E}[\mathrm{CO}] = \mathbb{E}[N_{\mathrm{BP}^*}]\,\mathbb{E}[\mathrm{rk}(\mathbf{H})] - K = \mathbb{E}[N_{\mathrm{BP}^*}]\bar{h} - K,$$

where the first equality holds by Wald's equation.[2]

For given K and $\bar{h}$, we can calculate $\mathbb{E}[N_{\mathrm{BP}^*}]$ and then obtain $\mathbb{E}[\mathrm{CO}]$. The next theorem gives another formula for $\mathbb{E}[N_{\mathrm{BP}^*}]$.

Theorem 7.15

$$\mathbb{E}[N_{\mathrm{BP}^*}] = \sum_{t=0}^{K-1} \sum_{i=0}^{2^t-1} \frac{\mathbf{V}_{t,i}[0]}{\sum_{\tau=0}^{t} p_\tau - \mathbf{\Delta}_{t,i}[0,0]}. \tag{7.14}$$

Proof. By (7.13), we have $\mathbb{E}[N_{\mathrm{BP}^*}] = \sum_{t=0}^{K-1} \sum_{n=0}^{\infty} P_{\mathrm{stop}}(t|n)$. The proof is completed by applying Theorem 7.11:

$$\sum_{n=0}^{\infty} P_{\mathrm{stop}}(t|n) = \sum_{n=0}^{\infty} \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0] \left(1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0]\right)^n = \sum_{i=0}^{2^t-1} \frac{\mathbf{V}_{t,i}[0]}{\sum_{\tau=0}^{t} p_\tau - \mathbf{\Delta}_{t,i}[0,0]}.$$

□

We prefer (7.13) to (7.14) for numerical evaluations. Fix a sufficiently large integer n_2, and we can approximate $\mathbb{E}[N_{\mathrm{BP}^*}]$ by

$$\mathbb{E}[N_{\mathrm{BP}^*}] \approx \sum_{n=0}^{n_2} P_{\mathrm{err}}(n). \tag{7.15}$$

The approximation error is exponentially small in terms of n_2 (implied by Corollary 7.14).

[2]The following sufficient conditions for Wald's equation [84] can readily be verified:

1. $\mathrm{rk}(\mathbf{H}_i)$, $i = 1, 2 \ldots$ are i.i.d and all have the same finite absolute expectation $\bar{h}$;
2. N_{BP^*} is a stopping time of $\{\mathrm{rk}(\mathbf{H}_i)\}$ (i.e., the event $\{N_{\mathrm{BP}^*} \geq n\}$ depends only on the first $n-1$ batches);
3. $\mathbb{E}[N_{\mathrm{BP}^*}] < \infty$.

This set of conditions are sufficient to show that $\mathbb{E}[\sum_{i=1}^{N_{\mathrm{BP}^*}} \mathrm{rk}(\mathbf{H}_i)] = \mathbb{E}[N_{\mathrm{BP}^*}]\bar{h}$ (see [7] for a proof).

7.3 POISSON NUMBER OF BATCHES

In this section, we study the stopping time of BP($\tilde{N}$) where $\tilde{N}$ is a Poisson random variable, i.e., the number of batches used by the BP decoder follows a Poisson distribution. In network communications, the number of received packets in a given time interval is very often modeled by a Poisson distribution. Therefore, the Poisson model for the number of the batches is useful for evaluating the performance of BATS code in such network models. In addition, the analysis of BP($\tilde{N}$) will provide an alternative formula for calculating $\mathbb{E}[N_{\mathrm{BP}^*}]$.

7.3.1 RECURSIVE FORMULAE

The Poisson random variable $\tilde{N}$ can be represented by its expectation $\bar{n}$, with

$$\Pr\left\{\tilde{N}=n\right\}=\frac{\bar{n}^n}{n!}e^{-\bar{n}}.$$

For any integer t $(0\le t\le K)$ and real value $\bar{n}>0$, define a row-vector $\tilde{\mathbf{\Lambda}}_{\bar{n}}^{(t)}$ of length $K-t+1$ as

$$\tilde{\mathbf{\Lambda}}_{\bar{n}}^{(t)}[r]\triangleq\sum_n \Pr\left\{\tilde{N}=n\right\}\Pr\left\{R_n^{(t)}=r, R_n^{(\tau)}>0, \tau<t\right\},\quad r=0,1,\ldots,K-t.$$

According to the definition in (7.1), we have

$$\tilde{\mathbf{\Lambda}}_{\bar{n}}^{(t)}=\sum_n \Pr\{\tilde{N}=n\}\sum_{c=0}^{n}\mathbf{\Lambda}_n^{(t)}[c,:]. \tag{7.16}$$

Denote by $\tilde{P}_{\mathrm{stop}}(t|\bar{n})$ the probability that BP($\tilde{N}$) stops at time t, where $\mathbb{E}[\tilde{N}]=\bar{n}$. We see that

$$\tilde{P}_{\mathrm{stop}}(t|\bar{n})=\tilde{\mathbf{\Lambda}}_{\bar{n}}^{(t)}[0]=\sum_n \Pr\{\tilde{N}=n\}P_{\mathrm{stop}}(t|n), \tag{7.17}$$

where the second equality follows from (7.2) and (7.16). The above formula for $\tilde{P}_{\mathrm{stop}}(t|\bar{n})$ can be calculated using Theorem 7.8 with complexity $\mathrm{O}(K^2M^2+K^2n_{\max}^2M)$ of real number operations, where we use the first $n_{\max}$ summands for approximation. Due to the fast decrease of $\Pr\{\tilde{N}=n\}$ when $n>\bar{n}$, we may choose $n_{\max}$ such that $\sum_{n=n_{\max}+1}^{\infty}\Pr\{\tilde{N}=n\}$ is small, which gives an upper bound on the approximation error tolerance.

In the following, we show that $\tilde{\mathbf{\Lambda}}_{\bar{n}}^{(t)}$ can be expressed using a different formula, which provides a new perspective on the quantity $\tilde{\mathbf{\Lambda}}_{\bar{n}}^{(t)}$ and a simpler method of evaluating $\tilde{P}_{\mathrm{stop}}(t|\bar{n})$ than (7.17) for certain cases. Define the matrix exponential $\exp(\mathbf{A})$ for a square matrix $\mathbf{A}$ as

$$\exp(\mathbf{A})\triangleq\sum_{i=0}^{\infty}\frac{\mathbf{A}^i}{i!}.$$

Theorem 7.16 *Consider BP decoding of a BATS code with K input symbols, degree distribution $\mathbf{\Psi}$, and transfer matrix rank distribution $\mathbf{h}$. When the number of batches used by BP decoding is Poisson distributed with expectation $\bar{n}$, for any integer $t \geq 0$,*

$$\tilde{\mathbf{\Lambda}}_{\bar{n}}^{(t)} = \tilde{\mathbf{\Lambda}}_{\bar{n}}^{(t-1)}[1{:}] \exp\left(\bar{n}\, p_t(\mathbf{Q}_t - \mathbf{I})\right), \tag{7.18}$$

where $\tilde{\mathbf{\Lambda}}_{\bar{n}}^{-1}[1{:}] \triangleq \mathbf{e}_0$.

Proof. We show the proof of (7.18) for $t = 0$ here. The remainder of the proof can be found in Appendix D.4. Substituting $\Pr\{\tilde{N} = n\}$ and $\mathbf{\Lambda}_n^{(0)}[c, :]$ given in Theorem 7.1, we have

$$\begin{aligned}
\tilde{\mathbf{\Lambda}}_{\bar{n}}^{(0)} &= \sum_n \frac{\bar{n}^n}{n!} e^{-\bar{n}} \sum_{c \leq n} \mathrm{Bi}(c; n, 1-\rho_0) \mathbf{e}_0 \mathbf{Q}_0^{n-c} \\
&= \sum_{c,n:c \leq n} \frac{\bar{n}^n}{n!} e^{-\bar{n}} \binom{n}{c} (1-\rho_0)^c (\rho_0)^{n-c} \mathbf{e}_0 \mathbf{Q}_0^{n-c} \\
&= e^{-\bar{n}} \mathbf{e}_0 \sum_{c,n:c \leq n} \frac{(\bar{n}(1-\rho_0))^c}{c!} \frac{(\bar{n}\rho_0 \mathbf{Q}_0)^{n-c}}{(n-c)!}.
\end{aligned}$$

By defining $m = n - c$ and using matrix exponential, we can further simplify the above formula as

$$\begin{aligned}
\tilde{\mathbf{\Lambda}}_{\bar{n}}^{(0)} &= e^{-\bar{n}} \mathbf{e}_0 \sum_c \frac{(\bar{n}(1-\rho_0))^c}{c!} \sum_m \frac{(\bar{n}\rho_0 \mathbf{Q}_0)^m}{m!} \\
&= e^{-\bar{n}} \mathbf{e}_0 \exp(\bar{n}(1-\rho_0)) \exp(\bar{n}\rho_0 \mathbf{Q}_0) \\
&= \mathbf{e}_0 \exp(-\bar{n}\rho_0) \exp(\bar{n}\rho_0 \mathbf{Q}_0) \\
&= \mathbf{e}_0 \exp(\bar{n}\rho_0(\mathbf{Q}_0 - \mathbf{I})),
\end{aligned} \tag{7.19}$$

where the last equality is obtained using the fact that $\exp(\mathbf{A})\exp(\mathbf{B}) = \exp(\mathbf{A}+\mathbf{B})$ whenever $\mathbf{AB} = \mathbf{BA}$. □

The formula provided in the above theorem involves only the distribution of the number of decodable input symbols at each time. In other words, for a Poisson number of batches, it is not necessary to consider the joint distribution of the number of decodable input symbols and the number of undecodable batches as in Theorem 7.1.

7.3.2 EVALUATION APPROACHES

To evaluate the formula in Theorem 7.16, we need to calculate the matrix exponential efficiently, which has been extensively studied (see [53] for a survey). We will discuss two approaches for evaluating the formula in Theorem 7.16. One of the widely used approaches for calculating matrix exponential is the scaling and squaring method [20], which has been implemented in many numerical computing environments (e.g., the **expm** function in Matlab). For a square

matrix $\mathbf{A}$, the computational cost of the algorithm in [20] for computing $\exp(\mathbf{A})$ is $O(\log \|\mathbf{A}\|_1)$ matrix multiplications (of size $\mathbf{A}$) with the truncation error no larger than a specified tolerance (e.g., the unit roundoff or 2^{-32}). Recall that the complexity for computing the quantities $\{p_{t,s}, p_t \mathbf{Q}_t\}_{0 \le t \le K, 0 \le s \le M}$ is $\mathrm{O}(K^2 M^2)$. Since each row of the matrix $\mathbf{Q}_t$ has at most $M+1$ nonzero entries, the computational cost of the algorithm in [20] for computing $\exp(\bar{n}\, p_t(\mathbf{Q}_t - \mathbf{I}))$ is $\mathrm{O}(KM \log \bar{n})$. Taking into account of the vector-matrix multiplication, the overall complexity for computing $\tilde{P}_{\text{stop}}(t|\bar{n})$, $t = 0, 1, \ldots, K$ is $\mathrm{O}(K^2M^2 + K^2 M \log \bar{n} + K^3)$ real number operations.

Now we discuss another approach. What we are calculating in (7.18) is a vector multiplying the matrix exponential, also called an action of the matrix exponential. In general, for a row vector $\mathbf{v}$ and a square matrix $\mathbf{A}$, the computation of $\mathbf{v}\exp(\mathbf{A})$ can be done by $O(\|\mathbf{A}\|_1)$ multiplications of a vector with matrix $\mathbf{A}$, using the algorithm in [3]. So for our case, the overall complexity for computing $\tilde{P}_{\text{stop}}(t|\bar{n})$, $t = 0, 1, \ldots, K$ is $\mathrm{O}(K^2M^2 + K^2 M \bar{n})$ real number operations, taking the structure of $\mathbf{Q}_t$ into consideration. When $\bar{n}$ is relatively small, we would prefer the approach using the action of the matrix exponential, while when $\bar{n}$ is large, we would choose the first approach to compute the matrix exponential directly.

We may want to evaluate $\tilde{P}_{\text{stop}}(t|\bar{n})$ for $\bar{n} \in \{i\bar{n}_0 : i = 1, \ldots, i_{\max}\}$, where $\bar{n}_0$ is a small number (e.g., 1 or 0.5). In this case, we calculate the matrix exponential $\exp(\bar{n}_0 p_t(\mathbf{Q}_t - \mathbf{I}))$ directly with complexity $\mathrm{O}(KM)$ using the algorithm in [20]. Then, we calculate $\exp(i\bar{n}_0 p_t(\mathbf{Q}_t - \mathbf{I}))$ for $i = 1, \ldots, i_{\max}$ recursively using

$$\exp(i\bar{n}_0 p_t(\mathbf{Q}_t - \mathbf{I})) = (\exp(\bar{n}_0 p_t(\mathbf{Q}_t - \mathbf{I})))^i .$$

The overall complexity for computing $\tilde{P}_{\text{stop}}(t|\bar{n})$, $t = 0, 1, \ldots, K$, $\bar{n} \in \{i\bar{n}_0 : i = 1, \ldots, i_{\max}\}$ is $\mathrm{O}(K^2M^2 + K^3 i_{\max})$ real number operations.

7.3.3 ERROR PROBABILITY AND EXPONENT

Similar to Theorem 7.12, we have the following characterization of $\tilde{P}_{\text{stop}}(t|\bar{n})$. Recall r_{BP} defined above Lemma 7.3, and q_t defined in (7.10).

Theorem 7.17 *Suppose a batch is decodable at time* 0 *with positive probability. We have:*

1. $\tilde{P}_{\text{stop}}(0|\bar{n}) = \exp(-\bar{n}(1 - q_0))$;
2. *for* $1 \le t < r_{\text{BP}}$, $\tilde{P}_{\text{stop}}(t|\bar{n}) = 0$; *and*
3. *for* $t \ge r_{\text{BP}}$,

$$\lim_{\bar{n}\to\infty} \frac{-\log \tilde{P}_{\text{stop}}(t|\bar{n})}{\bar{n}} = 1 - q_t .$$

Proof. Using Theorem 7.11 and (7.17), we obtain

$$\begin{aligned}\tilde{P}_{\text{stop}}(t|\bar{n}) &= \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0] \sum_n \frac{\bar{n}^n}{n!} e^{-\bar{n}} \left(1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0]\right)^n \\ &= \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0] \exp\left(-\bar{n}\left(\sum_{\tau=0}^{t} p_\tau - \mathbf{\Delta}_{t,i}[0,0]\right)\right).\end{aligned}$$

The proof then follows similarly as the one of Theorem 7.12 and the details are left to Appendix D.4. □

Let $\tilde{P}_{\text{err}}(\bar{n}) \triangleq 1 - \tilde{P}_{\text{stop}}(K|\bar{n})$, i.e., the probability that $\text{BP}(\tilde{N})$ cannot recover all the input packets. Recall that $q^* = \vee_{t=0}^{K-1} q_t$.

Corollary 7.18

$$\lim_{\bar{n}\to\infty} \frac{-\log \tilde{P}_{\text{err}}(\bar{n})}{n} = 1 - q^*.$$

Proof. The proof is similar to that of Corollary 7.14 except that Theorem 7.17 instead of Theorem 7.12 is applied, and hence it is omitted. □

7.3.4 ANOTHER FORMULA FOR $\mathbb{E}[N_{\text{BP}^*}]$

We can use $\tilde{P}_{\text{err}}(\bar{n})$ to characterize $\mathbb{E}[N_{\text{BP}^*}]$, the expected number of batches consumed by BP^*.

Theorem 7.19

$$\mathbb{E}[N_{\text{BP}^*}] = \int_0^\infty \tilde{P}_{\text{err}}(x)\, dx = \sum_{t=0}^{K-1} \int_0^\infty \tilde{\mathbf{\Lambda}}_x^{(t)}[0]\, dx.$$

Proof. We have

$$\begin{aligned}\int_0^\infty \tilde{P}_{\text{err}}(x)\, dx &= \int_0^\infty \textstyle\sum_{t=0}^{K-1} \tilde{P}_{\text{stop}}(t|x)\, dx \\ &= \int_0^\infty \textstyle\sum_{t=0}^{K-1} \sum_n \frac{x^n}{n!} e^{-x} P_{\text{stop}}(t|n)\, dx \\ &= \int_0^\infty \sum_n \frac{x^n}{n!} e^{-x} P_{\text{err}}(n)\, dx \\ &= \sum_n \frac{P_{\text{err}}(n)}{n!} \int_0^\infty x^n e^{-x}\, dx \\ &= \sum_n P_{\text{err}}(n) = \mathbb{E}[N_{\text{BP}^*}],\end{aligned}$$

where the change of the order of the integral and the infinite sum follows from the monotone convergence theorem and the second last step follows because the integral is the Gamma function of order $n+1$ and is equal to $n!$. □

Compared with the formulae for $\mathbb{E}[N_{\mathrm{BP}^*}]$ in (7.13) in the form of a summation, the formula here is in the form of an integration. When $\tilde{P}_{\mathrm{err}}(\bar{n})$ is easier to obtain than $P_{\mathrm{err}}(n)$, the new formula may have certain advantage for numerical evaluation.

Checking the proof of the above theorem, we see that the equivalence of these two formulae depends only on the properties of the Poisson distribution, but not on the underlaying distribution of N_{BP^*}. In general, let b_n be an infinite sequence such that $b_n \geq 0$ and $\sum_{n=0}^{\infty} b_n$ exists. Define $\tilde{b}(x) = \sum_n \frac{x^n e^{-x}}{n!} b_n$. Then we have

$$\int_0^\infty \tilde{b}(x)\,dx = \int_0^\infty \sum_n \frac{x^n e^{-x}}{n!} b_n\,dx = \sum_n \frac{b_n}{n!} \int_0^\infty x^n e^{-x}\,dx = \sum_n \frac{b_n}{n!} n! = \sum_n b_n.$$

7.4 FINITE-LENGTH DEGREE-DISTRIBUTION OPTIMIZATION

In this section, we demonstrate how to use the formulae to optimize the degree distribution for finite block lengths.

7.4.1 A GENERAL FRAMEWORK

Let first discuss a general framework for optimizing the degree distributions for finite-length BATS/fountain codes, which has two steps in each iteration with an initial degree distribution $\mathbf{\Psi}^{(0)}$. For the i-th iteration, $i = 0, 1, \ldots,$

1. find one or multiple new degree distributions which may be potentially better than $\mathbf{\Psi}^{(i)}$; and
2. evaluate the BP decoding performance of these new degree distributions in terms of an objective function, and select the degree distribution that outperforms $\mathbf{\Psi}^{(i)}$ the most as $\mathbf{\Psi}^{(i+1)}$.

The above framework has been used in the design of LT/Raptor codes. For example, in the design of finite-length Raptor codes discussed in [68], the first step is achieved by a heuristic bound on the input ripple size, and the second step is performed by means of the exact calculation of the error probability. In one of the optimizations performed in [8], a robust soliton distribution is sampled at the first step, and a heuristic formula of the expected number of inactivation is evaluated at the second step.

We adopt this framework to optimize the degree distribution of a finite-length BATS code. We can use $P_{\mathrm{err}}(n)$, $\mathbb{E}[N_{\mathrm{BP}^*}]$ or other characteristics as the objective function. The choice of the objective function $f(\mathbf{\Psi})$ will be discussed in the next subsection. Here we first introduce

an approach to update the degree distribution in each iteration. The initial degree distribution $\boldsymbol{\Psi}^{(0)} = \boldsymbol{\Psi}^{\text{asy}}$ is obtained from the asymptotic analysis (see Chapter 6). In each iteration, a new degree distribution $\boldsymbol{\Psi}^{(i+1)}$ is obtained:

$$\boldsymbol{\Psi}^{(i+1)} = \frac{\boldsymbol{\Psi}^{(i)} + \delta_i \mathbf{e}_{d_i - 1}}{1 + \delta_i}, \tag{7.20}$$

where $d_i \in \{1, 2, \ldots, K\}$, $\delta_i \in [-\boldsymbol{\Psi}^{(i)}[d_i - 1], \infty)$, and $\mathbf{e}_{d-1}$ is the all-zero vector except that the $(d-1)$-th component is 1. We try to pick d_i and δ_i such that $\boldsymbol{\Psi}^{(i+1)}$ reduces the objective function by the largest amount, which depends on the evaluation of our iterative formulae.

In our experience, this approach converges to a degree distribution that is significantly better than $\boldsymbol{\Psi}^{\text{asy}}$ after several iterations. Since our objective functions are not convex in general, we cannot guarantee the convergence to the global minimum. Moreover, our approach is a version of *coordinate descent* that selects the "best" coordinate (degree) that can reduce the objective function by the largest amount to update. In some other coordinate descent algorithms, the coordinate (degree) is selected sequentially or randomly. In practice, our approach in (7.20) converges much faster than random or sequential selection of degrees, with the computational cost of selecting the "best" coordinate taken into consideration. As discussed in [57], it may be a general phenomenon that coordinate descent converges faster by selecting the "best" coordinate than random selection.

Note that our purpose here is to illustrate the applications of the formulae obtained in this monograph, but not to propose an optimization approach for practical use. How to optimize the degree distribution for practical applications is beyond the scope of this monograph.

7.4.2 CHOICE OF THE OBJECTIVE FUNCTION

Suppose we want to have a degree distribution that has a smaller expected coding overhead than $\boldsymbol{\Psi}^{\text{asy}}$. To compare the two degree distributions in the second step of our optimization framework, we may use (7.15) to evaluate $\mathbb{E}[N_{\text{BP}^*}]$ which is accurate enough if a large value of n_2 is used. But it is indeed not necessary to evaluate $\mathbb{E}[N_{\text{BP}^*}]$ accurately in the second step. To make the evaluation in the second step faster, we instead use $\tilde{P}_{\text{err}}(\bar{n})$ (with a properly chosen value of $\bar{n}$) as a proxy of $\mathbb{E}[N_{\text{BP}^*}]$.

As hinted by Proposition 7.20 and observed in numerical evaluations, $P_{\text{err}}(n)$ is very close to 1 when $n < K/\bar{h}$. So we have the approximation that

$$\mathbb{E}[N_{\text{BP}^*}] \gtrapprox n_1 + \sum_{n=n_1}^{n_2} P_{\text{err}}(n),$$

where $n_1 = \lceil K / \sum_i h_i \rceil$ and n_2 is a sufficiently large integer. We only need to pick n_2 such that $\sum_{n=n_2+1}^{\infty} P_{\text{err}}(n)$ is sufficiently small for the desired degree distributions. For other degree distributions such that $\sum_{n=n_2+1}^{\infty} P_{\text{err}}(n)$ is large, the above approximation is roughly a lower

bound on the expected coding overhead, which is sufficient for our purpose of comparison. Similarly, we have the approximation

$$\tilde{P}_{\text{err}}(\bar{n}) \gtrapprox \sum_{n=0}^{n_1-1} \frac{\bar{n}^n e^{-\bar{n}}}{n!} + \sum_{n=n_1}^{n_2} \frac{\bar{n}^n e^{-\bar{n}}}{n!} P_{\text{err}}(n).$$

The first terms in the approximations of $\mathbb{E}[N_{\text{BP}*}]$ and $\tilde{P}_{\text{err}}(\bar{n})$ are constants. Since the p.m.f. of the Poisson distribution exhibits relatively small changes for the probability masses around its expectation, we can choose $\bar{n} = (n_1 + n_2)/2$ and expect that $\mathbb{E}[N_{\text{BP}*}]$ and $\tilde{P}_{\text{err}}(\bar{n})$ share a similar trend when the degree distribution changes.

7.4.3 EVALUATIONS FOR BP DECODING

We use an example to demonstrate the evaluation results of the formulae in this section. Consider a BATS code with $K = 256$, $q = 256$, $M = 16$ and the rank distribution in Table 7.1. The rank distribution is the one of the length-2 homogeneous line network with link erasure probability 0.2 (see [97, Section VII-A] for a formula for the rank distribution). Here $\bar{h} = 11.91$ is an upper bound on the achievable rates of BATS codes (in terms of packet per batch).

Table 7.1: The rank distribution for the evaluation examples. Here the BATS code has $q = 256$ and $M = 16$. The value of h_0 is 0 and is omitted in the table.

h_1	h_2	h_3	h_4	h_5	h_6	h_7	h_8
0	0	0	0	0.0001	0.0004	0.0025	0.0110
h_9	h_{10}	h_{11}	h_{12}	h_{13}	h_{14}	h_{15}	h_{16}
0.0387	0.1040	0.2062	0.2797	0.2339	0.1038	0.0190	0.0008

We obtain three degree distributions $\mathbf{\Psi}^{\text{asy}}$, $\mathbf{\Psi}^{\text{BP}}$, and $\mathbf{\Psi}^{\text{mee}}$ for this BATS code, which are given in Table 7.2.

- $\mathbf{\Psi}^{\text{asy}}$ is obtained by solving the degree-distribution optimization problem induced by the asymptotic analysis of BATS code in [97].
- $\mathbf{\Psi}^{\text{mee}}$ is obtained by solving (7.11), which maximizes the BP error exponent.
- $\mathbf{\Psi}^{\text{BP}}$ is obtained using the finite-length degree distribution optimization approach described in Section 7.4.1 and 7.4.2, with $\mathbf{\Psi}^{\text{asy}}$ as the initial degree distribution and $\tilde{P}_{\text{err}}(40)$ as the objective function.

From Table 7.2, we first observe that all these degree distributions are very sparse in the sense that most of the degrees have zero probability. Moreover, the supports of these three degree distributions largely overlap with each other. For the three degree distributions, we compare

them in terms of the average degree, BP error exponent, $\mathbb{E}[N_{\mathrm{BP}^*}]$ and $\mathbb{E}[\mathrm{CO}]$ in Table 7.3, and we also evaluate the error probability of $\mathrm{BP}(n)$, $n = 1, \ldots, 200$ (see Figure 7.1)

Table 7.2: Degree distributions for the rank distribution in Table 7.1. For the first three degree distributions, we give the values of the same set of probability masses, that include all the positive probability masses of these distributions. For the forth degree distribution, only the positive probability masses are listed.

Ψ_{11}^{asy}	Ψ_{12}^{asy}	Ψ_{13}^{asy}	Ψ_{14}^{asy}	Ψ_{15}^{asy}	Ψ_{20}^{asy}	Ψ_{21}^{asy}	Ψ_{26}^{asy}	Ψ_{27}^{asy}
0	0	0	0.0467	0.2502	0.1079	0.0781	0	0.0350
Ψ_{28}^{asy}	Ψ_{37}^{asy}	Ψ_{38}^{asy}	Ψ_{50}^{asy}	Ψ_{51}^{asy}	Ψ_{72}^{asy}	$\Psi_{116}^{\mathrm{asy}}$	$\Psi_{117}^{\mathrm{asy}}$	$\Psi_{256}^{\mathrm{asy}}$
0.0968	0.0728	0.0199	0.0676	0.0087	0.0679	0.0277	0.0312	0.0896

(a) Ψ^{asy}: the degree distribution obtained using the asymptotic analysis.

Ψ_{11}^{BP}	Ψ_{12}^{BP}	Ψ_{13}^{BP}	Ψ_{14}^{BP}	Ψ_{15}^{BP}	Ψ_{20}^{BP}	Ψ_{21}^{BP}	Ψ_{26}^{BP}	Ψ_{27}^{BP}
0.0826	0.0734	0.0550	0.0429	0.1745	0.0348	0.0809	0	0.0321
Ψ_{28}^{BP}	Ψ_{37}^{BP}	Ψ_{38}^{BP}	Ψ_{50}^{BP}	Ψ_{51}^{BP}	Ψ_{72}^{BP}	Ψ_{116}^{BP}	Ψ_{117}^{BP}	Ψ_{256}^{BP}
0.0888	0.0484	0.0183	0.0620	0.0080	0.0623	0.0254	0.0286	0.0822

(b) Ψ^{BP}: the degree distribution obtained by modifying Ψ^{asy} using the approach introduced in Section 7.4.1 for BP coding.

Ψ_{10}^{mee}	Ψ_{70}^{mee}	Ψ_{71}^{mee}	$\Psi_{110}^{\mathrm{mee}}$	$\Psi_{115}^{\mathrm{mee}}$	$\Psi_{165}^{\mathrm{mee}}$	$\Psi_{265}^{\mathrm{mee}}$
0.4584	0.0063	0.0706	0.0112	0.0650	0.0751	0.3135

(c) Ψ^{mee}: the degree distribution that maximizes the asymptotic decrease rate of error probability (and the number of inactivations) obtained by solving (7.11).

Table 7.3: Performance comparison of the three degree distributions given in Table 7.2

Degree Distribution	Average Degree	$\mathrm{EE}_{\mathrm{BP}}$	$\mathrm{E}[N_{\mathrm{BP}^*}]$	E[CO]
Ψ^{asy}	53.8	0.0107	>97	>898
Ψ^{BP}	49.3	0.1475	32.0	125.6
Ψ^{mee}	111.1	0.5692	82.5	727.1

We first observe that for all the degree distributions, the error probability decreases exponentially fast in n when n is large, which matches the findings in Section 7.2.3. For $\boldsymbol{\Psi}^{\mathrm{mee}}$,

the BP error decrease rate is the fastest asymptotically among these three distributions. We also observe that the error probability is almost one for small n. For a general degree distribution, the error probability for all $n < K/\bar{h}$ can be lower bounded as follows.

Proposition 7.20 *For any $n < K/\bar{h}$,*

$$P_{\text{err}}(n) \geq 1 - \exp\left(-\frac{1}{3}\left(\frac{K}{n\bar{h}} - 1\right)^2 \frac{\bar{h}}{M} n\right).$$

Proof. We have

$$P_{\text{err}}(n) \geq \Pr\left\{\sum_{i=1}^{n} \text{rk}(\mathbf{H}_i) < K\right\} = 1 - \Pr\left\{\sum_{i=1}^{n} \text{rk}(\mathbf{H}_i) \geq K\right\},$$

where $\text{rk}(\mathbf{H}_i), i = 1, \ldots, n$ are independent random variables with generic distribution $\mathbf{h}$. The proof is an application of the Chernoff bound. □

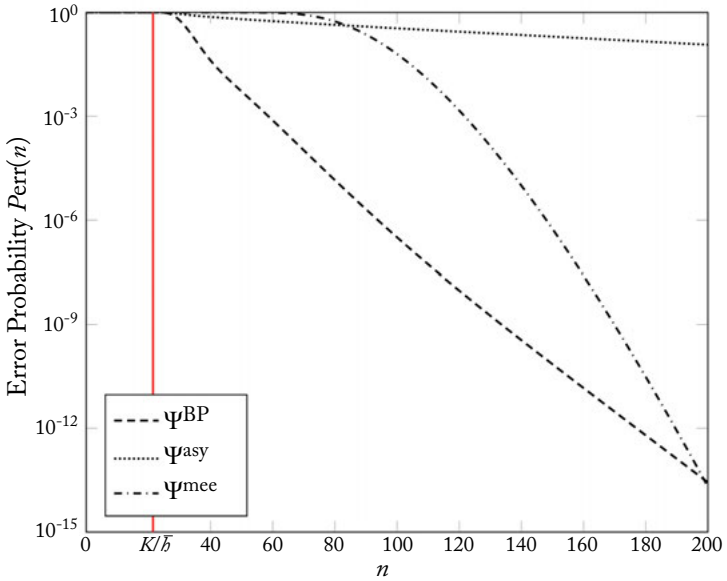
When K is sufficiently large, the above lower bound is close to 1. For relatively small values of k, the lower bound is loose. For this example, $K/\bar{h} = 21.49$. The bound in the above proposition gives $P_{\text{err}}(21) \geq 0.0029$, but our evaluations show that $P_{\text{err}}(21) = 1.0$ for all the three degree distributions.

From Figure 7.1b, we observe that $\boldsymbol{\Psi}^{\text{BP}}$ has the lowest error probability for n from 25 to 50. For example, if we want to achieve an error probability 0.01, it is sufficient to use $n = 47$ for $\boldsymbol{\Psi}^{\text{BP}}$. Unless we desire an extremely low error probability, e.g., 10^{-14}, $\boldsymbol{\Psi}^{\text{BP}}$ is preferred for BP decoding. It is not surprising that the degree distribution obtained from the asymptotic analysis does not perform well for short block lengths.

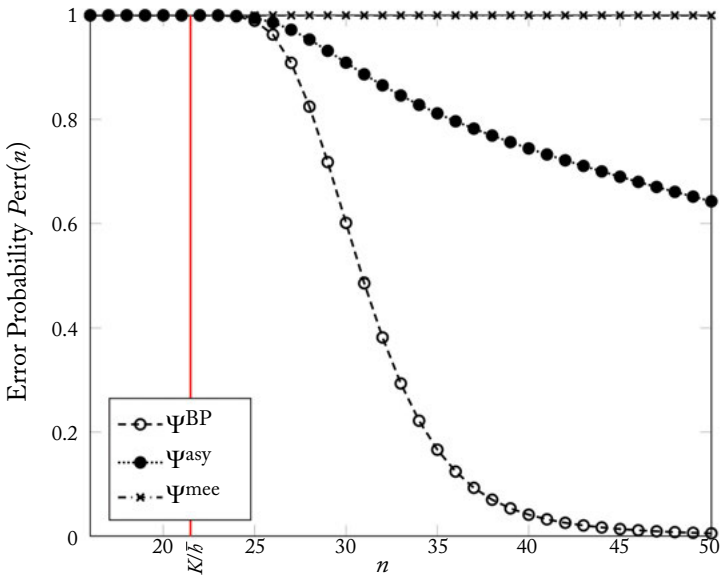
The BP error exponents of the three degree distributions are given in Table 7.3. Actually, $\boldsymbol{\Psi}^{\text{mee}}$ is the degree distribution that achieves the optimal value of (7.11) for $K = 256$, $q = 256$ and the rank distribution in Table 7.1.

The values of $\mathbb{E}[N_{\text{BP}^*}]$ and the expected coding overhead of the three degree distributions can be found using the approximation in (7.15). The trend of $\sum_{n=0}^{n_2} P_{\text{err}}(n)$ when n_2 increases can be found in Figure 7.2. We see that for both $\boldsymbol{\Psi}^{\text{BP}}$ and $\boldsymbol{\Psi}^{\text{mee}}$, the approximation converges fast due to the fast decrease of the corresponding error probability $P_{\text{err}}(n)$. For the range of n_2 in the evaluation, the value of $\sum_{n=0}^{n_2} P_{\text{err}}(n)$ does not converge for $\boldsymbol{\Psi}^{\text{asy}}$. But the value of $\sum_{n=0}^{n_2} P_{\text{err}}(n)$ for $\boldsymbol{\Psi}^{\text{asy}}$ provides a lower bound for $\mathbb{E}[N_{\text{BP}^*}]$ that is sufficient for us to compare these three degree distributions in terms of $\mathbb{E}[N_{\text{BP}^*}]$.

We also evaluate $\tilde{P}_{\text{err}}(\bar{n})$ for the degree distribution $\boldsymbol{\Psi}^{\text{BP}}$ and compare it with $P_{\text{err}}(n)$. From the illustration in Figure 7.3, we first observe that the two curves are similar except for the



(a) n from 1 to 200



(b) Zoom-in for n from 16 to 50

Figure 7.1: $P_{\text{err}}(n)$ for different degree distributions. Here $K = 256$, $q = 256$ and the rank distribution is given in Table 7.1. For this example, $K/\bar{h} = 21.49$, which is illustrated by the vertical line marked by $K/\bar{h}$ in the figures.

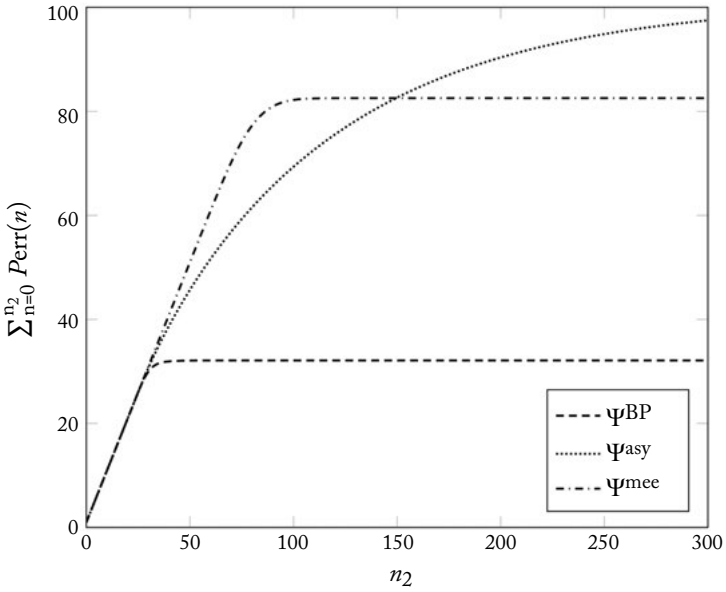
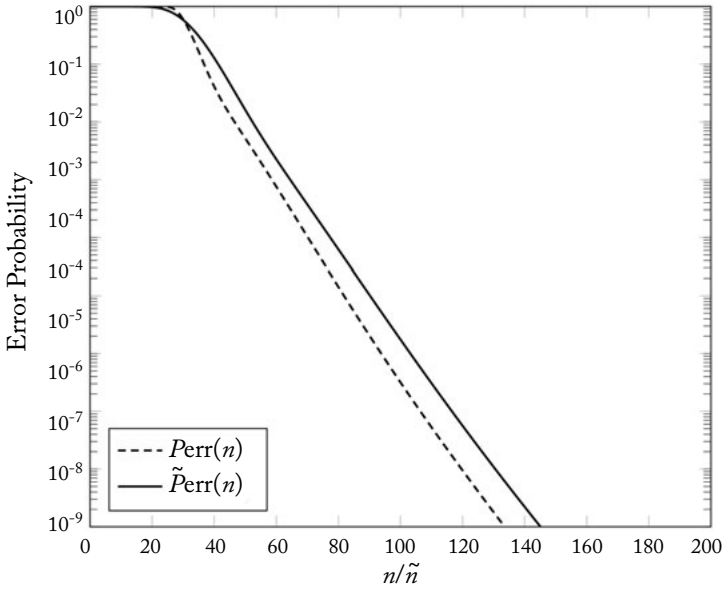


Figure 7.2: The trends of $\sum_{n=0}^{n_2} P_{\text{err}}(n)$ when n_2 increases for the three degree distributions given in Table 7.2.

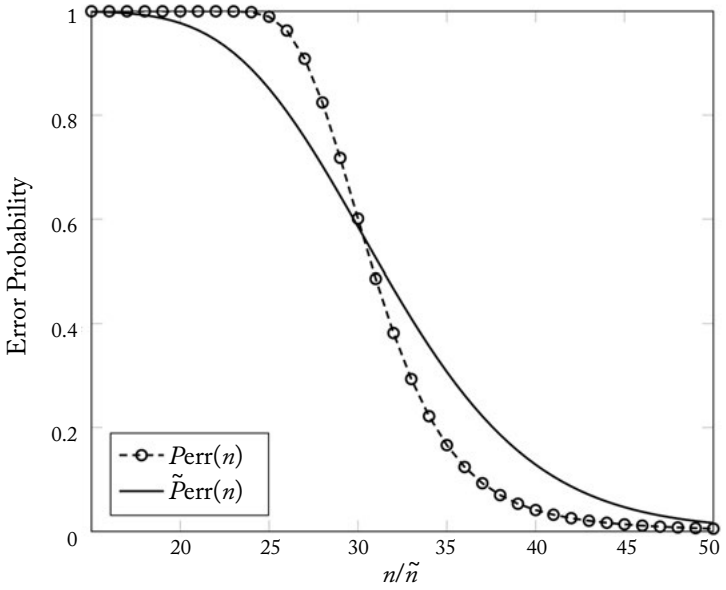
different decrease rates. $\tilde{P}_{\text{err}}(\bar{n})$ decreases slightly slower than $P_{\text{err}}(n)$ which is consistent with our characterization that

$$\lim_{n\to\infty} \frac{-\log(P_{\text{err}}(n))}{n} = -\log q^* \geq 1 - q^* = \lim_{\bar{n}\to\infty} \frac{-\log \tilde{P}_{\text{err}}(\bar{n})}{\bar{n}}.$$

Further, from the two formulae for $\mathbb{E}[N_{\text{BP}^*}]$ in terms of $P_{\text{err}}(n)$ and $\tilde{P}_{\text{err}}(\bar{n})$, respectively, we know that the areas below the two curves in Figure 7.3 are both $\mathbb{E}[N_{\text{BP}^*}]$.



(a) n from 1 to 200



(b) Zoom-in for n from 15 to 50

Figure 7.3: Comparison of $\tilde{P}_{\mathrm{err}}$ and P_{err} for degree distribution $\boldsymbol{\Psi}^{\mathrm{BP}}$. Here $K = 256$, $q = 256$ and the rank distribution is given in Table 7.1.

CHAPTER 8

Inactivation Decoding

The decoding of a BATS code is essentially solving a linear system of equations. Although Gaussian elimination can guarantee the success when the linear system of equations is uniquely solvable, its computational cost is high even when the number of input packet is not very large. In this chapter, we study *inactivation decoding*, which combines Gaussina elimination with BP decoding, can reduce the coding overhead compared with using BP decoding alone.

8.1 INTRODUCTION OF INACTIVATION DECODING

Inactivation decoding was proposed for LT/Raptor codes [67, 69] and can be regarded as an efficient way to solve sparse linear systems [34, 59], and a similar algorithm [63] has been used for efficient encoding of LDPC codes. Here we describe how to use inactivation decoding for BATS codes.

In the BP decoding algorithm discussed in Section 2.3, the decoding stops when no decodable input symbols remain. Although BP decoding stops, Gaussian elimination can still be used to decode the remaining input symbols (by combining the linear systems associated with the undecoded batches to a single linear system involving all the undecoded input symbols). However, the decoding complexity of Gaussian elimination is much higher than that of BP decoding. Inactivation decoding is a technique that efficiently combines BP decoding and Gaussian elimination.

We first describe an inactivation decoding process for a given number n of batches, denoted by INAC(n). The decoding of INAC(n) is the same BP(n) until there are no more decodable symbols. Instead of stopping the decoding as in BP(n), INAC(n) tries to resume the BP decoding process by "inactivating" certain undecoded input symbols. Specifically, suppose there are no decodable input symbols at time t. Then INAC(n) randomly picks an undecoded symbol b and marks it as *inactive*. The decoder substitutes this inactive symbol b into the batches like a decoded symbol, except that b is an indeterminate. The time index is then increased by one. For example, if the k-th input symbol b_k is inactivated at time t and $k \in A_i^{(t)}$, each component of $\mathbf{Y}_i^{(t+1)} = \mathbf{Y}_i^{(t)} - b_k g \mathbf{H}_i$ will be expressed as a linear polynomial in b_k. Since the time index is increased by one for each input symbol decoded or inactivated, the decoding process of INAC(n) is repeated until time K when all the input symbols are either decoded or inactivated.

Denote by I the number of inactive symbols when INAC(n) stops, and denote by $b_1, \ldots, b_I$ the inactive input symbols. A decoded input symbol b can now be expressed as

$$b = \sum_{i=1}^{I} \alpha_i b_i + \alpha_0,$$

where α_i $(0 \leq i \leq I)$ are determined by the decoding process. Note that for an input symbol b decoded before BP(n) stops, the coefficients α_i for all i are equal to 0, i.e., the value of b is α_0. Therefore, inactivation decoding recovers each decoded input symbol in the form of a linear formula involving the inactive symbols.

After INAC(n) stops, we need to recover the inactive symbols and substitute their values into the formulae of the decoded input symbols. To generate $K - I$ decoded input symbols, the decoder consumes $K - I$ of all the received symbols. The other received symbols are actually transformed into linear equations of the inactive symbols, and then used to solve the inactive symbols. For example, if all the input symbols of a batch is decoded (in terms of the inactive symbols), the received symbols of this batch cannot be used to decode more input symbols, but they impose linear constraints on the inactive symbols. Usually, this linear system of inactive symbols are solved by Gaussian elimination.

The inactive symbols are uniquely solvable if and only if the (global) linear system formed by the linear systems associated with all the batches is uniquely solvable. When being used with the precoding techniques of *high-density parity-check* and *per-inactivation*, the decoding of the inactive symbols can be successful with high probability for a small coding overhead. This will be discussed in Section 8.3.

8.2 FINITE-LENGTH ANALYSIS OF INACTIVATION DECODING

Inactivation decoding incurs extra computation cost that includes solving the inactive symbols using Gaussian elimination and substituting the values of the inactive symbols. Since both the former and the latter depend on the number of inactive symbols, knowing this number can help us understand the tradeoff between computation cost and coding rate. In this section, we provide methods for computing the expected number of inactive symbols, first given in [54, 96, 102]. The analysis to be presented is not associated with any specific precoding technique. Omitted proofs can be found in Appendix E.

8.2.1 EXPECTED NUMBER OF INACTIVATION

Since the inactive input symbols are treated as decoded during the inactivation decoding, the decodability of batches can be defined in the same way as for BP decoding. Let $\hat{R}_n^{(t)}$ and $\hat{C}_n^{(t)}$ be the number of decodable input symbols and the number of undecodable batches, respectively, at time t when using INAC(n). From the description of inactivation decoding, the probability

that a symbol is inactivated at time $t < K$ is

$$P_{\text{inac}}(t|n) \triangleq \Pr\{\hat{R}_n^{(t)} = 0\}. \tag{8.1}$$

At time K, the decoding stops (all the input symbols are either decoded or inactive). The expectation of the number of inactive symbols can be expressed as

$$\mathbb{E}[I|n] = \sum_{t=0}^{K-1} P_{\text{inac}}(t|n).$$

Define an $(n+1) \times (K-t+1)$ matrix $\mathbf{\Gamma}_n^{(t)}$ as

$$\mathbf{\Gamma}_n^{(t)}[c, r] \triangleq \Pr\left\{\hat{C}_n^{(t)} = c, \hat{R}_n^{(t)} = r\right\}.$$

According to the definition in (8.2), we can write

$$P_{\text{inac}}(t|n) = \sum_{c=0}^{n} \mathbf{\Gamma}_n^{(t)}[c, 0]. \tag{8.2}$$

Define $\mathbf{N}_t$ as a $(K-t+2) \times (K-t+1)$ matrix of the form $\begin{bmatrix} \mathbf{e}_0 \\ \mathbf{I} \end{bmatrix}$, so that

$$\mathbf{\Gamma}_n^{(t-1)}[c, :]\mathbf{N}_t = (\mathbf{\Gamma}_n^{(t-1)}[c, 0] + \mathbf{\Gamma}_n^{(t-1)}[c, 1], \mathbf{\Gamma}_n^{(t-1)}[c, 2 : K-t+1]).$$

The following theorem provides an iterative formula for $\mathbf{\Gamma}_n^{(t)}$, $t = 0, 1, \ldots, K$.

Theorem 8.1 *Consider a BATS code with K input symbols, n batches, degree distribution $\mathbf{\Psi}$, rank distribution $\mathbf{h}$ of the transfer matrix, and batch size M. We have for inactivation decoding*

$$\mathbf{\Gamma}_n^{(0)}[c, :] = \text{Bi}(c; n, 1-\rho_0)\mathbf{e}_0\mathbf{Q}_0^{n-c}, \tag{8.3}$$

and for $t > 0$,

$$\mathbf{\Gamma}_n^{(t)}[c, :] = \sum_{c'=c}^{n} \text{Bi}(c; c', 1-\rho_t)\mathbf{\Gamma}_n^{(t-1)}[c', :]\mathbf{N}_t\mathbf{Q}_t^{c'-c}. \tag{8.4}$$

Proof. The proof is similar to that of Theorem 7.1. □

If we replace $\mathbf{N}_t$ by $\begin{bmatrix}\mathbf{0}\\ \mathbf{I}\end{bmatrix}$ of proper dimension, the above theorem becomes Theorem 7.1. Due to this similarity, many discussions about BP decoding based on Theorem 7.1 apply to inactivation decoding as well. For example, the following formula is simpler for evaluating $P_{\text{inac}}(t|n)$ for a range of n.

Theorem 8.2 *For $n \geq 0$ and $t \geq 0$,*

$$P_{\text{inac}}(t|n) = \sum_{c=0}^{n} \binom{n}{c} \left(1 - \sum_{\tau=0}^{t} p_\tau\right)^c \mathbf{\Gamma}_{n-c}^{(t)}[0,0], \tag{8.5}$$

where the first row of the matrices $\mathbf{\Gamma}_{n'}^{(t)}$, $n' = 0, 1, \ldots, n$ can be computed by the following recursion: For $n' = 0, 1, \ldots, n$,

$$\mathbf{\Gamma}_{n'}^{(0)}[0,:] = (p_0\mathbf{Q}_0)^{n'}[0,:], \tag{8.6}$$

and for $t > 0$

$$\mathbf{\Gamma}_{n'}^{(t)}[0,:] = \sum_{c=0}^{n'} \binom{n'}{c} \mathbf{\Gamma}_{n'-c}^{(t-1)}[0,:]\mathbf{N}_t(p_t\mathbf{Q}_t)^c. \tag{8.7}$$

Proof. The proof is similar to that of Theorem 7.8. □

The formula in the above theorem can be evaluated in a way similar to the one in Theorem 7.8. Similar to $P_{\text{stop}}(t|n)$, $P_{\text{inac}}(t|n)$ can also be expressed as the linear combination of 2^t n-th powers.

Theorem 8.3 *For $n \geq 0$ and $t \geq 0$,*

$$P_{\text{inac}}(t|n) = \sum_{i=0}^{2^t-1} \mathbf{V}'_{t,i}[0] \left(1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0]\right)^n,$$

where matrix $\mathbf{\Delta}_{t,i}$ is defined in Theorem 7.11, and row vector $\mathbf{V}'_{t,i}$ is defined as follows.

1. $\mathbf{V}'_{0,0} \triangleq \mathbf{U}_0[0,:]$.
2. *For $t \geq 0$ and $i = 0, 1, \ldots, 2^t - 1$,*

$$\begin{aligned}\mathbf{V}'_{t+1,i} &= \mathbf{V}'_{t,i}[1:],\\ \mathbf{V}'_{t+1,2^t+i} &= \mathbf{V}'_{t,i}[0](\mathbf{U}_{t+1}[0,:] - \mathbf{U}_t[0,1:]).\end{aligned}$$

Proof. The proof is similar to that of Theorem 7.11. □

Recall $q_t = 1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,0}[0,0]$ (see (7.10)) and the definition of r_{BP} above Lemma 7.3. Applying Theorem 8.3, we can further obtain the following asymptotic behavior of $P_{\mathrm{inac}}(t|n)$ when n is large.

Theorem 8.4 *When $t < r_{\mathrm{BP}}$, $P_{\mathrm{inac}}(t|n) = q_t^n$, and when $t \geq r_{\mathrm{BP}}$,*

$$\lim_{n\to\infty} \frac{-\log P_{\mathrm{inac}}(t|n)}{n} = -\log q_t.$$

Corollary 8.5

$$\lim_{n\to\infty} \frac{-\log \mathbb{E}[I|n]}{n} = -\log q^*,$$

where $q^* = \vee_{t=0}^{K-1} q_t$.

8.2.2 POISSON NUMBER OF BATCHES

In this subsection, we assume that the number of received batches is a Poisson distributed random variable $\tilde{N}$ with mean $\bar{n}$. Denote by $\tilde{I}$ the number of inactive symbols after $\mathrm{INAC}(\tilde{N})$ stops.

Define a row vector $\tilde{\mathbf{\Gamma}}_{\bar{n}}^{(t)}$ of size $K - t + 1$ as

$$\tilde{\mathbf{\Gamma}}_{\bar{n}}^{(t)}[r] \triangleq \Pr\left\{\hat{R}_{\tilde{N}}^{(t)} = r\right\} = \sum_n \frac{\bar{n}^n}{n!} e^{-\bar{n}} \Pr\left\{\hat{R}_n^{(t)} = r\right\}.$$

Thus,

$$\tilde{\mathbf{\Gamma}}_{\bar{n}}^{(t)} = \sum_n \frac{\bar{n}^n}{n!} e^{-\bar{n}} \sum_{c=0}^{n} \mathbf{\Gamma}_n^{(t)}[c,:]. \tag{8.8}$$

The probability that an input symbol is inactive at time t is

$$\tilde{P}_{\mathrm{inac}}(t|\bar{n}) = \tilde{\mathbf{\Gamma}}_{\bar{n}}^{(t)}[0] = \sum_n \Pr\{\tilde{N} = n\} P_{\mathrm{inac}}(t|n), \tag{8.9}$$

and hence the expected number of inactive symbols is given by

$$\mathbb{E}[\tilde{I}|\bar{n}] = \sum_{t=0}^{K-1} \tilde{P}_{\mathrm{inac}}(t|\bar{n}) = \sum_{t=0}^{K-1} \tilde{\mathbf{\Gamma}}_{\bar{n}}^{(t)}[0]. \tag{8.10}$$

The next theorem provides a formula for calculating $\tilde{\mathbf{\Gamma}}_{\bar{n}}^{(t)}$.

Theorem 8.6 *Consider inactivation decoding of a BATS code with K input symbols, degree distribution $\mathbf{\Psi}$, and transfer matrix rank distribution $\mathbf{h}$. When the number of batches used by BP decoding is Poisson distributed with expectation $\bar{n}$, for any integer $t \geq 0$*

$$\tilde{\mathbf{\Gamma}}_{\bar{n}}^{(t)} = \tilde{\mathbf{\Gamma}}_{\bar{n}}^{(t-1)} \mathbf{N}_t \exp\left(\bar{n}\, p_t (\mathbf{Q}_t - \mathbf{I})\right),$$

where $\tilde{\mathbf{\Gamma}}_{\bar{n}}^{-1} \triangleq \mathbf{e}_0$.

Proof. This theorem can be proved similarly as Theorem 7.16. □

Recall that $q_t = 1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,0}[0,0]$ (see (7.10)) and the definition of r_{BP} above Lemma 7.3.

Theorem 8.7 *When $t < r_{\text{BP}}$, $\tilde{P}_{\text{inac}}(t|n) = \exp(-\bar{n}(1-q_t))$, and when $t \geq r_{\text{BP}}$,*

$$\lim_{n\to\infty} \frac{-\log \tilde{P}_{\text{inac}}(t|\bar{n})}{n} = 1 - q_t.$$

Proof. Using Theorem 8.3 and (8.9), we get

$$\tilde{P}_{\text{inac}}(t|\bar{n}) = \sum_{i=0}^{2^t-1} \mathbf{V}'_{t,i}[0] \exp\left(-\bar{n}\left(\sum_{\tau=0}^{t} p_\tau - \mathbf{\Delta}_{t,i}[0,0]\right)\right).$$

The remainder of the proof is similar to that of Theorem 8.4. □

Corollary 8.8

$$\lim_{\bar{n}\to\infty} \frac{-\log \mathbb{E}[\tilde{I}|\bar{n}]}{\bar{n}} = 1 - q^*.$$

Table 8.1: The degree distribution $\mathbf{\Psi}^{\text{inac}}$ obtained by modifying $\mathbf{\Psi}^{\text{asy}}$ using the approach introduced in Section 7.4.1 for inactivation decoding

Ψ_{11}^{inac}	Ψ_{12}^{inac}	Ψ_{13}^{inac}	Ψ_{14}^{inac}	Ψ_{15}^{inac}	Ψ_{20}^{inac}	Ψ_{21}^{inac}	Ψ_{26}^{inac}	Ψ_{27}^{inac}
0	0.0796	0.0973	0.0414	0.2126	0.0955	0.0692	0.0088	0.0309
Ψ_{28}^{inac}	Ψ_{37}^{inac}	Ψ_{38}^{inac}	Ψ_{50}^{inac}	Ψ_{51}^{inac}	Ψ_{72}^{inac}	Ψ_{116}^{inac}	Ψ_{117}^{inac}	Ψ_{256}^{inac}
0.0857	0.0644	0.0176	0.0598	0.0077	0.0512	0.0245	0.0276	0.0262

8.2.3 EVALUATION EXAMPLES

For inactivation decoding, we want to have a degree distribution that has a smaller expected number of inactivations than $\mathbf{\Psi}^{\text{asy}}$. To optimize the degree distribution using the approach introduced in Sections 7.4.1 and 7.4.2, we use $\mathbb{E}[\tilde{I}|n]$ instead of $\mathbb{E}[I|n]$ as the objective function to reduce the evaluation time. For the example in Section 7.4.3, $\mathbf{\Psi}^{\text{inac}}$ is the degree distribution obtained using this approach, as given in Table 8.1.

We evaluate the inactivation decoding performance of the three degree distributions $\mathbf{\Psi}^{\text{asy}}$ and $\mathbf{\Psi}^{\text{mee}}$ in Table 7.2 and $\mathbf{\Psi}^{\text{inac}}$. We evaluate $\mathbb{E}[I|n]$, $n = 1, \ldots, 200$ for the three degree distributions. See Figure 8.1 for an illustration of the evaluation results. We first observe that for all the degree distributions, the expected number of inactivation decreases exponentially fast when n is large. For $\mathbf{\Psi}^{\text{mee}}$, the asymptotic decrease rate of the expected number of inactivation is the fastest among these three distributions. From Figure 7.1b, we observe that $\mathbf{\Psi}^{\text{inac}}$ has the smallest expected number of inactivation for n from 20 to 50. For example, if we use $n = 25$ for $\mathbf{\Psi}^{\text{inac}}$, the expected number of inactivation is about 17.

We also evaluate $\mathbb{E}[\tilde{I}|\bar{n}]$ and compare it with $\mathbb{E}[I|n]$ for degree distribution $\mathbf{\Psi}^{\text{inac}}$. From the illustration in Figure 8.2, we observe that the two curves are similar except for the different decrease rates. $\tilde{P}_{\text{err}}(\bar{n})$ decreases slightly slower than $P_{\text{err}}(n)$ which is consistent with our characterization that

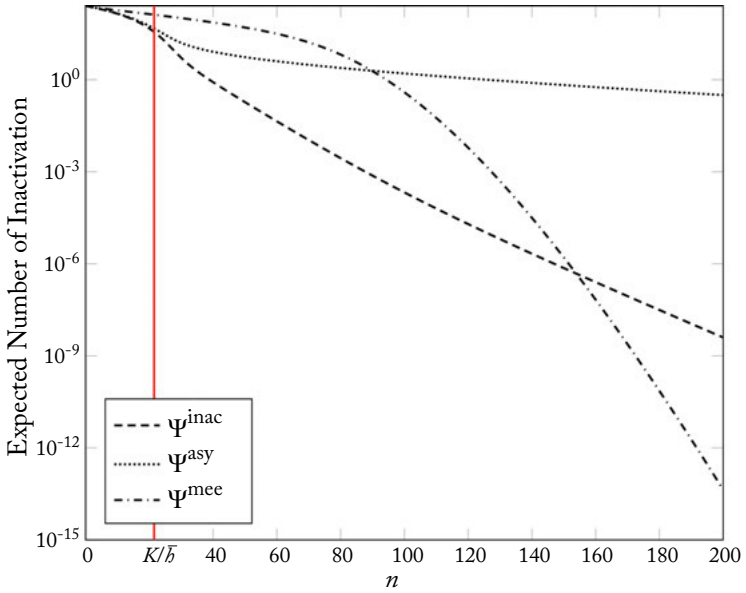
$$\lim_{n\to\infty} \frac{-\log \mathbb{E}[I|n]}{n} = -\log q^* \geq 1 - q^* = \lim_{\bar{n}\to\infty} \frac{-\log \mathbb{E}[\tilde{I}|\bar{n}]}{\bar{n}}.$$

8.3 PRACTICAL DESIGN

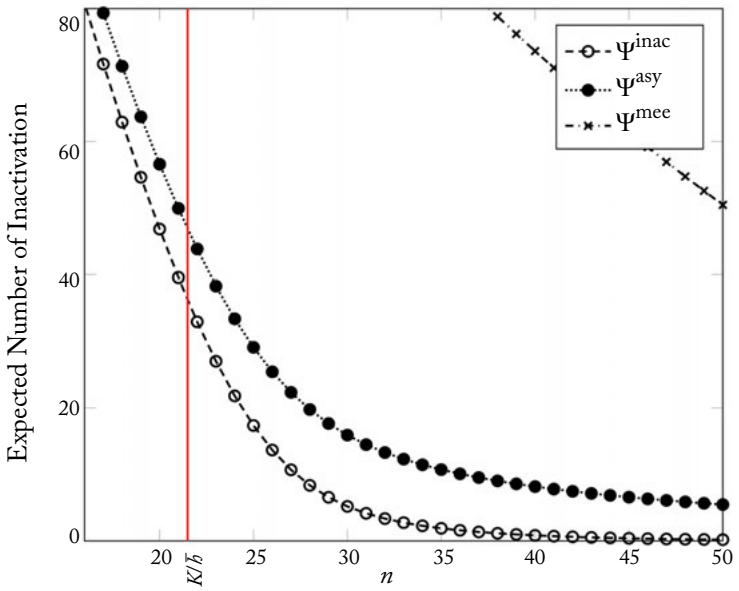
In this selection, we discuss how to design the precode and the BATS encoding to increase the probability that the inactive packets can be decoded.

8.3.1 HIGH-DENSITY PARITY CHECK AND PRE-INACTIVATION

Following the notations in Chapter 2, we formulate BATS encoding with systematic precoding. We have K' input packets given by a $T \times K'$ matrix $\mathbf{B}'$, and we use a systematic precode with

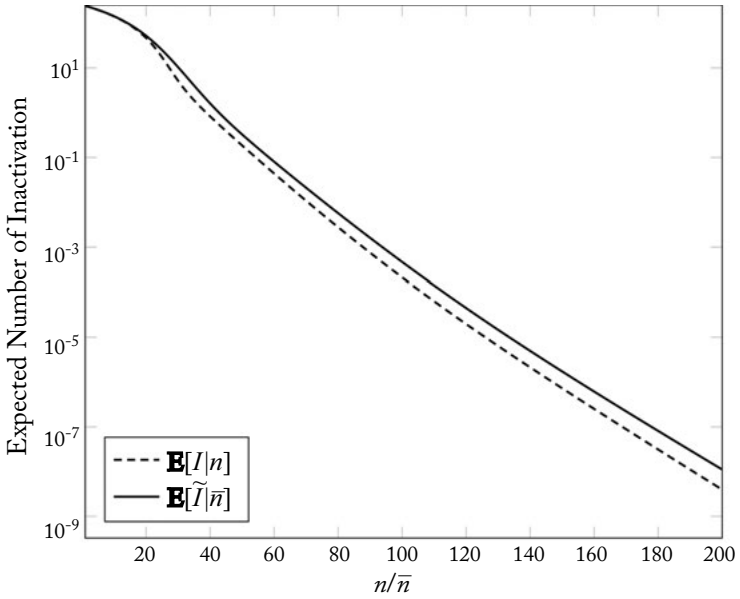


(a) n from 1 to 200

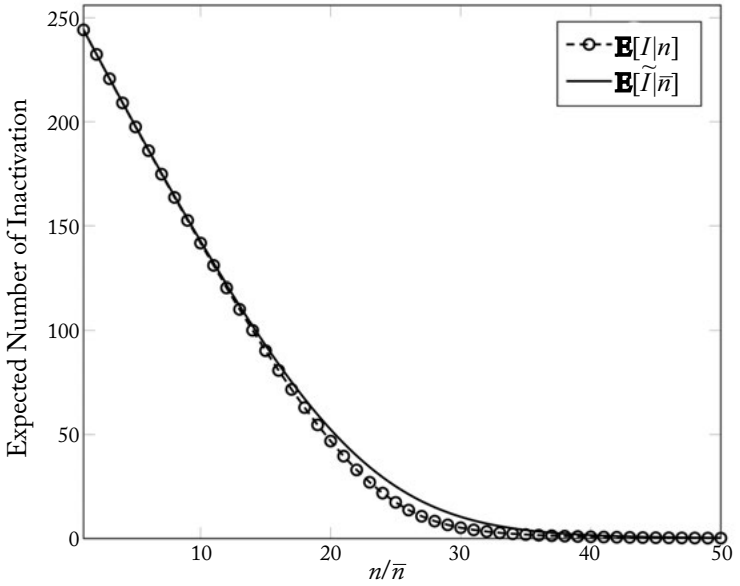


(b) Zoom-in for n from 16 to 50

Figure 8.1: Expected number of inactivation $\mathbb{E}[I|n]$ for different degree distributions. Here $K = 256$, $q = 256$ and the rank distribution is given in Table 7.1. For this example, $K/\bar{h} = 21.49$, which is illustrated by the vertical line marked by $K/\bar{h}$ in the figures.



(a) n from 1 to 200



(b) Zoom-in for n from 1 to 50

Figure 8.2: Comparison of $\mathbb{E}[I|n]$ and $\mathbb{E}[\tilde{I}|\bar{n}]$ for degree distribution $\mathbf{\Psi}^{\text{inac}}$. Here $K = 256$, $q = 256$ and the rank distribution is given in Table 7.1.

parity check matrix $\mathbf{W} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix}$, where $\mathbf{W}_1$ is a $K' \times (K - K')$ matrix and $\mathbf{W}_2$ is a $(K - K') \times (K - K')$ invertible matrix. Denoting by $\mathbf{P}$ the $K - K'$ parity-check packets, we have

$$\begin{bmatrix} \mathbf{B}' & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \end{bmatrix} = \mathbf{0},$$

and hence

$$\mathbf{P} = -\mathbf{B}'\mathbf{W}_1\mathbf{W}_2^{-1}. \tag{8.11}$$

Following the precoding, the batch encoding process is applied to the K precoded input packets $\mathbf{B} = \begin{bmatrix} \mathbf{B}' & \mathbf{P} \end{bmatrix}$ generated by the precode.

Suppose n batches are received at a destination node with $\mathbf{Y}_i$ being the received packets of batch i. We know from (2.1) that

$$\mathbf{Y}_i = \mathbf{B}_i\mathbf{G}_i\mathbf{H}_i = \mathbf{B}\tilde{\mathbf{G}}_i\mathbf{H}_i, \quad i = 1, \ldots, n,$$

where $\tilde{\mathbf{G}}_i$ a $K \times M$ matrix formed by expanding $\mathbf{G}_i$ with the all-zero rows such that $\mathbf{B}_i\mathbf{G}_i = \mathbf{B}\tilde{\mathbf{G}}_i$. Let

$$\mathbf{Z} = \begin{bmatrix} \tilde{\mathbf{G}}_1\mathbf{H}_1 & \cdots & \tilde{\mathbf{G}}_n\mathbf{H}_n \end{bmatrix}. \tag{8.12}$$

Then the BATS code decoding is to solve the following system of linear equations with $\mathbf{B}'$ as the variable

$$\begin{bmatrix} \mathbf{B}' & \mathbf{P} \end{bmatrix} \begin{bmatrix} \mathbf{Z}_1 & \mathbf{W}_1 \\ \mathbf{Z}_2 & \mathbf{W}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{Y} & \mathbf{0} \end{bmatrix}, \tag{8.13}$$

where $\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 & \cdots & \mathbf{Y}_n \end{bmatrix}$ and $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are the first K' and the last $K - K'$ rows of $\mathbf{Z}$, respectively. By (8.11), we further write (8.13) as

$$\mathbf{B}'\mathbf{A} = \mathbf{Y},$$

where

$$\mathbf{A} = \mathbf{Z}_1 - \mathbf{W}_1\mathbf{W}_2^{-1}\mathbf{Z}_2. \tag{8.14}$$

The inactivation decoding of BATS codes can be successful if and only if $\mathrm{rk}(\mathbf{A}) = K'$. A necessary condition such that $\mathrm{rk}(\mathbf{A}) = K'$ is $\sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i) \geq K'$, since otherwise $\mathrm{rk}(\mathbf{A}) \leq \mathrm{rk}(\mathbf{Z}) \leq \sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i) < K'$. Suppose the total rank of all the batches is at least $(1 + \delta)K'$, $\delta \geq 0$. We hope the probability of $\mathrm{rk}(\mathbf{A}) = K'$ is high. Write

$$\begin{aligned} &\Pr\left\{\mathrm{rk}(\mathbf{A}) = K' \,\middle|\, \sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i) \geq (1+\delta)K'\right\} \\ &= \Pr\left\{\mathrm{rk}(\mathbf{A}) = K' \,\middle|\, \mathrm{rk}(\mathbf{Z}) \geq (1+\delta)K'\right\} \Pr\left\{\mathrm{rk}(\mathbf{Z}) \geq (1+\delta)K' \,\middle|\, \sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i) \geq (1+\delta)K'\right\}. \end{aligned}$$

By examining the two terms on RHS, we can gain insight on how to to improve the probability that $\mathrm{rk}(\mathbf{A}) = K'$ through the design of the precoding and batch encoding.

High-density Parity Check

We now show that if the parity check matrix of the precoding behaves like a totally random matrix, then $\Pr\{\mathrm{rk}(\mathbf{A}) = K' | \mathrm{rk}(\mathbf{Z}) \geq (1+\delta)K'\}$ is very close to 1.

Lemma 8.9 *When* $\mathbf{W}_1\mathbf{W}_2^{-1}$ *is a totally random matrix over the base field* $\mathrm{GF}(q)$*, for* $0 \leq \delta \leq K/K' - 1$*,*

$$\Pr\left\{rk(\mathbf{A}) = K' | \, rk(\mathbf{Z}) \geq (1+\delta)K'\right\} \geq \zeta_{K-(1+\delta)K'}^{K-K'}(q).$$

Proof. We will prove that for $k \geq (1+\delta)K'$ and $k - K' \leq k_2 \leq K - K'$,

$$\begin{aligned} \Pr\left\{\mathrm{rk}(\mathbf{A}) = K' | \, \mathrm{rk}(\mathbf{Z}_2) = k_2, \mathrm{rk}(\mathbf{Z}) = k\right\} &= \zeta_{K'-k+k_2}^{k_2}(q) \\ &\geq \zeta_{k_2-\delta K'}^{k_2}(q) \\ &\geq \zeta_{K-(1+\delta)K'}^{K-K'}(q), \end{aligned} \tag{8.15}$$

which implies the lemma. The first inequality follows from $(1+\delta)K' \leq k$ and $\zeta_{r+1}^{m} < \zeta_r^m$, and the second inequality follows from $k_2 \leq K - K'$ and $\zeta_{r+1}^{r+1} < \zeta_r^r$. The equality (8.15) is verified as follows.

The row vectors of $\mathbf{Z}_2$ span a vector space $\mathcal{V}$ of dimension k_2. Assume that the first $k - k_2$ rows of $\mathbf{Z}_1$ are not in $\mathcal{V}$ and all the other rows are in $\mathcal{V}$. If the assumption does not holds, we can apply elementary row operations on both sides of (8.14) so that the transformed $\mathbf{Z}_1$ satisfies the assumption. Note that the elementary row operations preserves the total randomness of $\mathbf{W}_1\mathbf{W}_2^{-1}$.

Write $\mathbf{Z}_1 = \mathbf{Z}_1' + \mathbf{Z}_1''$ where the last $K' - k + k_2$ rows of $\mathbf{Z}_1'$ are all zero, and the first $k - k_2$ rows of $\mathbf{Z}_1''$ are all zero. Then, $\mathbf{A} = \mathbf{Z}_1' + \mathbf{Z}_2'$ where $\mathbf{Z}_2' = \mathbf{Z}_1'' - \mathbf{W}_1\mathbf{W}_2^{-1}\mathbf{Z}_2$, and $\mathbf{A}$ has a full row rank if and only if the last $K' - k + k_2$ rows of $\mathbf{Z}_2'$ are linearly independent. Since the rows of $\mathbf{Z}_2'$ are i.i.d. and uniformly distributed in $\mathcal{V}$, the probability that the $K' - k + k_2$ rows of $\mathbf{Z}_2'$ are linearly independent is $\zeta_{K'-k+k_2}^{k_2}(q)$, which proves (8.15). □

We know that

1. when K, K', and δ are constants with $K/K' \geq 1 + \delta$, $\zeta_{K-(1+\delta)K'}^{K-K'}(q) \to 1$ as $q \to \infty$; and
2. when q is a constant, and $\lim_{K'\to\infty} K/K' > 1 + \delta$, $\zeta_{K-(1+\delta)K'}^{K-K'}(q) \to 1$ as $K' \to \infty$.

For practical implementation, we can use a dense matrix to emulate a totally random matrix. However, directly solving the parity check matrix using matrix multiplication as in (8.11) has a high computation cost when $\mathbf{W}_1\mathbf{W}_2^{-1}$ is dense, and hence results in a high precoding and encoding complexity. Moreover, we would prefer a spare parity check matrix so that the BP decoding of BATS codes can be applied to the precode. Therefore, we prefer a precode with the following prescription for the parity check matrix.

1. $\mathbf{W}$ is close to be sparse so that BP decoding can be applied to solve the precode.

2. The parity check packets can be efficiently solved.

3. $\mathbf{W}_1\mathbf{W}_2^{-1}$ behaves like a totally random matrix.

An example design can be found in practical Raptor codes [69], where most of the columns of $\mathbf{W}$ are sparse but several columns of $\mathbf{W}$ are dense. The sparse columns of $\mathbf{W}$ are called LDPC and the dense columns of $\mathbf{W}$ are called *high-density parity check* (HDPC). Suppose the last h columns of $\mathbf{W}$ are dense. We write

$$\mathbf{W}_1 = \begin{bmatrix}\mathbf{S}_1 & \mathbf{D}_1\end{bmatrix},$$
$$\mathbf{W}_2 = \begin{bmatrix}\mathbf{I} & \mathbf{D}_2 \\ \mathbf{S}_2 & \mathbf{I}'\end{bmatrix},$$

where $\mathbf{D}_1$ and $\mathbf{D}_2$ have h columns, and $\mathbf{I}$ and $\mathbf{I}'$ are identity matrices. Then,

$$\mathbf{W}_1\mathbf{W}_2^{-1} = \begin{bmatrix}(\mathbf{S}_1 + \mathbf{D}_1\mathbf{S}_2)(\mathbf{D}_2\mathbf{S}_2 + \mathbf{I})^{-1} & (\mathbf{D}_1 - \mathbf{S}_1\mathbf{D}_2)(\mathbf{S}_2\mathbf{D}_2 + \mathbf{I}')^{-1}\end{bmatrix}.$$

When $\mathbf{D}_1$ is a dense matrix, the last h columns of $\mathbf{W}_1\mathbf{W}_2^{-1}$ are more likely to be dense even when $\mathbf{D}_2 = \mathbf{0}$. But $\mathbf{S}_2$ must be nonzero to guarantee that the first $K - K' - h$ columns of $\mathbf{W}_1\mathbf{W}_2^{-1}$ are dense.

With such a parity-check matrix, we write the parity-check packets as $\mathbf{P} = \begin{bmatrix}\mathbf{P}_{\mathrm{L}} & \mathbf{P}_{\mathrm{H}}\end{bmatrix}$ where $\mathbf{P}_{\mathrm{H}}$ is formed by the last h columns of $\mathbf{P}$. The packets in $\mathbf{P}_{\mathrm{L}}$ and $\mathbf{P}_{\mathrm{H}}$ are called the LDPC and HDPC packets, respectively.

Pre-inactivation

We now show that if the batch degrees are high, the probability of $\mathrm{rk}(\mathbf{Z}) \geq K'$ given that the total rank is at least K' is close to 1.

Lemma 8.10 *When all the batches have degree K, for $K/K' \geq 1 + \delta$,*

$$\Pr\left\{rk(\mathbf{Z}) \geq (1+\delta)K' \,\middle|\, \sum_{i=1}^{n} rk(\mathbf{H}_i) \geq (1+\delta)K'\right\} \geq \zeta^K_{(1+\delta)K'}(q).$$

Proof. The matrix $\tilde{\mathbf{G}} = \begin{bmatrix}\mathbf{G}_1 & \cdots & \mathbf{G}_n\end{bmatrix}$ is a $K \times nM$ totally random matrix. Form a matrix $\tilde{\mathbf{H}}'$ by putting together $(1+\delta)K'$ linearly independent columns of $\tilde{\mathbf{H}} = \mathrm{diag}(\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_n)$. Note

that $\mathbf{Z} = \tilde{\mathbf{G}}\tilde{\mathbf{H}}$. Then,

$$\begin{aligned}
&\Pr\left\{\mathrm{rk}(\mathbf{Z}) \geq (1+\delta)K' \middle| \sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i) \geq (1+\delta)K'\right\} \\
&= \Pr\left\{\mathrm{rk}(\tilde{\mathbf{G}}\tilde{\mathbf{H}}) \geq (1+\delta)K' \middle| \sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i) \geq (1+\delta)K'\right\} \\
&\geq \Pr\left\{\mathrm{rk}(\tilde{\mathbf{G}}\tilde{\mathbf{H}}') = (1+\delta)K' \middle| \sum_{i=1}^{n} \mathrm{rk}(\mathbf{H}_i) \geq (1+\delta)K'\right\} \\
&= \zeta^{K}_{(1+\delta)K'}(q).
\end{aligned}$$

□

We know that:

1. when K, K', and δ are constants with $K/K' \geq 1+\delta$, $\zeta^{K}_{(1+\delta)K'}(q) \to 1$ when $q \to \infty$; and
2. when q is a constant, and $\lim_{K'\to\infty} K/K' > 1+\delta$, $\zeta^{K}_{(1+\delta)K'}(q) \to 1$ when $K' \to \infty$.

The above lemma suggests the use of a large batch degree to increase the probability that the inactive packets can be decoded. We also know that a BATS code must employ a proper degree distribution to guarantee the low encoding/decoding complexity. Pre-inactivation enables us to design batch encoding satisfying both requirements.

Specifically, the (precoded) input packets are separated into *active* input packets and *pre-inactive* input packets. The encoding of a batch with pre-inactivation consists of two steps.

1. The normal BATS code encoding is applied on the active input packets only. The degree used in this step follows a degree distribution optimized for BP decoding (as discussed in Chapter 6 and Section 7.4).
2. Each encoded packet in the first step is further combined with a number of pre-inactive packets.

These pre-inactive packets are inactive from the beginning of the decoding, so their involvement in the encoding does not affect the BP decoding.

8.3.2 ENCODING WITH HDPC AND PRE-INACTIVATION

Now we use an example to show how precoding and BATS encoding work when using both HDPC and pre-inactivation. Here, the HDPC packets are always pre-inactive since they are not efficient for BP decoding, and a subset of the input packets is pre-inactivate. All the LDPC packets are active since precoding can increase the number of packets recovered by BP decoding.

Generation of Parity-check Packets

We first see how to generate the parity-check packets of precoding. Write $\mathbf{B}' = \begin{bmatrix}\mathbf{B}'_{\mathrm{A}} & \mathbf{B}'_{\mathrm{I}}\end{bmatrix}$ with $\mathbf{B}'_{\mathrm{I}}$ being the pre-inactive input packets, and write $\mathbf{P} = \begin{bmatrix}\mathbf{P}_{\mathrm{L}} & \mathbf{P}_{\mathrm{H}}\end{bmatrix}$ with $\mathbf{P}_{\mathrm{H}}$ being the HDPC packets. With HDPC and pre-inactivation, the parity check constraints can be written as

$$\begin{bmatrix}\mathbf{B}'_{\mathrm{A}} & \mathbf{P}_{\mathrm{L}} & \mathbf{B}'_{\mathrm{I}} & \mathbf{P}_{\mathrm{H}}\end{bmatrix}\begin{bmatrix}\mathbf{S}_1 & \mathbf{D}_1\\ \mathbf{I} & \mathbf{D}_2\\ \mathbf{S}_{\mathrm{P}} & \mathbf{D}_{\mathrm{P}}\\ \mathbf{S}_2 & \mathbf{I}'\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \mathbf{0}'\end{bmatrix}, \tag{8.16}$$

where the columns of $\mathbf{S}_1$ and $\mathbf{D}_1$ are LDPC and HDPC parity check matrices, respectively. The parity-check packets can be obtained by solving

$$\begin{bmatrix}\mathbf{P}_{\mathrm{L}} & \mathbf{P}_{\mathrm{H}}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{D}_2\\ \mathbf{S}_2 & \mathbf{I}'\end{bmatrix} = -\begin{bmatrix}\mathbf{B}'_{\mathrm{A}} & \mathbf{B}'_{\mathrm{I}}\end{bmatrix}\begin{bmatrix}\mathbf{S}_1 & \mathbf{D}_1\\ \mathbf{S}_{\mathrm{P}} & \mathbf{D}_{\mathrm{P}}\end{bmatrix},$$

which is equivalent to

$$\begin{bmatrix}\mathbf{P}_{\mathrm{L}} & \mathbf{P}_{\mathrm{H}}\end{bmatrix}\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \mathbf{S}_2 & \mathbf{I}' - \mathbf{S}_2\mathbf{D}_2\end{bmatrix} = -\begin{bmatrix}\mathbf{B}'_{\mathrm{A}} & \mathbf{B}'_{\mathrm{I}}\end{bmatrix}\begin{bmatrix}\mathbf{S}_1 & \mathbf{D}_1 - \mathbf{S}_1\mathbf{D}_2\\ \mathbf{S}_{\mathrm{P}} & \mathbf{D}_{\mathrm{P}} - \mathbf{S}_{\mathrm{P}}\mathbf{D}_2\end{bmatrix}.$$

First, calculate

$$\mathbf{B}^* = -\begin{bmatrix}\mathbf{B}'_{\mathrm{A}} & \mathbf{B}'_{\mathrm{I}}\end{bmatrix}\begin{bmatrix}\mathbf{S}_1\\ \mathbf{S}_{\mathrm{P}}\end{bmatrix},$$

which can be done efficiently since $\mathbf{S}_1$ and $\mathbf{S}_{\mathrm{P}}$ are both sparse. Then, $\mathbf{P}_{\mathrm{H}}$ is obtained by solving

$$\mathbf{P}_{\mathrm{H}}(\mathbf{I}' - \mathbf{S}_2\mathbf{D}_2) = -\begin{bmatrix}\mathbf{B}'_{\mathrm{A}} & \mathbf{B}^* & \mathbf{B}'_{\mathrm{I}}\end{bmatrix}\begin{bmatrix}\mathbf{D}_1\\ \mathbf{D}_2\\ \mathbf{D}_{\mathrm{P}}\end{bmatrix},$$

which can be efficiently solved if the matrix multiplication from the left of $\mathbf{D} \triangleq \begin{bmatrix}\mathbf{D}_1\\ \mathbf{D}_2\\ \mathbf{D}_{\mathrm{P}}\end{bmatrix}$ can be efficiently calculated. We refer readers to [69] for an example of HDPC design to achieve this goal. Lastly,

$$\mathbf{P}_{\mathrm{L}} = \mathbf{B}^* - \mathbf{P}_{\mathrm{H}}\mathbf{S}_2.$$

BATS Encoding with Pre-inactivation

Let $\mathbf{B} = \begin{bmatrix}\mathbf{B}'_{\mathrm{A}} & \mathbf{P}_{\mathrm{L}} & \mathbf{B}'_{\mathrm{I}} & \mathbf{P}_{\mathrm{H}}\end{bmatrix}$, $\mathbf{B}_{\mathrm{A}} = \begin{bmatrix}\mathbf{B}'_{\mathrm{A}} & \mathbf{P}_{\mathrm{L}}\end{bmatrix}$, and $\mathbf{B}_{\mathrm{I}} = \begin{bmatrix}\mathbf{B}'_{\mathrm{I}} & \mathbf{P}_{\mathrm{H}}\end{bmatrix}$. Packets in $\mathbf{B}_{\mathrm{I}}$ are pre-inactive. With pre-inactivation, batch i is encoded as

$$\mathbf{X}_i = \begin{bmatrix}\mathbf{B}_i & \bar{\mathbf{B}}_i\end{bmatrix}\begin{bmatrix}\mathbf{G}_i\\ \bar{\mathbf{G}}_i\end{bmatrix} = \mathbf{B}_i\mathbf{G}_i + \bar{\mathbf{B}}_i\bar{\mathbf{G}}_i,$$

where $\mathbf{B}_i \subset \mathbf{B}_\mathrm{A}$ and $\mathbf{G}_i$ have the same meanings as in the BATS code encoding in Section 2.1; $\bar{\mathbf{B}}_i \subset \mathbf{B}_\mathrm{I}$ are the pre-inactive packets used in the encoding of this batch; and $\bar{\mathbf{G}}_i$ is the generator matrix of the pre-inactive packets. Readers can find a design example of $\bar{\mathbf{B}}_i$ and $\bar{\mathbf{G}}_i$ in [69].

The index set A_i now only includes the indices of the active packets. Pre-inactivation is transparent for recoding. Recall that $\mathbf{H}_i$ the batch transfer matrix of batch i, so the received packets of batch i is given by $\mathbf{Y}_i = \mathbf{X}_i\mathbf{H}_i$.

With HDPC and pre-inactivation, the system of equations formed by BATS codes becomes

$$\begin{bmatrix}\mathbf{B}'_\mathrm{A} & \mathbf{P}_\mathrm{L} & \mathbf{B}'_\mathrm{I} & \mathbf{P}_\mathrm{H}\end{bmatrix}\begin{bmatrix}\mathbf{Z}_1 & \mathbf{S}_1 & \mathbf{D}_1\\ \mathbf{Z}_2 & \mathbf{I} & \mathbf{D}_2\\ \mathbf{Z}_\mathrm{I} & \mathbf{S}_\mathrm{P} & \mathbf{D}_\mathrm{P}\\ \mathbf{Z}_\mathrm{H} & \mathbf{S}_2 & \mathbf{I}'\end{bmatrix} = \begin{bmatrix}\mathbf{Y} & \mathbf{0} & \mathbf{0}'\end{bmatrix}, \tag{8.17}$$

where

- $\begin{bmatrix}\mathbf{Z}_1\\ \mathbf{Z}_2\end{bmatrix}$ is determined by $\mathbf{G}_i$ and $\mathbf{H}_i$, $i = 1, \ldots, n$ as in (8.12); and
- $\begin{bmatrix}\mathbf{Z}_\mathrm{I}\\ \mathbf{Z}_\mathrm{H}\end{bmatrix}$ is formed by the coefficients of the pre-inactive packets $\bar{\mathbf{G}}_i$, $i = 1, \ldots, n$.

8.3.3 DECODING OF INACTIVE PACKETS

To make it clear how the inative packets are solved, we discuss the operations of INAC(n) in a way similar to our discussion of BP(n) in Section 2.3.1. To simplify the notation, we omit the superscripts for the time index.

Denote by $\mathcal{I}$ the index set of the inactive (precoded) packets. Initially, $\mathcal{I}$ includes the indices of $\mathbf{B}_\mathrm{I}$. Let $\mathbf{J}_i = \mathbf{G}_i\mathbf{H}_i$. The associated linear system of batch i can be rewritten as

$$\mathbf{Y}_i = \mathbf{B}_i\mathbf{J}_i + \bar{\mathbf{B}}_i\bar{\mathbf{G}}_i\mathbf{H}_i = \mathbf{B}_i\mathbf{J}_i + \mathbf{B}_\mathcal{I}\mathbf{J}_{i,\mathcal{I}}, \tag{8.18}$$

where $\mathbf{B}_\mathcal{I}$ is the subset of $\mathbf{B}$ with columns of indices in $\mathcal{I}$, and $\mathbf{J}_{i,\mathcal{I}}$ is formed by expending $\bar{\mathbf{G}}_i\mathbf{H}_i$ with all-zero rows such that $\bar{\mathbf{B}}_i\bar{\mathbf{G}}_i\mathbf{H}_i = \mathbf{B}_\mathcal{I}\mathbf{J}_{i,\mathcal{I}}$.

We say that a batch i is *decodable* if $\mathrm{rk}(\mathbf{J}_i) = |A_i|$, where only the active packets are taken into consideration. INAC(n) updates $A_i, \mathbf{J}_i, \mathbf{J}_{i,\mathcal{I}}, \mathcal{I}$, and $\mathbf{Y}_i$ as follows. For each decoding iteration, if a decodable batch exists, we select one such batch, say batch i, and apply the following operations.

1. Apply column operations on the linear system (8.18) of batch i so that $\mathbf{J}_i$ is transformed into the form $\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}$. $\mathbf{Y}_i$ and $\mathbf{J}_{i,\mathcal{I}}$ are updated by the same transformation. The associated linear system of batch i after the transformation becomes

$$\mathbf{Y}_i = \mathbf{B}_i\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix} + \mathbf{B}_\mathcal{I}\mathbf{J}_{i,\mathcal{I}},$$

or

$$\mathbf{Y}'_i = \mathbf{B}_i + \mathbf{B}_{\mathcal{I}}\mathbf{J}'_{i,\mathcal{I}}, \tag{8.19}$$
$$\mathbf{Y}''_i = \mathbf{B}_{\mathcal{I}}\mathbf{J}''_{i,\mathcal{I}}, \tag{8.20}$$

where $\begin{bmatrix}\mathbf{Y}'_i & \mathbf{Y}''_i\end{bmatrix} = \mathbf{Y}_i$ and $\begin{bmatrix}\mathbf{J}'_{i,\mathcal{I}} & \mathbf{J}''_{i,\mathcal{I}}\end{bmatrix} = \mathbf{J}_{i,\mathcal{I}}$. Equation (8.19) will be used to solve the active packets involved in batch i and (8.20) provides the constraints for inactive packets.

2. Select an active input packets from $\mathbf{B}_i$, say b_j. We treat b_j as decoded in the form

$$b_j = \tilde{b}_j - \mathbf{B}_{\mathcal{I}}c_j,$$

where $\tilde{b}_j$ and c_j are the corresponding columns of $\mathbf{Y}'_i$ and $\mathbf{J}'_{i,\mathcal{I}}$, respectively, and the inactive packets $\mathbf{B}_{\mathcal{I}}$ will be substituted back when they are solved. We then substitute b_j into the batches it contributes to. Suppose $j \in A_{i'}$. The substitution of b_j in batch i' includes the following operations:

   (a) remove j from $A_{i'}$;

   (b) remove the row z of $\mathbf{J}_{i'}$ corresponding to b_j;

   (c) change $\mathbf{Y}_{i'}$ to $\mathbf{Y}_{i'} - \tilde{b}_j z$; and

   (d) change $\mathbf{J}_{i',\mathcal{I}}$ to $\mathbf{J}_{i',\mathcal{I}} - c_j z$.

If there are no decodable batches, we select an active input packet, say b_j, to inactivate.

1. Add j into $\mathcal{I}$.

2. For each batch i,

   (a) if $j \in A_i$, then remove j from A_i, remove the row z of $\mathbf{J}_i$ corresponding to b_j, and insert row z to $\mathbf{J}_{i,\mathcal{I}}$;

   (b) if $j \notin A_i$, then insert an all-zero row to $\mathbf{J}_{i,\mathcal{I}}$.

The LDPC constraints can be processed in a way similar to the above BP decoding process: Each LDPC constraint can be regarded as a batch of size one with an all-zero received packet. If the HDPC constraints are employed, they will be used for solving the inactive packets.

The above decoding process is repeated until all the packets in $\mathbf{B}$ are either decoded or inactive. Let $\bar{\mathcal{I}}$ be the index set of the decoded packets. The decoder then tries to solve the inactive packets. A system of linear equations of the inactive packets is formed by

1. Equation (8.20) from all the batches, and

2. the HDPC constraints.

Write the decoded packets $\mathbf{B}_{\bar{\mathcal{I}}}$ as

$$\mathbf{B}_{\bar{\mathcal{I}}} = \tilde{\mathbf{B}}_{\bar{\mathcal{I}}} - \mathbf{B}_{\mathcal{I}}\mathbf{J}_{\bar{\mathcal{I}}}, \tag{8.21}$$

where $\tilde{\mathbf{B}}_{\bar{\mathcal{I}}}$ and $\mathbf{J}_{\bar{\mathcal{I}}}$ are formed by juxtaposing $\tilde{b}_j$ and c_j, $j \in \bar{\mathcal{I}}$, respectively. Then, $\mathbf{B}\begin{bmatrix}\mathbf{D}\\ \mathbf{I}'\end{bmatrix} = \mathbf{0}$ gives a part of the constrains of $\mathbf{B}_{\mathcal{I}}$ from HDPC.

The decoder should try to solve the inactive packets after the total rank of the received batches is at least K'. After the inactive packets $\mathbf{B}_{\mathcal{I}}$ are decoded, they will be substituted into (8.21) to recover $\mathbf{B}_{\mathcal{I}}$.

CHAPTER 9

BATS Codes in General Networks

Line networks are mainly used as the example to discuss the design and analysis of BATS protocols in Chapters 3 and 4. Line networks not only have important applications, but also are the building blocks of more complicated networks. The BATS protocols designed for a line network can be used in a more complicated network with some modifications.

In this chapter, we study BATS protocols for networks beyond line networks and discuss some of their applications. We assume that the network links all have unit capacity (per use). We allow multiple edges between two network nodes, which can be used to model a network link of non-unit capacity. If all the network links in a network have the same packet loss rate, these links are said to be *homogeneous*, otherwise *heterogeneous*.

9.1 UNICAST NETWORKS

A unicast network is represented by a *directed acyclic* graph with one source node and one destination node. See Figure 9.1 for a network with a non-line topology. The BATS code protocol designed for a line network, e.g., BATS-Pro-0, can be applied to homogeneous unicast networks except that it needs to handle two new situations at the network layer:

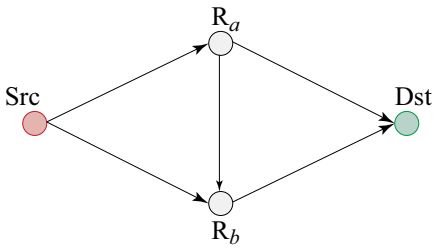


Figure 9.1: An example of unicast network.

1. At network nodes with more than one *incoming* link (e.g., R_b and Dst in Figure 9.1), the network layer should handle the packets received from all links.
2. At network nodes with more than one *outgoing* link (e.g., Src and R_a in Figure 9.1), the network layer should decide how to transmit packets on its outgoing links.

Since the network layer can process packets from all incoming links in the exactly same way, the first situation can be simply be handled by a multiplexer. We focus on the second situation, where a network node has multiple outgoing links. We first consider unicast networks with homogeneous links, and then discuss unicast networks with heterogeneous links.

9.1.1 HOMOGENEOUS UNICAST

Consider a unicast network with packet loss rate ϵ for all links. The network layer of a BATS protocol should know how to transmit the recoded packets on its outgoing links. We introduce one such approach. Find a set of L edge-disjoint paths from the source node to the destination node. A network node transmits the recoded packets only on the L outgoing links in these paths, and all the packets belonging to the same batch are transmitted on the same outgoing links. We further require that a batch is always transmitted on the same path, which is not a necessity but it simplifies our analysis. In practice, a network node may assign an available outgoing link randomly for each batch.

The above transmission scheme in a unicast network is equivalent to the transmission on L line networks. It is not necessary to use L BATS encoders/decoders since one encoder/decoder can serve all the paths. For each path, when using the RLNC recoding discussed in Chapter 3, the normalized expected rank of the transfer matrices converges to $1-\epsilon$ as the batch size $M \to \infty$. So the overall throughput of a BATS protocol can be very close to $L(1-\epsilon)$, which is the min-cut capacity of the network if L is the maximum number of edge-disjoint paths from the source node to the destination node.

If we apply the advanced recoding schemes in Chapter 4, the numbers of recoded packets of a batch can be different for different outgoing links, and they can be optimized for each path individually.

9.1.2 HETEROGENEOUS UNICAST

Now we consider a general unicast network where the network links may have different packet loss rates, for which direct application of the edge-disjoint path approach may not be optimal. Consider the example in Figure 9.2 with $\epsilon_1 = \epsilon_3 = 0.1$ and $\epsilon_2 = \epsilon_4 = 0.2$. The capacity of the network is the min-cut 1.7. If we pick the edge-disjoint paths (e_1, e_4) and (e_2, e_3), where the min-cut of both paths is 0.8, and apply the BATS protocol on both paths, the throughput is upper bounded by 1.6. A better choice is to use the path (e_1, e_3) and (e_2, e_4), for which the min-cuts of are 0.9 and 0.8, respectively, so that the throughput of a BATS protocol can approach the capacity.

For a general unicast network G with heterogeneous links, we can apply BATS codes in three steps.

1. Obtain a unicast network G^* with homogeneous links that has the same min-cut G.

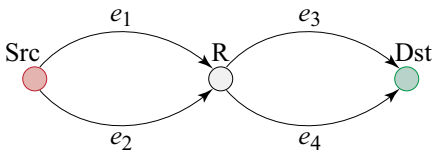


Figure 9.2: An example of heterogeneous unicast network G_8, where edge e_i has a packet loss rate ϵ_i.

2. Design a BATS protocol for network G^* using an approach for homogeneous unicast networks.

3. Convert the protocol on G^* to one that can be used in network G while preserving the performance.

The second step has already been discussed. The first and third steps are now explained.

Assume that the link erasure probabilities are all rational. Fix a sufficiently large integer N such that $(1-\epsilon)N$ is an integer for the packet loss rate ϵ of any link in the network. Then,

$$1 \leq (1-\epsilon)N,$$

or

$$\epsilon \leq 1 - 1/N. \tag{9.1}$$

Network G^* has the same set of nodes as network G. For any link e between nodes a and b in G with packet loss rate ϵ, we have a set of $(1-\epsilon)N$ parallel links, denoted by e^i, $i = 1, \ldots, (1-\epsilon)N$ between nodes a and b in G^* with packet loss rate $1 - 1/N$, which does not depend on ϵ. Network G^* is called the *homogenized network* of G, and the min-cut of network G and G^* are the same. For G^*, we can find edge-disjoint paths from the source node to the destination node and applied the BATS protocols for a line network on each path. The protocol on G^* can be converted to one on G as follows. Each node in G can use the same network layer operations as in the corresponding node in G^*, except that (1) the batches transmitted on an edge e^i in G^* is transmitted in the corresponding edge e in G and (2) the number of packets transmitted for each batch is adjusted (to be explained below) so that the transmission on e can emulate the transmission on e^i.

We use the network G_8 in Figure 9.2 as an example to illustrate the above method. Suppose $\epsilon_i = i/10$, $i = 1, 2, 3, 4$. The min-cut of the network is 1.3. Let $N = 10$. We see that $N(1-\epsilon_i)$ is an integer for $i = 1, 2, 3, 4$. The homogenized network G_8^* has 17 parallel links from node Src to node R, where 9 correspond to e_1 and 8 correspond to e_2, and 13 parallel links from node R to node Dst, where 7 correspond to e_3 and 6 correspond to e_4. All the links in the homogenized network have packet loss rate $1 - 1/N = 0.9$.

We can find 13 edge-disjoint paths from Src to Dst, where the min-cut of each path is 0.1. Suppose a BATS code with batch size M is used for G_8^* and the number of recoded packets

of a batch is $\tilde{M}$. For example, when $M = 16$, the optimal value of $\tilde{M}$ is about 200, and the average number of received packets for a batch in G_8^* is $\frac{\tilde{M}}{N} = 20$. When M is large, this BATS code protocol can achieve a throughput very close to 1.3 for G_8^*.

When using this BATS code protocol on G_8, we do not need to transmit $\tilde{M}$ packets for each batch on edge e_k, $k = 1, 2, 3, 4$ since the packet loss rate of edge e_i is $\epsilon_k < 0.9$. To emulate the transmission in G_8^*, we can artificially delete a packet with probability $1 - \frac{1}{N(1-\epsilon_i)}$ before transmitting the packet on edge e_i, so that the effective packet loss rate on edge e_i is $1 - \frac{1}{N}$. Hence, a BATS protocol can as well achieve a throughput very close to the min-cut of G_8.

Instead of deleting the packets artificially, a better approach in practice is to generate and transmit $\frac{\tilde{M}}{N(1-\epsilon)}$ packets for each batch on an edge in G_8, so that the number of received packets of a batch has the binomial distribution $B(\frac{\tilde{M}}{N(1-\epsilon)}, 1-\epsilon)$. Note that in G_8^*, the number of received packets of a batch has the binomial distribution $B(\tilde{M}, 1/N)$. Both distributions have the same mean $\frac{\tilde{M}}{N}$, but the latter has a smaller variance due to (9.1).

9.2 MULTICAST NETWORKS

Now we turn to a multicast network represented by a directed acyclic graph with one source node and multiple destination nodes. We will only discuss multicast networks with homogeneous links, and our discussion can be applied on multicast networks with heterogeneous links using the same approach for unicast networks.

9.2.1 TREE PACKING

We first consider a homogeneous network with a tree topology, where the root is the source node and all the leaves are destination nodes. There is at most one edge between any two nodes; multi-edge tree will be considered later. A BATS protocol for a line network can be extended to a tree network by allowing a node to transmit the same packets on all its outgoing links, which is called a *tree protocol*.

Suppose the network has k destination nodes $t_1, t_2, \ldots, t_k$ where t_1 has the largest depth among all the destination nodes. The min-cut of the tree network is the min-cut of the node t_1. Intuitively, a destination node with a smaller depth can emulate node t_1, so that we can use the BATS code optimized for node t_1 to approach the multicast capacity. We actually can show that the rank distribution obtain at node t_j, $j > 1$, dominates that of t_1. Hence, by Lemma 6.12, if t_1 can decode successfully by BP decoding with a high probability, all the other destination nodes do so. Therefore, a BATS protocol can achieve a throughput very close to the min-cut of the tree.

We can extend the tree protocol for a general multicast network G with homogeneous links by packing trees. Find a collection $\mathcal{T}$ of subgraphs of G with a tree topology, where the the root is the source node and the leaves are the destination nodes. Note that a tree in $\mathcal{T}$ may

include only a subset of the destination nodes as their leaves, and different trees in $\mathcal{T}$ may share common links and nodes. The source node of G can generate and transmit batches on different trees using the tree protocol.

To see how to optimally pack the trees in $\mathcal{T}$, we introduce an optimization problem. Suppose the network G has E pairs of nodes between each of which there exists at least one link. Define an order on these E pairs of nodes and let C be a vector of E entries where $C[i]$ is the number of links between the i-th pair of nodes. For $T \in \mathcal{T}$, let L_T be a vector of E entries where $L_T[i]$ is 1 if a link between the i-th pair of nodes is used in T, and 0 otherwise. Define an order on the destination nodes, and for $T \in \mathcal{T}$, let D_T be a vector indicating the destination nodes in T, i.e., $D_T[i]$ is 1 if the i-th destination node is a leaf in T, and 0 otherwise. Consider the following optimization problem:

$$\begin{aligned} \max_{\alpha_T \geq \mathbf{0}, T \in \mathcal{T}} \quad & R \\ \text{s.t.} \quad & \sum_{T \in \mathcal{T}} \alpha_T L_T \leq C \\ & \sum_{T \in \mathcal{T}} \alpha_T D_T \geq R \cdot \mathbf{1}, \end{aligned} \tag{9.2}$$

where the inequalities are evaluated component-wise. Let R^* be the optimal value of (9.2). We know that a BATS protocol can achieve a rate very close to $R^*(1-\epsilon)$, where ϵ is the packet loss rate.

In general, $R^*(1-\epsilon)$ is smaller than the min-cut of the network. We give two examples where $R^*(1-\epsilon)$ is the min-cut. The first example is a multi-edge tree network with homogeneous links. In this case, R^* is the maximum number of edge-disjoint trees with the source node as the root and all the destination nodes as the leaves.

Another example is a three-layer network with homogeneous links, where the top layer has only the source node, the bottom layer consists of only the destination nodes, and there exist no direct links between the source node and a destination node. See Figure 9.3 for an example. In this case, R^* is the minimum degree of all the destination nodes, and can be achieved by packing the trees each of which is formed by a node at the middle layer, together with its incoming and outgoing edges. In the network in Figure 9.3, we have three such edge-disjoint trees and $R^* = 2$.

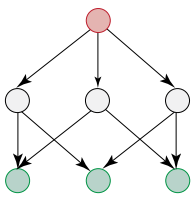


Figure 9.3: A three-layer network. Node at the top layer is the source node. Nodes at the middle layer are the intermediate nodes. Nodes at the bottom layer are the destination nodes.

A three-layer network models a content distribution network (CDN), where the source node is the server, the nodes at the middle layer are the edge servers that are close to the users represented by the nodes at the bottom layer. Each edge server stores a certain amount of data, which is represented by the transmissions from the source node to the middle layer. The data stored at the edge servers will be further consumed by multiple users, which is represented by the transmissions from the middle layer to the bottom layer. Classical CDN stores parts of the original files at the edge servers, which is not optimal. In general, coding at the source node is necessary in terms of achieving the min-cut, even when there is no packet loss [103]. In other words, to satisfy the demands of multiple destination nodes, the data stored at the middle layer should be coded. With packet loss, recoding at the middle layer is also necessary to achieve the min-cut. Using BATS codes for CDN, each edge serve can store a number batches received from the server. When a user wants to download the file, the edge servers transmit recoded packets to the user, which decodes the file after a sufficient number of packets are received.

9.2.2 MULTICAST PROTOCOL I

Tree packing is not optimal in general. The butterfly network in Figure 9.4 is such an example, where the network links are homogeneous and have packet loss rate ϵ. The min-cut of the network is $2(1-\epsilon)$. Using tree packing, the optimal value of (9.2) is 1.5 so that the throughput of a BATS protocol is upper bounded by $1.5(1-\epsilon)$. From the literature of network coding, we know that to achieve the min-cut of the butterfly network, network coding among the packets received from both the incoming edges of node c is necessary [103]. The tree packing approach, however, does not employ network coding among the packets received from both the incoming edges of node c.

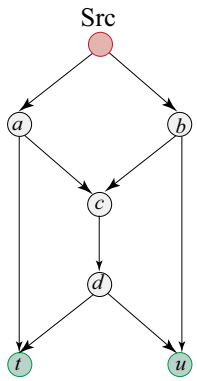


Figure 9.4: Butterfly network. Node Src is the source node. Node t and u are destination nodes.

We discuss two BATS code protocols that can benefit from network coding among the packets received from both the incoming edges of node c. To simplify the discussion, we as-

sume the number of recoded packets is the same as the batch size. The performance of these two protocols can be improved by optimizing the number of recoded packets as discussed in Chapter 4.

We use the butterfly network to illustrate the first protocol. Suppose the source node generates batches of size M, an even number. The source node transmits $M/2$ packets of a batch on link (Src, a) and the other $M/2$ packets of the batch on link (Src, b). For each batch received by node a (resp. b or d), $M/2$ recoded packets are generated and transmitted on both outgoing links. Node c receives packets belonging to the same batch from both of its incoming edges, generates $M/2$ recoded packets, and transmits them on its outgoing edge. Node c can potentially receive up to M packets of a batch from both of its incoming edges, but it only generates $M/2$ recoded packets for each batch. This ensures that on each link in the network, it takes exactly $M/2$ uses of the link for the transmission of a batch. Note that all the network nodes only apply network coding for packets belonging to the same batch.

Each sink node can potentially receive up to M packets of a batch from both of its incoming edges. The expected rank of the transfer matrix normalized by $M/2$ converges to $2(1-\epsilon)$ when the batch size is sufficiently large. Due to symmetry, the two destination nodes have the same rank distribution of the batch transfer matrices. Therefore, a BATS code protocol can achieve a network throughput very close to the multicast capacity of the butterfly network when the batch size is sufficiently large.

This protocol, called Multicast Protocol I, can be applied to a general multicast network (see also [92]). Consider a multicast network G with homogeneous links of packet loss rate ϵ. Without loss of generality, we assume that the source node has L outgoing links, all the destination nodes have L incoming edges, and the number of edge-disjoint paths from the source node to each destination node is L.

In Multicast Protocol I, the source node generates batches of size M, which can be divided by L, and transmits the L packets of a batch on all its outgoing links evenly using M/L time slots. All the intermediate network nodes also use M/L time slots to transmit a batch on each of its outgoing links. Using an upstream-to-downstream order, each intermediate network node applies the following (vector) linear network coding of block length M/L. Fix an intermediate node v. Let $\mathbf{X}_v$ be the k $(0 \le k \le M)$ packets of a batch received by node v from all its incoming edges, which is a matrix of k columns. For each outgoing edge e of v, the recoded packets of the batch transmitted on edge e are

$$\mathbf{X}_e = \mathbf{X}_v \Phi_e,$$

where Φ_e is a $k \times \frac{M}{L}$ matrix. When using RLNC recoding, Φ_e is totally random.

Note that in Multicast Protocol I, the linear network coding at all the intermediate nodes applies only on packets belonging to the same batch. According to the theory of random linear network coding, the expected rank of the transfer matrix normalized by the network uses converges to the min-cut $L(1-\epsilon)$ for each destination node when the batch size tends to infinity [13, 45, 88]. However, the rank distributions of different destination nodes may be different.

In general, we need the approach discussed in Section 6.3 to optimize the degree distribution for multicast.

This protocol may be difficult to implement when L is large. In the above protocol, M/L recoded packets are transmitted on each edge. We need M/L to be sufficient large so that the BATS code can compensate for the packet loss effectively. Hence, the batch size M must be a number that increases with L, e.g., we may use $M = 16L$. However, for $L = 5$, M may already be too large for efficient encoding and decoding on many devices.

9.2.3 MULTICAST PROTOCOL II

We discuss another multicast protocol, called Multicast Protocol II, that may resolve the issues of the Multicast Protocol I. We again use the network G in the last subsection. In Multicast Protocol II, we first fix L edge-disjoint paths from the source node to each sink node, and denote by $\mathcal{P}$ the collection of the paths to all the destination nodes. Assume that each edge in G is used by at least one path in $\mathcal{P}$. The second protocol operates as follows.

- The source node generates a sequence of batches of size M and transmits L different batches on its L outgoing links.
- At an intermediate node v, vector linear network coding of block length M is applied on batches received from different incoming links. In particular, denote by a matrix $\mathbf{X}_e$ the packets of a batch received from an incoming edge e of node v. Note that $\mathbf{X}_e$ and $\mathbf{X}_{e'}$ are different batches whenever $e' \neq e$. For each outgoing edge f of node v, the packets transmitted on edge f is formed by

$$\mathbf{X}_f = \sum_{e \in \mathrm{In}(v)} \mathbf{X}_e \Phi_{e,f},$$

  where $\Phi_{e,f}$ is an M-column matrix, and $\mathrm{In}(v)$ is the set of incoming edges of v. When using RLNC recoding, $\Phi_{e,f}$ is totally random. $\mathbf{X}_f$ is regarded as a *new batch*. This operation at node v is executed according to an upstream-to-downstream order.

Unlike Multicast Protocol I, Multicast Protocol II does not require the batch size to be large when L is large, but the network coding operations use packets belonging to different batches. Analyzing the performance of BP decoding when packets belonging to different batches are jointly recoded is beyond the scope of this monograph. Here, we use an example to illustrate that it is possible to decode efficiently when packets of two batches are combined at the intermediate nodes.

Consider again the butterfly network. The source node separates its input packets into two groups A and B. The source node encodes packets in group A and in group B using two BATS codes of batch size M, and the batches generated using groups A and B are denoted by $\mathbf{X}_A[i]$ and $\mathbf{X}_B[i]$, $i = 1, 2, \ldots$, respectively. Batches $\mathbf{X}_A[i]$, $i = 1, 2, \ldots$ are transmitted on link (s, a) and batches $\mathbf{X}_B[i]$, $i = 1, 2, \ldots$ are transmitted on link (s, b). Nodes a, b, c, and d apply the

recoding as described in Multicast Protocol II. When all the network links have the same delay, node c would receive packets of $\mathbf{X}_A[i]$ and $\mathbf{X}_B[i]$ simultaneously from both incoming edges. After recoding, node c transmits the packets $\mathbf{X}_A[i]\mathbf{H}_A^c[i] + \mathbf{X}_B[j]\mathbf{H}_B^c[i]$, where $\mathbf{H}_A^c[i]$ and $\mathbf{H}_B^c[i]$ are the transfer matrices for $\mathbf{X}_A[i]$ and $\mathbf{X}_B[i]$ at node c, respectively.

Node t first decodes the packets in group A using the batches received from link (a,t). The packets received from link (d,t) are in batches of the form

$$\mathbf{Y}[i] = \mathbf{X}_A[i]\mathbf{H}_A[i] + \mathbf{X}_B[j]\mathbf{H}_B[j],$$

where $\mathbf{H}_A[i]$ and $\mathbf{H}_B[i]$ are the corresponding batch transfer matrices. Since group A has been decoded, node t can recover the batch $\mathbf{X}_A[i]$ and cancel the effect of $\mathbf{X}_A[i]$ from the received batch $Y[i]$. Then node t decodes the packets in B. Node u applies a similar decoding procedure.

The butterfly network has two sub-trees with node Src as the root and nodes t and u as the leaves: one sub-tree contains nodes a, c, and d; and the other sub-tree contains nodes b, c, and d. In the above scheme, for each group of the input packets, we apply a BATS code for multicast in one of the two sub-trees. Since the two sub-trees share the network link (c,d), the batches of these two BATS codes are mixed together to share the network link (c,d). Note that we do not mix the batches of the same BATS code. The decoding at a destination node is a kind of successive cancellation: one group of the input packets is first decoded using BP decoding for the corresponding BATS code; the effect of this group is canceled out from the mixed batches; the other group of input packets is then decoded using BP decoding for the other BATS code.

9.3 MORE WIRELESS NETWORK APPLICATIONS

BATS code is in general superior than other approaches for multi-hop wireless network communications, especially in the scenario that the physical communication links have a high packet loss rate and long transmission delay. In wireless communications, packet loss rate is usually high due to interference, channel dynamics, mobility, etc. In deep-space communication, the transmission delay is long. In underwater acoustic communication, the packet loss rate is high and the transmission delay is long.

Multi-hop wireless communication networks find applications in many areas. In urban wireless sensor networks, multi-hop transmission can extend the coverage of the fixed network infrastructure and reduce the cost of using fixed network infrastructure. More than 70% of the surface of Earth is covered by ocean, where communications can only be supported by satellites nowadays. Satellite communication is not only expensive, but is also limited in bandwidth. Moreover, satellite communication cannot be directly used under the water. On the other hand, multi-hop wireless communications can be used in ocean communication for both under and above the water.

The application of BATS codes in wireless networks is being studied extensively. BATS code is ideal for wireless multicast due to its network coding nature. We have discussed such an example in Section 6.3. BATS code is also suitable for multi-path transmission to improve

reliability and security. Instead of discussing more details of wireless network applications, we refer readers to the literature. A two-phase cooperative broadcasting based on BATS codes was proposed in [89]. BATS codes in two-way relay networks has been studied in [24, 107]. Applications of BATS codes in deep-space networks has been discussed in [109].

APPENDIX A

Proof of Theorem 5.7

A.1 A GENERAL THEOREM

The main technique to prove Theorem 5.7 is a general theorem by Wormald [86, 87] with a small modification. The statement of the next theorem follows that of [87, Theorem 5.1] with an extra initial condition. A similar version is provided in [62, Theorem C.28] with a deterministic boundedness condition.

We say a function $f(u_1, \dots, u_j)$ satisfies a *Lipschitz condition* on $\mathcal{D} \subset \mathbb{R}^j$ if there exists a constant C_L such that

$$|f(u_1, \cdots, u_j) - f(v_1, \cdots, v_j)| \le C_L \max_{1 \le i \le j} |u_i - v_i|$$

for all $(u_1, \cdots, u_j)$ and $(v_1, \cdots, v_j)$ in $\mathcal{D}$. We call C_L the Lipschitz constant for f. Note that $\max_{1 \le i \le j} |u_i - v_i|$ is the distance between $(u_1, \cdots, u_j)$ and $(v_1, \cdots, v_j)$ in the l^∞-norm.

Theorem A.1 *Let $\mathcal{G}_0, \mathcal{G}_1, \dots$ be a random process with a positive integer parameter n, and let $(Y_l(t))_{l=0}^L$ be a random vector determined by $\mathcal{G}_0, \dots, \mathcal{G}_t$. For some constant C_0 and all l, $|Y_l(t)| < C_0 n$ for $t \ge 0$ and all n. Let $\mathcal{D}$ be some bounded connected open set containing the closure of*

$$\{(0, z_1, \dots, z_L) : \exists n, \Pr\{Y_l(0) = z_l n, 1 \le l \le L\} \ne 0\}.$$

Define the stopping time $T_\mathcal{D}$ *to be the minimum t such that $(t/n, Y_1(t)/n, \dots, Y_L(t)/n) \notin \mathcal{D}$. Assume the following conditions hold.*

(i) (Boundedness) For some functions $\beta = \beta(n) \ge 1$ and $\gamma = \gamma(n)$, the probability that

$$\max_l |Y_l(t+1) - Y_l(t)| \le \beta,$$

is at least $1 - \gamma$ for $t < T_\mathcal{D}$.

(ii) (Trend) For some function $\lambda_1 = \lambda_1(n) = o(1)$, if $t < T_\mathcal{D}$,

$$\mathbb{E}[Y_l(t+1) - Y_l(t) | \mathcal{G}_1, \dots, \mathcal{G}_t] = f_l\left(\frac{t}{n}, \left(\frac{Y_i^{(t)}}{n}\right)_{i=0}^L\right) + O(\lambda_1),$$

for $1 \le l \le L$.

(iii) (Lipschitz) Each function f_l satisfies a Lipschitz condition on $\mathcal{D} \cap \{(t, z_1, \ldots, z_L), t \geq 0\}$ with the same Lipschitz constant C_L for each l.

(iv) (Initial condition) For some point $(0, z_1^0, \ldots, z_l^0) \in \mathcal{D}$,

$$|Y_l(0)/n - z_l^0| \leq \sigma = o(1), 0 \leq l \leq L.$$

Then the following are true.

(a) For $(0, (\hat{z}_l)_{l=1}^L) \in \mathcal{D}$, the system of differential equations

$$\frac{\mathrm{d}\, z_l(\tau)}{\mathrm{d}\, \tau} = f_l(\tau, (z_{l'}(\tau))_{l'=1}^L), \quad l = 1, \ldots, L,$$

has a unique solution in $\mathcal{D}$ for $z_l : \mathbb{R} \to \mathbb{R}$ passing through $z_l(0) = \hat{z}_l$, $l = 1, \ldots, L$, and this solution extends to points arbitrarily close to the boundary of $\mathcal{D}$.

(b) Let $\lambda > \max\{\sigma, \lambda_1 + C_0 n\gamma\}$ with $\lambda = o(1)$. There exists a sufficiently large constant C_1 such that when n is sufficiently large, with probability $1 - \mathrm{O}(n\gamma + \frac{\beta}{\lambda} \exp(-\frac{n\lambda^3}{\beta^3}))$,

$$|Y_l(t) - n z_l(t/n)| = \mathrm{O}(\lambda n) \tag{A.1}$$

uniformly for $0 \leq t \leq \bar{\tau} n$ and for each l, where $\hat{z}_l = z_l^0$, and $\bar{\tau} = \bar{\tau}(n)$ is the supremum of those τ to which the solution of the system of differential equations in (a) can be extended before reaching within l^∞-distance $C_1\lambda$ of the boundary of $\mathcal{D}$.

Proof. The proof follows exactly the proof of [87, Theorem 5.1] except for the place where we need to handle the initial condition (iv). We only have to modify the definition of B_j (below (5.9) in [87]) in the original proof to

$$B_j = (n\lambda + \omega)\left(\left(1 + \frac{B\omega}{n}\right)^j - 1\right) + B_0\left(1 + \frac{B\omega}{n}\right)^j,$$

where $B_0 = n\lambda$. The induction in the original proof now begins by the fact that $|z_l(0) - Y_l(0)/n| \leq \sigma < \mathrm{O}(\lambda)$. The other part of the proof stays the same as that of [87, Theorem 5.1]. □

A.2 COMPLETING THE PROOF

We first prove two technical lemmas. For BATS(K, n, Π), the degrees of the variable nodes are not independent but follow the same distribution. The following lemma shows that the degree of a variable node is not likely to be much larger than its expectation.

Lemma A.2 *Let V be the degree of a variable node of BATS(K, n, Π). For any $\alpha > 0$,*

$$\Pr\{V > (1+\alpha)\bar{\Psi}/\theta\} < \left(\frac{e^\alpha}{(1+\alpha)^{(1+\alpha)}}\right)^{\bar{\Psi}/\theta},$$

where $\theta = K/n$.

Proof. Fix a variable node. Let X_i be the indicator random variable of the ith check node being the neighbor of the specific variable node. Then $V = \sum_i X_i$. We have $\mathbb{E}[V] = \sum_i \mathbb{E}[X_i] = \sum_i \sum_d \frac{d}{K}\Psi_d = \frac{n}{K}\bar{\Psi} = \frac{\bar{\Psi}}{\theta}$. Since X_i, $i = 1, \ldots, n$, are mutually independent, the lemma is proved by applying the Chernoff bound. □

The following lemma verifies the boundedness condition of Theorem A.1. Let

$$\mathcal{F} = \{(d, r) : 1 \le r \le M, r < d \le D\}.$$

Lemma A.3 *When* $\beta/D > \bar{\Psi}/\theta$, *the probability that*

$$\max_{\iota \in \mathcal{F} \cup \{0\}} |E_\iota^{(t+1)} - E_\iota^{(t)}| \le \beta,$$

is at least

$$1 - \theta n \exp\left(-\frac{\beta}{D}(\ln(\beta/D) - \ln(\bar{\Psi}/\theta) - 1) - \frac{\bar{\Psi}}{\theta}\right).$$

Proof. Let V be the degree of the variable node to be removed at the beginning of time $t + 1$. By (5.17), we have for $(d, r) \in \mathcal{F}$,

$$|E_{d,r}^{(t+1)} - E_{d,r}^{(t)}| \le DV,$$

and by (5.18), we have

$$|E_0^{(t+1)} - E_0^{(t)}| \le DV.$$

Hence when $\beta/D > \bar{\Psi}/\theta$,

$$\begin{aligned}
&\Pr\left\{\max_{\iota \in \mathcal{F} \cup \{0\}} |E_\iota^{(t+1)} - E_\iota^{(t)}| \le \beta\right\} \\
&\ge \Pr\{VD \le \beta\} \\
&\ge \Pr\{\text{degrees of all variable nodes at time zero} \le \beta/D\} \\
&> 1 - \theta n \exp\left(-\frac{\beta}{D}(\ln(\beta/D) - \ln(\bar{\Psi}/\theta) - 1) - \frac{\bar{\Psi}}{\theta}\right),
\end{aligned}$$

where the last inequality follows from Lemma A.2 and the union bound. □

Proof of Theorem 5.7. We consider in the proof only the instances of BATS(K, n, Π) satisfying

$$\left| \frac{E_{d,r}}{n} - \rho_{d,r} \right| < n^{-1/6}, \ 1 \le r \le M, r \le d \le D. \tag{A.2}$$

By Lemma 5.3 this will decrease the probability bounds we will obtain by at most $\lambda(n) + 2M^2 D \exp(-2n^{2/3})$.

Define the stopping time T_0 as the first time t such that $E_0^{(t)} = 0$. By defining suitable functions $f_{d,r}, (d,r) \in \mathcal{F}$ and f_0 we can rewrite (5.15) and (5.16) as

$$\begin{aligned}
\mathbb{E}\left[E_{d,r}^{(t+1)} - E_{d,r}^{(t)} | \bar{E}^{(t)}\right] &= f_{d,r}\left(\frac{t}{n}, \left(\frac{E_0^{(t)}}{n}\right), \left(\frac{E_{d',r'}^{(t)}}{n}\right)_{(d',r')\in\mathcal{F}}\right), \ (d,r) \in \mathcal{F} \\
\mathbb{E}\left[E_0^{(t+1)} - E_0^{(t)} | \bar{E}^{(t)}\right] &= f_0\left(\frac{t}{n}, \left(\frac{E_0^{(t)}}{n}\right), \left(\frac{E_{d',r'}^{(t)}}{n}\right)_{(d',r')\in\mathcal{F}}\right) + \mathrm{O}\left(\frac{1}{n}\right),
\end{aligned}$$

for $t < T_0$. For $\iota \in \mathcal{F} \cup \{0\}$, define random variable $\hat{E}_\iota$ as $\hat{E}_\iota(0) = E_\iota(0)$, and for $t \ge 0$,

$$\hat{E}_\iota^{(t+1)} = \begin{cases} E_\iota^{(t+1)} & t < T_0 \\ \hat{E}_\iota^{(t)} + f_\iota\left(\frac{t}{n}, \left(\frac{E_0^{(t)}}{n}\right), \left(\frac{E_{\tilde{d},\tilde{r}}^{(t)}}{n}\right)_{(\tilde{d},\tilde{r})\in\mathcal{F}}\right) & t \ge T_0. \end{cases}$$

Note that T_0 is also the first time that $\hat{E}_0^{(t)}$ becomes 0.

We now apply Theorem A.1 with $(\hat{E}_0^{(t)}, (\hat{E}_{d,r}^{(t)})_{(d,r)\in\mathcal{F}})$ in place of $(Y_l(t))_{l=1}^L$. The region $\mathcal{D}$ is defined as

$$\mathcal{D} = (-\eta, (1-\eta/2)\theta) \times (-M, M+\eta) \times (-\eta, d)^{|\mathcal{F}|}.$$

So (1) t/n is in the interval $(-\eta, (1-\eta/2)\theta)$; (2) $\hat{E}_0^{(t)}/n$ is in the interval $(-M, M+\eta)$; and (3) $\hat{E}_{d,r}^{(t)}/n$, $(d,r) \in \mathcal{F}$, is in the interval $(-\eta, d)$. As required, $\mathcal{D}$ is a bounded connected open set and containing all the possible initial state $(0, \hat{E}_0(0)/n, (\hat{E}_{d,r}(0)/n)_{(d,r)\in\mathcal{F}})$.

The conditions of Theorem A.1 can readily be verified. When $t \ge T_0$, the change $|\hat{E}_\iota^{(t+1)} - \hat{E}_\iota^{(t)}|$ for $\iota \in \mathcal{F} \cup \{0\}$ is deterministic and upper bounded. When $t < T_0$, by Lemma A.3 with $\beta = n^{1/8}$, the boundedness condition (i) holds with

$$\gamma = n \exp\left(-n^{1/8}\left(c_{1,3} \ln n - c_{1,1}\right) - c_{1,2}\right),$$

where $c_{1,1}$, $c_{1,2}$, and $c_{1,3}$ are only related to $\bar{\Psi}$ and θ. The trend condition (ii) is satisfied with $\lambda_1 = \mathrm{O}(1/n)$. By definition, it can be verified that f_ι, $\iota \in \mathcal{F} \cup \{0\}$ satisfy the Lipschitz condition (iii). The initial condition (iv) holds with $\sigma = \mathrm{O}(n^{-1/6})$.

Wormald's method [86, 87] leads us to consider the system of differential equations

$$\begin{aligned}\frac{\mathrm{d}\,\rho_{d,r}(\tau)}{\mathrm{d}\,\tau} &= f_{d,r}(\tau, \rho_0(\tau), (\rho_{d',r'}(\tau))_{(d',r')\in\mathcal{F}}), \quad (d,r)\in\mathcal{F}\\ \frac{\mathrm{d}\,\rho_0(\tau)}{\mathrm{d}\,\tau} &= f_0(\tau, \rho_0(\tau), (\rho_{d',r'}(\tau))_{(d',r')\in\mathcal{F}})\end{aligned}$$

with the initial condition $\rho_{d,r}(0) = \rho_{d,r}$, $(d,r)\in\mathcal{F}$, and $\rho_0(0) = \sum_r \rho_{r,r}$. The conclusion (a) of Theorem A.1 shows the existence and uniqueness of the solution of the above system of differential equations. Here the system of differential equations can be solved explicitly.

Let $\lambda' = \mathrm{O}(n^{-1/6})$. By the conclusion (b) of Theorem A.1, we know that for a sufficiently large constant C_1, with probability $1 - \mathrm{O}(n\gamma + \frac{\beta}{\lambda'}\exp(-\frac{n\lambda'^3}{\beta^3}))$,

$$\begin{aligned}|\hat{E}^{(t)}_{d,r} - n\rho_{d,r}(t/n)| &= \mathrm{O}(n^{5/6}),\ (d,r)\in\mathcal{F},\\ |\hat{E}^{(t)}_0 - n\rho_0(t/n)| &= \mathrm{O}(n^{5/6})\end{aligned}$$

uniformly for $0\le t\le \bar{\tau}n$, where $\bar{\tau}$ is defined in Theorem A.1. Increase n if necessary so that $\frac{\beta}{\lambda'}\exp(-\frac{n\lambda'^3}{\beta^3}) = n^{7/24}\exp(-n^{-1/8}) > n\gamma$ and $C_1\lambda' < \frac{\eta}{2}\theta$, which implies $\bar{\tau}\ge(1-\eta)\theta$. So there exists constants c_0 and c_0' such that the event

$$E_0 = \{|\hat{E}^{(t)}_0/n - \rho_0(t/n)| \le c_0 n^{-1/6},\ 0\le t\le(1-\eta)K\}$$

holds with probability at least $1 - c_0' n^{7/24}\exp(-n^{-1/8})$.

Now we consider the two cases in the theorem to prove. (i) If $\rho_0(\tau) > 0$ for $\tau\in[0,(1-\eta)\theta]$, then there exists $\epsilon > 0$ such that $\rho_0(\tau)\ge\epsilon$ for $\tau\in[0,(1-\eta)\theta]$. Increase n if necessary so that $c_0 n^{-1/6} < \epsilon$. Then, we have

$$\begin{aligned}\Pr\{T_0 > (1-\eta)K\} &= \Pr\{\hat{E}^{(t)} > 0, 0\le t\le(1-\eta)K\}\\ &\ge \Pr\{E_0\}\\ &\ge 1 - c_0' n^{7/24}\exp(-n^{-1/8}),\end{aligned} \tag{A.3}$$

where (A.3) follows that under the condition E_0, for all $t\in[0,(1-\eta)K]$, $\hat{E}^{(t)}_0/n \ge \rho_0(t/n) - c_0 n^{-1/6} > 0$. Since $\hat{E}_\iota = E_\iota$, $\iota\in\mathcal{F}\cup\{0\}$, when $t < T_0$, the first part of the theorem is proved.

(ii) Consider $\rho_0(\tau_0) < 0$ for $\tau_0\in[0,(1-\eta)\theta]$. There exists $\epsilon > 0$ such that $\rho_0(\tau)\le -\epsilon$ for all $\tau\in[\tau_0-\epsilon,\tau_0+\epsilon]\cap[0,(1-\eta)\theta]$. Increase n if necessary so that $c_0 n^{-1/6} < \epsilon$ and $n\epsilon > 1$. Then, we have

$$\begin{aligned}&\Pr\{T_0\le(1-\eta)K\}\\ &\quad= \Pr\{\hat{E}^{(t)}_0 < 0,\ \text{for some } t\in[0,(1-\eta)K]\}\\ &\quad\ge \Pr\{E_0\}\\ &\quad\ge 1 - c_0' n^{7/24}\exp(-n^{-1/8}),\end{aligned} \tag{A.4}$$

where (A.4) can be shown as follows. Since $n\epsilon > 1$, there exists t_0 such that $t_0/n\in[\tau_0-\epsilon,\tau_0+\epsilon]\cap[0,(1-\eta)\theta]$. Hence, under the condition E_0, $\hat{E}_0(t_0)/n\le c_0 n^{-1/6} + \rho_0(t_0/n) < 0$.

The proof of the theorem is completed by subtracting the probability that (A.2) does not hold. □

A.3 A SYSTEM OF DIFFERENTIAL EQUATIONS

We solve the following system of differential equations given in (5.19) and (5.20), which is reproduced as follows:

$$\begin{aligned}
\frac{d\rho_{d,r}(\tau)}{d\tau} &= \left(\alpha_{d+1,r}\rho_{d+1,r}(\tau) + \bar{\alpha}_{d+1,r+1}\rho_{d+1,r+1}(\tau) - \rho_{d,r}(\tau)\right)\frac{d}{\theta-\tau}, \\
&\qquad 1 \le r \le M,\ r < d \le D, \\
\frac{d\rho_0(\tau)}{d\tau} &= \frac{\sum_{r=1}^{D-1} r\alpha_{r+1,r}\rho_{r+1,r}(\tau) - \rho_0(\tau)}{\theta-\tau} - 1,
\end{aligned}$$

with initial values $\rho_{d,r}(0) = \rho_{d,r}$ and $\rho_0(0) = \sum_r \rho_{r,r}$.

Let $y_{d,r}(\tau) = (1-\tau/\theta)^{-d}\rho_{d,r}(\tau)$. We have

$$\frac{dy_{d,r}(\tau)}{d\tau} = \frac{d}{\theta}(\alpha_{d+1,r}y_{d+1,r}(\tau) + \bar{\alpha}_{d+1,r+1}y_{d+1,r+1}(\tau)).$$

We see that $y_{d,r}(0) = \rho_{d,r}(0)$. We can verify that

$$y_{d,r}(\tau) = \sum_{j=d}^{D}\binom{j-1}{d-1}(\tau/\theta)^{j-d}\rho_{j,r}^{(j-d)},$$

where $\rho_{j,r}^{(j-d)}$ is defined in (5.14). Thus,

$$\rho_{d,r}(\tau) = (1-\tau/\theta)^d\sum_{j=d}^{D}\binom{j-1}{d-1}(\tau/\theta)^{j-d}\rho_{j,r}^{(j-d)}. \tag{A.5}$$

Using the general solution of linear differential equations, we obtain that

$$\begin{aligned}
\rho_0(\tau) &= \left(1-\frac{\tau}{\theta}\right)\left(\int_0^\tau \frac{\sum_{r=1}^{M} r\alpha_{r+1,r}\rho_{r+1,r}(t)}{\theta-t}(1-t/\theta)^{-1}dt + \theta\ln(1-\tau/\theta) + \rho_0(0)\right) \\
&= \left(1-\frac{\tau}{\theta}\right)\left(\sum_{r=1}^{M} r\alpha_{r+1,r}\int_0^\tau \frac{\rho_{r+1,r}(t)}{\theta-t}(1-t/\theta)^{-1}dt + \theta\ln(1-\tau/\theta) + \rho_0(0)\right).
\end{aligned} \tag{A.6}$$

The integral in (A.6) can be further calculated as follows:

$$\begin{aligned}
\int_0^\tau \frac{\rho_{r+1,r}(t)}{\theta - t}(1 - t/\theta)^{-1} dt &= \int_0^\tau \frac{\sum_{j=r+1}^{D} \rho_{j,r}^{(j-r-1)} \binom{j-1}{r} (1 - t/\theta)^{r+1} (t/\theta)^{j-r-1}}{(\theta - t)(1 - t/\theta)} dt \\
&= \int_0^\tau \sum_{j=r+1}^{D} \rho_{j,r}^{(j-r-1)} \binom{j-1}{r} (1 - t/\theta)^{r-1} (t/\theta)^{j-r-1} \frac{dt}{\theta} \\
&= \sum_{j=r+1}^{D} \rho_{j,r}^{(j-r-1)} \binom{j-1}{r} \int_0^{\tau/\theta} (1-t)^{r-1} t^{j-r-1} dt \\
&= \sum_{j=r+1}^{D} \rho_{j,r}^{(j-r-1)} \binom{j-1}{r} \frac{(j-r-1)!(r-1)!}{(j-1)!} \mathrm{I}_{j-r,r}(\tau/\theta) \\
&= 1/r \sum_{j=r+1}^{D} \rho_{j,r}^{(j-r-1)} \mathrm{I}_{j-r,r}(\tau/\theta),
\end{aligned}$$

where the first equality is obtained by substituting $\rho_{r+1,r}(t)$ in (A.5), and the second last equality is obtained by the definition of incomplete beta function (see (5.6)). Therefore, the solution for $\rho_0(\tau)$ is

$$\rho_0(\tau) = \left(1 - \frac{\tau}{\theta}\right) \left(\sum_{r=1}^{M} \alpha_{r+1,r} \sum_{d=r+1}^{D} \rho_{d,r}^{(d-r-1)} \mathrm{I}_{d-r,r}\left(\frac{\tau}{\theta}\right) + \sum_{r=1}^{M} \rho_{r,r} + \theta \ln(1 - \tau/\theta) \right).$$

APPENDIX B

Incomplete Beta Function

Beta function with integer parameters is used extensively in this work. Related results are summarized here. For positive integer a and b, the *beta function* is defined by

$$B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\,\mathrm{d}\,t = \frac{(a-1)!(b-1)!}{(a+b-1)!}.$$

The *(regularized) incomplete beta function* is defined as

$$\begin{aligned} \mathrm{I}_{a,b}(x) &= \frac{\int_0^x t^{a-1}(1-t)^{b-1}\,\mathrm{d}\,t}{B(a,b)} \\ &= \sum_{j=a}^{a+b-1} \binom{a+b-1}{j} x^j (1-x)^{a+b-1-j}. \end{aligned} \tag{B.1}$$

For more general discussion of beta functions, as well as incomplete beta functions, please refer to [106].

Using the above definitions, we can easily show that

$$\int_0^1 \mathrm{I}_{a,b}(x)\,\mathrm{d}\,x = \frac{b}{a+b}, \tag{B.2}$$

and

$$\mathrm{I}_{a+1,b}(x) = \mathrm{I}_{a,b}(x) - \frac{x^a(1-x)^b}{aB(a,b)}. \tag{B.3}$$

Lemma B.1 *$\frac{\mathrm{I}_{a+1,b}(x)}{\mathrm{I}_{a,b}(x)}$ is monotonically increasing in x.*

Proof. By (B.3),

$$\begin{aligned} \frac{\mathrm{I}_{a+1,b}(x)}{\mathrm{I}_{a,b}(x)} &= 1 - \frac{x^a(1-x)^b}{aB(a,b)\,\mathrm{I}_{a,b}(x)} \\ &= 1 - \frac{1}{aB(a,b)\sum_{j=a}^{a+b-1}\binom{a+b-1}{j}x^{j-a}(1-x)^{a-1-j}} \\ &= 1 - \frac{1}{aB(a,b)\sum_{j=0}^{b-1}\binom{a+b-1}{j+a}x^{j}(1-x)^{-1-j}}, \end{aligned}$$

in which $x^j(1-x)^{-1-j}$ is monotonically increasing. □

Lemma B.2 *When* $\frac{b-1}{a+1} \le \frac{\eta}{1-\eta}$ *where* $0 < \eta < 1$, $\frac{\mathrm{I}_{a+1,b}(x)}{\mathrm{I}_{a,b}(x)} \le 1 - \frac{\eta}{b}$ *for* $0 < x \le 1-\eta$ *with equality when* $b = 1$ *and* $x = 1-\eta$.

Proof. Since $\frac{\mathrm{I}_{a+1,b}(x)}{\mathrm{I}_{a,b}(x)}$ is monotonically increasing in x (cf. Lemma B.1), it is sufficient to show $\frac{\mathrm{I}_{a+1,b}(1-\eta)}{\mathrm{I}_{a,b}(1-\eta)} \le 1 - \frac{\eta}{b}$. Since $a + 1 \ge (b-1)\frac{1-\eta}{\eta}$,

$$\begin{aligned}\mathrm{I}_{a,b}(1-\eta) &= \sum_{j=a}^{a+b-1} \binom{a+b-1}{j} (1-\eta)^j \eta^{a+b-1-j} \\ &\le b\binom{a+b-1}{a}(1-\eta)^a \eta^{b-1},\end{aligned}$$

where the equality holds for $b = 1$. Thus,

$$\begin{aligned}\frac{\mathrm{I}_{a+1,b}(1-\eta)}{\mathrm{I}_{a,b}(1-\eta)} &= 1 - \frac{(1-\eta)^a \eta^b}{aB(a,b)\,\mathrm{I}_{a,b}(1-\eta)} \\ &\le 1 - \frac{(1-\eta)^a \eta^b}{abB(a,b)\binom{a+b-1}{a}(1-\eta)^a \eta^{b-1}} \\ &= 1 - \frac{\eta}{b}.\end{aligned}$$

□

We will use the following result about the summation of binomial coefficients:

$$\sum_{j=0}^{n} (-1)^{j-n} \binom{j+m}{n} \binom{n}{j} = 1, \quad m \ge n. \tag{B.4}$$

The above equality can be verified as follows:

$$\begin{aligned}
&\sum_{j=0}^{n}(-1)^{j-n}\binom{j+m}{n}\binom{n}{j} \\
&\quad=\sum_{j=0}^{n}(-1)^{j-n}\binom{j+m}{j+m-n}\binom{n}{j} \\
&\quad=\sum_{j=0}^{n}(-1)^{j-n}(-1)^{j+m-n} \\
&\qquad\qquad\times\binom{-j-m+j+m-n-1}{j+m-n}\binom{n}{j} && \text{(B.5)} \\
&\quad=\sum_{j=0}^{n}(-1)^{m}\binom{-n-1}{j+m-n}\binom{n}{n-j} \\
&\quad=(-1)^{m}\binom{-1}{m} && \text{(B.6)} \\
&\quad=1, && \text{(B.7)}
\end{aligned}$$

where (B.6) follows from Vandermonde's identity; (B.5) and (B.7) use the relation between binomial coefficients with negative integers and positive integers.

Lemma B.3 *For $r \geq 1$,*

$$\sum_{d=r+1}^{\infty}\frac{1}{d-1}\,\mathrm{I}_{d-r,r}(x) = -\ln(1-x), \quad x \in [0,\ 1).$$

Proof. As a special case, when $r = 1$, the equality becomes

$$\sum_{d=2}^{\infty}\frac{x^{d-1}}{d-1} = -\ln(1-x), \tag{B.8}$$

which is the Taylor expansion of $-\ln(1-x)$ for $x \in [0, 1)$.

To prove the general case, let us first derive an alternative form of $\mathrm{I}_{d-r,r}(x)$. For $a > 0$,

$$\begin{aligned}
&\mathrm{I}_{a,b}(x) \\
&= \sum_{j=a}^{a+b-1} \binom{a+b-1}{j} x^j \sum_{i=0}^{a+b-1-j} (-1)^i \binom{a+b-1-j}{i} x^i \\
&= \sum_{m=a}^{a+b-1} x^m \sum_{j=a}^{m} \binom{a+b-1}{j} (-1)^{m-j} \binom{a+b-1-j}{m-j} \\
&= \sum_{m=a}^{a+b-1} (-x)^m \binom{a+b-1}{m} \sum_{j=a}^{m} \binom{m}{j} (-1)^j \\
&= \sum_{m=a}^{a+b-1} (-x)^m \binom{a+b-1}{m} \binom{m-1}{a-1} (-1)^a \\
&= b \binom{a+b-1}{b} (-1)^a \sum_{m=a}^{a+b-1} \frac{(-x)^m}{m} \binom{b-1}{m-a}.
\end{aligned}$$

Using this form for $\mathrm{I}_{d-r,r}(x)$, we have

$$\begin{aligned}
&\sum_{d=r+1}^{\infty} \frac{1}{d-1} \mathrm{I}_{d-r,r}(x) \\
&= \sum_{d=r+1}^{\infty} \frac{r}{d-1} \binom{d-1}{r} \sum_{m=d-r}^{d-1} \binom{r-1}{m-d+r} (-1)^{m-d+r} \frac{x^m}{m} \\
&= \sum_{m=1}^{\infty} \frac{x^m}{m} A_m, \qquad \text{(B.9)}
\end{aligned}$$

where

$$A_m \triangleq \sum_{d=\max\{m,r\}+1}^{m+r} \frac{r}{d-1} \binom{d-1}{r} \binom{r-1}{m-d+r} (-1)^{m-d+r}.$$

For $m \le r$,

$$\begin{aligned}
A_m &= \sum_{d=r+1}^{m+r} \frac{r}{d-1}\binom{d-1}{r}\binom{r-1}{m-d+r}(-1)^{m-d+r} \\
&= \sum_{d=r+1}^{m+r} \binom{d-2}{r-1}\binom{r-1}{m-d+r}(-1)^{m-d+r} \\
&= \sum_{j=0}^{m-1} \binom{j+r-1}{r-1}\binom{r-1}{m-j-1}(-1)^{m-j-1} \\
&= \sum_{j=0}^{m-1} \binom{j+r-1}{m-1}\binom{m-1}{m-j-1}(-1)^{m-j-1} \\
&= 1,
\end{aligned}$$

where the last equality follows from (B.4). Similarly, for $m > r$,

$$\begin{aligned}
A_m &= \sum_{d=m+1}^{m+r} \frac{r}{d-1}\binom{d-1}{r}\binom{r-1}{m-d+r}(-1)^{m-d+r} \\
&= \sum_{d=m+1}^{m+r} \binom{d-2}{r-1}\binom{r-1}{m-d+r}(-1)^{m-d+r} \\
&= \sum_{j=0}^{r-1} \binom{j+m-1}{r-1}\binom{r-1}{r-j-1}(-1)^{r-j-1} \\
&= 1.
\end{aligned}$$

The proof is completed by referring to (B.8) and (B.9) with $A_m = 1$. □

APPENDIX C

Vertices of Convex Polytope

We have the necessary and sufficient condition of a vertex of a convex polytope [108].

Theorem C.1 *Let $\mathcal{F} = \{x \in (\mathbb{R}^+)^n : Ax = b\}$, where $A \in \mathbb{R}^{m\times n}$ has a rank equal to m. A point $x \in \mathcal{F}$ is a vertex of $\mathcal{F}$ if and only if there exists an index partition (B, N) of $\{1, 2, \ldots, n\}$ such that*

$$|B| = rk(A_B) = m, \quad A_B^{-1} b \geq 0$$

and

$$x_B = A_B^{-1} b, x_N = 0,$$

where A_B is the submatrix of A of columns with indices in B, x_B (x_N) are subvector of x of components with indices in B (N).

Let us identify the vertices of

$$\mathcal{D}_\mu = \left\{ h \in (\mathbb{R}^+)^M : \sum_{i=1}^{M} h_i \leq 1, \sum_{i=1}^{M} i h_i = \mu \right\}.$$

To apply the above theorem, we define

$$\mathcal{D}_\mu^* = \left\{ (h_0, h) \in \mathbb{R}^+ \times \mathcal{D}_\mu : \sum_{i=0}^{M} h_i = 1 \right\}.$$

We see that $\mathcal{D}_\mu$ is the image of $\mathcal{D}_\mu^*$ under the projection P defined by $(h_0, h_1, \ldots, h_M) \mapsto (h_1, \ldots, h_M)$. We first find the vertices set $\mathcal{A}^*$ of $\mathcal{D}_\mu^*$ using the above theorem and then take the image $\mathcal{A}$ of $\mathcal{A}^*$ under the projection P. We know that $\mathcal{A}$ is a subset of $\mathcal{D}_\mu$ and includes all vertices of $\mathcal{D}_\mu$.

Let

$$A = \begin{bmatrix} 0 & 1 & \cdots & M \\ 1 & 1 & \cdots & 1 \end{bmatrix}, \quad b = \begin{bmatrix} \mu \\ 1 \end{bmatrix}.$$

We have

$$\mathcal{D}_\mu^* = \{h \in (\mathbb{R}^+)^{M+1} : Ah = b\}.$$

All the vertices of $\mathcal{D}_\mu^*$ are given as follows. For all integers i, j such that $i < \mu$ and $j \geq \mu$, vector $(h_0, h_1, \ldots, h_M) \in (\mathbb{R}^+)^{M+1}$ with $h_i = \frac{j-\mu}{j-i}$ and $h_j = \frac{\mu - i}{j-i}$ is a vertex of $\mathcal{D}_\mu^*$.

APPENDIX D

Proofs about Finite-length Analysis

D.1 PROOF OF THEOREM 7.1

D.1.1 INITIAL STATUS OF BP DECODING

The subscripts of $R_n^{(t)}$ and $C_n^{(t)}$ are omitted in this proof. Let $\bar{\Theta}_s^{(t)}$ be the set of indices of batches that both the degree and the rank at time t equal to s. In other words, a batch with index in $\bar{\Theta}_s^{(t)}$, $s > 0$, is decodable and can decode s symbols. Let $\Theta^{(t)}$ be the set of indices of batches that are not in $\bar{\Theta}^{(t)} \triangleq \cup_{s=0}^{M} \bar{\Theta}_s^{(t)}$. We see that $R^{(t)} = |\cup_{i\in\bar{\Theta}^{(t)}} A_i^{(t)}|$, which is valid since $A_i^{(t)} = \emptyset$ for $i \in \bar{\Theta}_0^{(t)}$. Also, we see that $C^{(t)} = |\Theta^{(t)}|$.

We first calculate $\mathbf{\Lambda}_n^{(0)}[c, r] = \Pr\{C^{(0)} = c, R^{(0)} = r\}$. When $t = 0$, a batch with degree s has the probability Ψ_s and is decodable with probability $\hbar'_s$ (see (2.7) for the definition of $\hbar'_s$). Therefore, the probability that a batch is in $\bar{\Theta}_s^{(0)}$ is $\Psi_s\hbar'_s$, i.e., for $1 \le i \le n$ and $0 \le s \le M$,

$$\Pr\left\{i \in \bar{\Theta}_s^{(0)}\right\} = p_{0,s} \triangleq \Psi_s\hbar'_s.$$

Hence,

$$\Pr\left\{i \in \bar{\Theta}^{(0)}\right\} = \sum_{s=0}^{M} p_{0,s} \triangleq \rho_0. \tag{D.1}$$

Since all batches are independently generated, we have

$$\Pr\left\{C^{(0)} = k\right\} = \Pr\left\{|\Theta^{(0)}| = k\right\} = \mathrm{Bi}(k; n, 1-\rho_0). \tag{D.2}$$

When $\rho_0 = 0$, $\Pr\{C^{(0)} = n, R^{(0)} = 0\} = 1$ and the formula in (7.3) holds. Henceforth in this subsection, we assume $\rho_0 > 0$. Recall $\mathbf{Q_0}$ defined in (7.5).

Lemma D.1 *We have for* $k = 0, 1, \ldots, n$,

$$\left(\Pr\left\{R^{(0)} = j \,|C^{(0)} = n - k\right\} : j = 0, \ldots, K\right) = \mathbf{e}_0\mathbf{Q}_0^k,$$

where $\mathbf{e}_0 = (1, 0, \ldots, 0)$.

Proof. Fix n. If $k = 0$, then $\bar{\Theta}^{(0)} = \emptyset$, and hence $\Pr\{R^{(0)} = 0 | C^{(0)} = n\} = 1$, i.e., the lemma with $k = 0$ is proved. Henceforth, we assume $k > 0$. The condition $C^{(0)} = n - k$ means that k batches becomes decodable at time 0. Suppose that $\bar{\Theta}^{(0)} = \{1, \ldots, k\}$, which does not change the distribution of $R^{(0)}$. Define $Z_0 \equiv 0$ as a constant random variable on $\{0, 1, \ldots, K\}$, and for $r = 1, \ldots, k$ define $Z_r = |\cup_{m=1}^{r} A_m|$. These random variables are defined under the condition $\{\bar{\Theta}^{(0)} = \{1, \ldots, k\}\} \triangleq E$. Note that $Z_k = R^{(0)}$. Since the contributors of each batch are independently chosen, $Z_0, \ldots, Z_k$ forms a Markov chain. Specifically, for $j < i$, $\Pr\{Z_r = j | Z_{r-1} = i\} = 0$ and for $j \geq i$,

$$\begin{aligned}\Pr\{Z_r = j | Z_{r-1} = i\} &= \Pr\left\{|\cup_{m=1}^{r} A_m| = j \,\middle|\, |\cup_{m=1}^{r-1} A_m| = j, E\right\} \\ &= \sum_{s=j-i}^{j} \underbrace{\Pr\left\{|\cup_{m=1}^{r} A_m| = j \,\middle|\, |\cup_{m=1}^{r-1} A_m| = i, |A_r| = s, E\right\}}_{(a)} \\ &\quad \times \underbrace{\Pr\left\{|A_r| = s \,\middle|\, |\cup_{m=1}^{r-1} A_m| = i, E\right\}}_{(b)}.\end{aligned}$$

Term (a) is a hypergeometric distribution $\mathrm{hyge}(s - j + i; K, i, s)$. Term (b) is equal to $\Pr\left\{|A_r| = s \middle| r \in \bar{\Theta}^{(0)}\right\} = \frac{p_{0,s}}{\rho_0}$ for $s \leq M$ and zero otherwise. Overall, we have $\Pr\{Z_r = j | Z_{r-1} = i\} = \mathbf{Q}_0[i, j]$, independent of r. Therefore, $Z_0, \ldots, Z_k$ forms a homogeneous Markov chain with transition matrix $\mathbf{Q}_0$. The proof is completed by noting that $\mathbf{e}_0$ is the probability vector corresponding to the distribution of Z_0. □

By (D.2) and Lemma D.1, we have

$$\begin{aligned}\mathbf{\Lambda}_n^{(0)}[c, :] &= (\Pr\{C^{(0)} = c, R^{(0)} = j\} : j = 0, \ldots, K) \\ &= \Pr\{C^{(0)} = c\}(\Pr\{R^{(0)} = j | C^{(0)} = c\} : j = 0, \ldots, K) \\ &= \mathrm{Bi}(c; n, 1 - \rho_0)\mathbf{e}_0\mathbf{Q}_0^{n-c},\end{aligned}$$

which proves (7.3).

D.1.2 RECURSIVE FORMULA

Consider $t > 0$ and we prove the recursion of $\mathbf{\Lambda}_n^{(t)}$ in (7.4). Define event E_t as $\{R^{(\tau)} > 0, \tau < t\}$, i.e.,

$$E_t = \left\{\cup_{s=1}^{M} \bar{\Theta}_s^{(\tau)} \neq \emptyset, \tau < t\right\}.$$

We have for $t > 0$

$$\begin{aligned}
\mathbf{\Lambda}_n^{(t)}[c,r] &= \Pr\left\{C^{(t)} = c, R^{(t)} = r, R^{(\tau)} > 0, \tau < t\right\} \\
&= \sum_{c',r'>0} \Pr\left\{C^{(t)} = c, C^{(t-1)} = c', R^{(t)} = r, R^{(t-1)} = r', R^{(\tau)} > 0, \tau < t\right\} \\
&= \sum_{c',r'>0} \Pr\left\{C^{(t)} = c, R^{(t)} = r, |C^{(t-1)} = c', R^{(t-1)} = r', R^{(\tau)} > 0, \tau < t-1\right\} \\
&\quad \cdot \mathbf{\Lambda}_n^{(t-1)}[c',r'] \\
&= \sum_{c',r'>0} \underbrace{\Pr\left\{R^{(t)} = r | C^{(t)} = c, C^{(t-1)} = c', R^{(t-1)} = r', E_{t-1}\right\}}_{(c)} \times \\
&\quad \times \underbrace{\Pr\left\{C^{(t)} = c | C^{(t-1)} = c', R^{(t-1)} = r', E_{t-1}\right\}}_{(d)} \mathbf{\Lambda}_n^{(t-1)}[c',r'].
\end{aligned}$$

We characterize (c) and (d) in the above equation respectively. Recall that for $t \geq 1$

$$\begin{aligned}
p_{t,s} &\triangleq \hbar_s \sum_{d=s+1}^{D} \Psi_d \frac{d}{K} \mathrm{hyge}(d-s-1; K-1, d-1, t-1), \\
\rho_t &\triangleq \frac{\sum_s p_{t,s}}{1 - \sum_{\tau=0}^{t-1} \sum_s p_{\tau,s}}.
\end{aligned}$$

Lemma D.2 *For $r' > 0$ and $c' \geq c$,*

$$\Pr\left\{C^{(t)} = c | C^{(t-1)} = c', R^{(t-1)} = r', E_{t-1}\right\} = \mathrm{Bi}(c; c', 1-\rho_t).$$

Proof. Under the condition of $R^{(t-1)} = r' > 0$ and E_{t-1}, the BP decoding does not stop at time $t-1$. Note that if $c' = 0$, i.e., all the batches are decodable at time $t-1$, then $C^{(t)} = 0$ with probability one. We henceforth assume $c' > 0$ in this proof. Since $\Theta^{(0)} \supset \Theta^{(1)} \supset \cdots \supset \Theta^{(t-1)}$, we have $C^{(\tau)} > 0$ for $\tau = 0, 1, \ldots, t-1$. We consider a special instance of the condition $C^{(t-1)} = c'$, $R^{(t-1)} = r'$ and E_{t-1} such that the input symbol decoded from time $\tau - 1$ to τ has index $\tau - 1$ for $1 \leq \tau \leq t$, and study the probability of $j \in \bar{\Theta}_s^{(\tau)} \cap \Theta^{(\tau-1)}$ under this instance. Since the probability to be obtain does not depend on the instance, the probability is equal to the probability of the lemma. To simplify the notation, the condition $C^{(t-1)} = c'$, $R^{(t-1)} = r'$ and E_{t-1} is omitted in the remainder of the proof.

For $\tau = 1, \ldots, t$, we first study $\Pr\{j \in \bar{\Theta}_s^{(\tau)} \cap \Theta^{(\tau-1)}\}$ for an arbitrary batch j. There are totally τ input symbols decoded at time τ, where $\tau - 1$ is the index of the input symbol decoded at the step from $\tau - 1$ to τ. Given the initial degree of batch j being d, $j \in \bar{\Theta}_s^{(\tau)} \cap \Theta^{(\tau-1)}$ is equivalent to

(1) $\tau - 1 \in A_j$,

(2) $|A_j^{(\tau)}| = s$, and

(3) $\mathrm{rk}(\mathbf{G}_j^{(\tau-1)}\mathbf{H}_j) = \mathrm{rk}(\mathbf{G}_j^{(\tau)}\mathbf{H}_j) = s$.

Since all batches are formed independently, we know that (1) holds with probability d/K; given (1) the probability that (2) holds is the hypergeometric distribution $\mathrm{hyge}(d-s-1; K-1, \tau-1, d-1)$; given both (1) and (2) the probability that (3) holds is $\hbar_s$ (see (2.6)). Therefore, the probability for (1), (2) and (3) to hold given $|A_j| = d$ is

$$\frac{d}{K}\hbar_s \mathrm{hyge}(d-s-1; K-1, \tau-1, d-1).$$

Hence, after considering the distribution of the degree,

$$\Pr\left\{j \in \bar{\Theta}_s^{(\tau)} \cap \Theta^{(\tau-1)}\right\} = p_{\tau,s}. \tag{D.3}$$

Now we study $\Pr\{j \in \Theta^{(\tau)}\}$. Since $\Theta^{(\tau)}, \bar{\Theta}_s^{(\tau)} \cap \Theta^{(\tau-1)}, s = 0, 1, \ldots, M$ forms a partition of $\Theta^{(\tau-1)}$,

$$\begin{aligned}\Pr\left\{j \in \Theta^{(\tau-1)}\right\} &= \Pr\left\{j \in \Theta^{(\tau)}\right\} + \sum_{s=0}^{M} \Pr\left\{j \in \bar{\Theta}_s^{(\tau)} \cap \Theta^{(\tau-1)}\right\} \\ &= \Pr\left\{j \in \Theta^{(\tau)}\right\} + \sum_{s=0}^{M} p_{\tau,s}.\end{aligned}$$

Using $\Pr\{j \in \Theta^{(0)}\} = 1 - \sum_{s=0}^{M} p_{0,s}$ (see (D.1)), we obtain that

$$\Pr\left\{j \in \Theta^{(\tau)}\right\} = 1 - \sum_{\tau'=0}^{\tau}\sum_{s=0}^{M} p_{\tau',s}.$$

Hence we have

$$\Pr\left\{j \in \bar{\Theta}^{(t)} | j \in \Theta^{(t-1)}\right\} = \frac{\Pr\{j \in \bar{\Theta}^{(t)} \cap \Theta^{(t-1)}\}}{\Pr\{j \in \Theta^{(t-1)}\}} = \rho_t. \tag{D.4}$$

In other words, for a batch in $\Theta^{(t-1)}$, it would stay in $\Theta^{(t)}$ with probability $1 - \rho_t$. Since batches in $\Theta^{(t-1)}$ stay in $\Theta^{(t)}$ independently, for $B \subset \{1, \ldots, n\}$ with $|B| = c'$,

$$\begin{aligned}\Pr\left\{C^{(t)} = c | \Theta^{(t-1)} = B, R^{(t-1)} = r', E_{t-1}\right\} &= \Pr\left\{|\Theta^{(t)}| = c | \Theta^{(t-1)} = B, R^{(t-1)} = r', E_{t-1}\right\} \\ &= \mathrm{Bi}(c; c', 1-\rho_t).\end{aligned}$$

Since the above distribution depends on B only through its cardinality, we have

$$\begin{aligned}
&\Pr\left\{C^{(t)} = c | C^{(t-1)} = c', R^{(t-1)} = r', E_{t-1}\right\} \\
&= \sum_{B \subset \{1,\ldots,n\}: |B| = c'} \Pr\{C^{(t)} = c | \Theta^{(t-1)} = B, R^{(t-1)} = r', E_{t-1}\} \\
&\quad \cdot \Pr\{\Theta^{(t-1)} = B | C^{(t-1)} = c', R^{(t-1)} = r', E_{t-1}\} \\
&= \mathrm{Bi}(c; c', 1-\rho_t).
\end{aligned}$$

The proof of the lemma is completed. □

Assume $\sum_{s=0}^{M} p_{t,s} > 0$, which holds when BP decoding can start (see Lemma 7.3).

Lemma D.3 *For $r' > 0$ and $c' \geq c$,*

$$\Pr\left\{R^{(t)} = r | C^{(t)} = c, C^{(t-1)} = c', R^{(t-1)} = r', E_{t-1}\right\} = (\mathbf{Q}_t^{c'-c})[r'-1, r].$$

Proof. First, if $c = c'$, then $\mathbf{Q}_t^{c'-c}$ is the identity matrix, and no batches become decodable for the first time at time t. Therefore, $R^{(t)} = R^{(t-1)} - 1$, which proves the lemma with $c = c'$. Henceforth, we assume $c' > c$. Consider an instance of $\{C^{(t)} = c, C^{(t-1)} = c', R^{(t-1)} = r', E_{t-1}\}$ with $\Theta^{(t-1)} \setminus \Theta^{(t)} = \{1, \ldots, c'-c\}$. We will compute the distribution of $R^{(t)}$ by assuming this instance. Since the distribution we will obtain only depends on the instance through c, c' and r', the distribution of $R^{(t)}$ under the condition $\{C^{(t)} = c, C^{(t-1)} = c', R^{(t-1)} = r', E_{t-1}\}$ is the same.

Let $\mathcal{A}$ be the set of indices of decodable input symbols at time $t-1$, excluding the input symbol decoded from time $t-1$ to t. We have $|\mathcal{A}| = r' - 1$, which is valid since $r' > 0$. Since batches with index in $B' \setminus B$ become decodable only starting at time t, we have $R^{(t)} = |\mathcal{A} \cup (\cup_{i=1}^{\delta} A_i^{(t)})|$. We use a similar method as in Lemma D.1 to compute the distribution of $R^{(t)}$. Define $Z_0 \equiv |\mathcal{A}|$ as a constant random variable on $\{0, 1, \ldots, K-t\}$, and for $r = 1, \ldots, c'-c$ define $Z_r = |\mathcal{A} \cup_{m=1}^{r} A_m|$. Note that $Z_{c'-c} = R^{(t)}$. Since the contributors of each batch are independently chosen, $Z_0, \ldots, Z_{c'-c}$ forms a Markov chain. Specifically, for $j < i$, $\Pr\{Z_r = j | Z_{r-1} = i\} = 0$ and for $j \geq i$,

$$\begin{aligned}
\Pr\{Z_r = j | Z_{r-1} = i\} &= \Pr\left\{|\mathcal{A} \cup (\cup_{m=1}^{r} A_m^{(t)})| = j \,\middle|\, |\mathcal{A} \cup (\cup_{m=1}^{r-1} A_m^{(t)})| = i\right\} \\
&= \sum_{s=j-i}^{j} \underbrace{\Pr\left\{|A_r^{(t)}| = s \,\middle|\, |\mathcal{A} \cup (\cup_{m=1}^{r-1} A_m^{(t)})| = i\right\}}_{(e)} \\
&\quad \times \underbrace{\Pr\left\{|\mathcal{A} \cup (\cup_{m=1}^{r} A_m^{(t)})| = j \,\middle|\, |\mathcal{A} \cup (\cup_{m=1}^{r-1} A_m^{(t)})| = i, |A_r^{(t)}| = s\right\}}_{(f)}.
\end{aligned}$$

Term (e) is equal to $\Pr\{r \in \bar{\Theta}_s^{(t)} | r \in \Theta^{(t-1)} \cap \bar{\Theta}^{(t)}\} = \frac{p_{t,s}}{\sum_s p_{t,s}}$ (see (D.3)) for $s \le M$. Term (f) is a hypergeometric distribution $\mathrm{hyge}(s-j+i; K-t, i, s)$. Overall, we have $\Pr\{Z_r = j | Z_{r-1} = i\} = \mathbf{Q}_t[i,j]$, independent of r. Therefore, $Z_0, \ldots, Z_{c'-c}$ forms a homogeneous Markov chain with transition matrix $\mathbf{Q}_t$. The proof is completed by considering the transition matrix from Z_0 to $Z_{c'-c}$. □

Now we are ready to complete the proof of Theorem 7.1. With the above two lemmas, we can write

$$\begin{aligned}\mathbf{\Lambda}_n^{(t)}[c,:] &= \sum_{c',r'>0} (\mathbf{Q}_t^{c'-c})[r'-1,:]\mathrm{Bi}(c;c',1-\rho_t)\mathbf{\Lambda}_n^{(t-1)}[c',r'] \\ &= \sum_{c'\ge c} \mathrm{Bi}(c;c',1-\rho_t)\mathbf{\Lambda}_n^{(t-1)}[c',1:]\mathbf{Q}_t^{c'-c}.\end{aligned}$$

This completes the proof of Theorem 7.1.

D.2 PROOFS OF SEVERAL PROPERTIES

Proof of Lemma 7.4. The first claim can be proved by induction over t. First $1-\rho_0 = 1-p_0$ by definition. Suppose that (1) holds for certain $t \ge 0$. We have $\prod_{\tau=0}^{t+1}(1-\rho_\tau) = (1-\rho_{t+1})(1-\sum_{\tau=0}^{t} p_\tau) = 1-\sum_{\tau=0}^{t+1} p_\tau$, where the first equality follows by the induction hypothesis and the second equality follows the definition of ρ_t. To prove the second claim, we have $\rho_t \prod_{\tau=0}^{t-1}(1-\rho_\tau) = \rho_t(1-\sum_{\tau=0}^{t-1} p_\tau) = p_t$, where the first equality follows by (1) and the second equality follows the definition of ρ_t. □

Proof of Lemma 7.10. We first prove the formula of $\mathbf{U}_t^{-1}$. Let $\mathbf{U}'_t$ be an upper-triangular matrix with $\mathbf{U}'_t[i,j] = (-1)^{j-i}\binom{K-t-i}{j-i}$ for $i \le j$. We check that $\mathbf{U}_t\mathbf{U}'_t = \mathbf{I}$. We write

$$(\mathbf{U}_t\mathbf{U}'_t)[i,j] = \sum_{k=i}^{j} \mathbf{U}_t[i,k]\mathbf{U}'_t[k,j]. \tag{D.5}$$

When $i=j$, it is clear that $(\mathbf{U}_t\mathbf{U}'_t)[i,i] = 1$. Since $\mathbf{U}_t\mathbf{U}'_t$ is upper triangular, we verify that $(\mathbf{U}_t\mathbf{U}'_t)[i,j] = 0$ for $j > i$. Expanding the RHS of (D.5), we get

$$\begin{aligned}(\mathbf{U}_t\mathbf{U}'_t)[i,j] &= \sum_{k=i}^{j}\binom{K-t-i}{k-i}(-1)^{j-k}\binom{K-t-k}{j-k} \\ &= \binom{K-t-i}{j-i}\sum_{k=i}^{j}(-1)^{j-k}\binom{j-i}{k-i} \\ &= \binom{K-t-i}{j-i}\sum_{k=0}^{j-i}(-1)^{j-i-k}\binom{j-i}{k} \\ &= 0.\end{aligned}$$

Therefore, $\mathbf{U}_t^{-1} = \mathbf{U}_t'$.

To complete the proof, we need to verify the equality $\mathbf{Q}_t = \mathbf{U}_t\mathbf{D}_t\mathbf{U}_t^{-1}$. Write

$$\begin{aligned}(\mathbf{U}_t\mathbf{D}_t\mathbf{U}_t^{-1})[i,j] &= \sum_{k=i}^{j}\binom{K-t-i}{k-i}\mathbf{Q}_t[k,k](-1)^{j-k}\binom{K-t-k}{j-k}\\ &= \binom{K-t-i}{j-i}\sum_{k=i}^{j}(-1)^{j-k}\mathbf{Q}_t[k,k]\binom{j-i}{k-i}.\end{aligned}$$

When $i = j$, it is clear that $(\mathbf{U}_t\mathbf{D}_t\mathbf{U}_t^{-1})[i,i] = \mathbf{Q}_t[i,i]$. Since $\mathbf{U}_t\mathbf{D}_t\mathbf{U}_t^{-1}$ is upper triangular, we consider $j > i$ henceforth. By the definition of $\mathbf{Q}_t[k,k]$, we have

$$\begin{aligned}\sum_{k=i}^{j}(-1)^{j-k}\mathbf{Q}_t[k,k]\binom{j-i}{k-i} &= \sum_{k=i}^{j}(-1)^{j-k}\binom{j-i}{k-i}\sum_{s=0}^{k\wedge M}\frac{p_{t,s}}{p_t}\frac{\binom{k}{s}}{\binom{K-t}{s}}\\ &= \sum_{s=0}^{j\wedge M}\frac{p_{t,s}}{p_t\binom{K-t}{s}}\sum_{k=i\vee s}^{j}(-1)^{j-k}\binom{j-i}{k-i}\binom{k}{s}.\end{aligned}$$

In the following, we show that

$$\sum_{k=i\vee s}^{j}(-1)^{j-k}\binom{j-i}{k-i}\binom{k}{s} = \begin{cases}\binom{i}{s-j+i} & j-i\le s\le j,\\ 0 & s<j-i,\end{cases} \tag{D.6}$$

which completes the proof that $(\mathbf{U}_t\mathbf{D}_t\mathbf{U}_t^{-1})[i,j] = \mathbf{Q}_t[i,j]$.

The proof of (D.6) using binomial coefficients with negative integers. We write

$$\begin{aligned}\sum_{k=i\vee s}^{j}(-1)^{j-k}\binom{j-i}{k-i}\binom{k}{s} &= \sum_{k=i\vee s}^{j}(-1)^{j-k}\binom{j-i}{j-k}\binom{k}{k-s}\\ &= \sum_{k=i\vee s}^{j}(-1)^{j-k}\binom{j-i}{j-k}(-1)^{k-s}\binom{-s-1}{k-s}\\ &= (-1)^{j-s}\sum_{k=i\vee s}^{j}\binom{j-i}{j-k}\binom{-s-1}{k-s}\\ &= (-1)^{j-s}\binom{j-i-s-1}{j-s},\end{aligned}$$

where the last equality is obtained by Vandermonde's identity by considering the two cases $i < s$ and $i \ge s$. Note that when $s < j-i$, $\binom{j-i-s-1}{j-s} = 0$. Otherwise,

$$(-1)^{j-s}\binom{j-i-s-1}{j-s} = \binom{i}{j-s}.$$

The proof of the lemma is completed. □

D.3 PROOFS ABOUT STOPPING TIME DISTRIBUTION

Proof of Theorem 7.8. We will show that for $1 \le c \le n$ and $t \ge 0$,

$$\mathbf{\Lambda}_n^{(t)}[c,:] = \frac{n}{c}\prod_{i=0}^{t}(1-\rho_i)\mathbf{\Lambda}_{n-1}^{(t)}[c-1,:]. \tag{D.7}$$

By expanding the above recursive formula, we have for $c \ge 0$ and $t \ge 0$,

$$\mathbf{\Lambda}_n^{(t)}[c,:] = \binom{n}{c}\prod_{i=0}^{t}(1-\rho_i)^c\mathbf{\Lambda}_{n-c}^{(t)}[0,:]. \tag{D.8}$$

Substituting (D.8) into (7.2) and by Lemma 7.4, we get

$$P_{\text{stop}}(t\,|n) = \sum_{c=0}^{n}\binom{n}{c}\prod_{i=0}^{t}(1-\rho_i)^c\mathbf{\Lambda}_{n-c}^{(t)}[0,0] = \sum_{c=0}^{n}\binom{n}{c}\left(1-\sum_{\tau=0}^{t}p_\tau\right)^c\mathbf{\Lambda}_{n-c}^{(t)}[0,0],$$

proving (7.6). Further, (7.7) is obtained by (7.3) for $c = 0$. To prove (7.8), we have

$$\begin{aligned}\mathbf{\Lambda}_n^{(t)}[0,:] &= \sum_{c=0}^{n}\rho_t^c\mathbf{\Lambda}_n^{(t-1)}[c,1:]\mathbf{Q}_t^c\\ &= \sum_{c=0}^{n}\binom{n}{c}\rho_t^c\prod_{i=0}^{t-1}(1-\rho_i)^c\mathbf{\Lambda}_{n-c}^{(t-1)}[0,1:]\mathbf{Q}_t^c\\ &= \sum_{c=0}^{n}\binom{n}{c}p_t^c\mathbf{\Lambda}_{n-c}^{(t-1)}[0,1:]\mathbf{Q}_t^c\end{aligned}$$

where the first equality follows from (7.4) with $c = 0$, the second equality is obtained by substituting (D.8), and the last step is obtained by applying Lemma 7.4.

Now we prove (D.7) by induction. When $t = 0$, we have by Theorem 7.1 that

$$\begin{aligned}\mathbf{\Lambda}_n^{(0)}[c,:] &= \text{Bi}(c;n,1-\rho_0)\mathbf{Q}_0^{n-c}[0,:]\\ &= \frac{n}{c}(1-\rho_0)\text{Bi}(c-1;n-1,1-\rho_0)\mathbf{Q}_0^{(n-1)-(c-1)}[0,:]\\ &= \frac{n}{c}(1-\rho_0)\mathbf{\Lambda}_{n-1}^{(0)}[c-1,:].\end{aligned} \tag{D.9}$$

Suppose that (D.7) holds for $t \geq 0$. Applying the recursive formula of Theorem 7.1, we can show that

$$\begin{aligned}
\mathbf{\Lambda}_{n+1}^{(t)}[c,:] &= \sum_{c'=c}^{n+1} \mathrm{Bi}(c;c',1-\rho_t)\mathbf{\Lambda}_{n+1}^{(t-1)}[c',1:]\mathbf{Q}_t^{c'-c} \\
&= \sum_{c'=c}^{n+1} \mathrm{Bi}(c;c',1-\rho_t)\frac{n+1}{c'}\prod_{i=0}^{t-1}(1-\rho_i)\mathbf{\Lambda}_n^{(t-1)}[c'-1,1:]\mathbf{Q}_t^{c'-c} \\
&= \frac{n+1}{c}\prod_{i=0}^{t}(1-\rho_i)\sum_{c'=c}^{n+1}\mathrm{Bi}(c-1;c'-1,1-\rho_t)\mathbf{\Lambda}_n^{(t-1)}[c'-1,1:]\mathbf{Q}_t^{c'-c} \\
&= \frac{n+1}{c}\prod_{i=0}^{t}(1-\rho_i)\sum_{c''=c-1}^{n}\mathrm{Bi}(c-1;c'',1-\rho_t)\mathbf{\Lambda}_n^{(t-1)}[c'',1:]\mathbf{Q}_t^{c''-(c-1)} \\
&= \frac{n+1}{c}\prod_{i=0}^{t}(1-\rho_i)\mathbf{\Lambda}_n^{(t)}[c-1,:].
\end{aligned}$$

The proof is completed. □

Proof of Theorem 7.11. We first show

$$\mathbf{\Lambda}_n^{(t)}[0,:] = \sum_{i=0}^{2^t-1}\mathbf{V}_{t,i}\mathbf{\Delta}_{t,i}^n\mathbf{U}_t^{-1} \tag{D.10}$$

by induction in t. The claim for $t=0$ can be shown by replacing $p_0\mathbf{Q}_0$ in (7.7) with the decomposition in Lemma 7.10. Suppose that the claim of the theorem holds for certain $t \geq 0$. Substituting this form of $\mathbf{\Lambda}_n^t$ into (7.8) with $t+1$ in place of t, we obtain

$$\begin{aligned}
\mathbf{\Lambda}_n^{(t+1)} &= \sum_{c=0}^{n}\binom{n}{c}\sum_{i=0}^{2^t-1}\mathbf{V}_{t,i}\mathbf{\Delta}_{t,i}^n\mathbf{U}_t^{-1}[:,1:](p_{t+1}\mathbf{Q}_{t+1})^c \\
&= \sum_{i=0}^{2^t-1}\mathbf{V}_{t,i}\sum_{c=0}^{n}\binom{n}{c}\mathbf{\Delta}_{t,i}^{n-c}\mathbf{U}_t^{-1}[:,1:]\mathbf{U}_{t+1}(p_{t+1}\mathbf{D}_{t+1})^c\mathbf{U}_{t+1}^{-1}.
\end{aligned}$$

Using the same technique as proving (D.6), we can verify that

$$\mathbf{U}_t^{-1}[:,1:]\mathbf{U}_{t+1} = \begin{bmatrix}-\mathbf{U}_t[0,1:]\\ \mathbf{I}\end{bmatrix} = \begin{bmatrix}-\mathbf{U}_t[0,1:]\\ \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \mathbf{I}\end{bmatrix}. \tag{D.11}$$

Substituting the above equation into (D.11), we get

$$\begin{aligned}
\mathbf{\Lambda}_n^{(t+1)} &= \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i} \sum_{c=0}^{n} \binom{n}{c} \mathbf{\Delta}_{t,i}^{n-c} \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} (p_{t+1}\mathbf{D}_{t+1})^c \mathbf{U}_{t+1}^{-1} \\
&\quad + \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i} \sum_{c=0}^{n} \binom{n}{c} \mathbf{\Delta}_{t,i}^{n-c} \begin{bmatrix} -\mathbf{U}_t[0,1:] \\ \mathbf{0} \end{bmatrix} (p_{t+1}\mathbf{D}_{t+1})^c \mathbf{U}_{t+1}^{-1} \\
&= \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[1:] \sum_{c=0}^{n} \binom{n}{c} (\mathbf{\Delta}_{t,i}[1:,1:])^{n-c} (p_{t+1}\mathbf{D}_{t+1})^c \mathbf{U}_{t+1}^{-1} \\
&\quad - \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0]\mathbf{U}_t[0,1:] \sum_{c=0}^{n} \binom{n}{c} (\mathbf{\Delta}_{t,i}[0,0])^{n-c} (p_{t+1}\mathbf{D}_{t+1})^c \mathbf{U}_{t+1}^{-1} \qquad \text{(D.12)} \\
&= \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[1:] \left(\mathbf{\Delta}_{t,i}[1:,1:] + p_{t+1}\mathbf{D}_{t+1}\right)^n \mathbf{U}_{t+1}^{-1} \\
&\quad - \sum_{i=0}^{2^t-1} \mathbf{U}_t[0,1:]\mathbf{V}_{t,i}[0] \left(\mathbf{\Delta}_{t,i}[0,0]\mathbf{I} + p_{t+1}\mathbf{D}_{t+1}\right)^n \mathbf{U}_{t+1}^{-1}, \qquad \text{(D.13)}
\end{aligned}$$

where (D.12) is obtained by noting $\mathbf{\Delta}_{t,i}$ is diagonal and (D.13) is obtained by combining the binomial terms. The proof of (D.10) is completed by checking the definition of $\mathbf{V}_{t+1,i}$ and $\mathbf{\Delta}_{t+1,i}$.

Substituting the formula of $\mathbf{\Lambda}_n^{(t)}[0,:]$ in (D.10) into (7.6), we get

$$\begin{aligned}
P_{\text{stop}}(t|n) &= \sum_{c=0}^{n} \binom{n}{c} \left(1 - \sum_{\tau=0}^{t} p_\tau\right)^c \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i} \mathbf{\Delta}_{t,i}^{n-c} \mathbf{U}_t^{-1}[:,0] \\
&= \sum_{c=0}^{n} \binom{n}{c} \left(1 - \sum_{\tau=0}^{t} p_\tau\right)^c \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0] \mathbf{\Delta}_{t,i}^{n-c}[0,0] \\
&= \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0] \sum_{c=0}^{n} \binom{n}{c} \left(1 - \sum_{\tau=0}^{t} p_\tau\right)^c (\mathbf{\Delta}_{t,i}[0,0])^{n-c} \\
&= \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0] \left(1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0]\right)^n,
\end{aligned}$$

where the second equality is obtained using the facts that (i) $\mathbf{\Delta}_{t,i}$ is diagonal, (ii) $\mathbf{U}_t^{-1}$ is upper-triangular, and (iii) $\mathbf{U}_t^{-1}[0,0] = 1$. □

Proof of Theorem 7.12. For $0 \le t \le K$ and $0 \le i \le K - t$, let

$$\lambda_{t,i} = p_t \mathbf{Q}_t[i,i] = p_t \mathbf{D}_t[i,i] = \sum_{s=0}^{i \wedge M} p_{t,s} \frac{\binom{i}{s}}{\binom{K-t}{s}}, \qquad \text{(D.14)}$$

with which we can rewrite

$$q_t = 1 - \sum_{\tau=0}^{t} p_\tau + \sum_{\tau=0}^{t} \lambda_{\tau,t-\tau}.$$

Using Lemma 7.3 and the definition of $\lambda_{t,j}$, we have that

$$\lambda_{t,j} = 0, \text{ when } 0 \le t < r_{\text{BP}}, t + j < r_{\text{BP}}; \tag{D.15}$$
$$\lambda_{t,j} > \lambda_{t,j-1}, \text{ when } 0 \le t < r_{\text{BP}}, t + j \ge r_{\text{BP}}; \tag{D.16}$$
$$0 < \lambda_{t,0} < \lambda_{t,1} < \ldots < \lambda_{t,K-t}, \text{ when } t \ge r_{\text{BP}}. \tag{D.17}$$

We further show inductively that for $i = 0, 1, \ldots, 2^t - 1$,

$$\mathbf{\Delta}_{t,i}[j, j] = 0, \text{ when } t + j < r_{\text{BP}}, \tag{D.18}$$
$$\mathbf{\Delta}_{t,i}[j, j] > \mathbf{\Delta}_{t,i}[j - 1, j - 1], \text{ when } t + j \ge r_{\text{BP}}. \tag{D.19}$$

By the definition of $\mathbf{\Delta}_{0,0}$ in Theorem 7.11, we write $\mathbf{\Delta}_{0,0}[j, j] = p_0\mathbf{D}_0[j, j] = \lambda_{0,j}$, which, together with (D.15)–(D.17) with $t = 0$, implies (D.18) and (D.19) for $t = 0$. Suppose that (D.18) and (D.19) hold for certain $t \ge 0$. By the recursive formula in Theorem 7.11, we have for $i = 0, 1, \ldots, 2^t - 1$,

$$\mathbf{\Delta}_{t+1,i}[j, j] = \mathbf{\Delta}_{t,i}[j + 1, j + 1] + p_{t+1}\mathbf{D}_{t+1}[j, j] = \mathbf{\Delta}_{t,i}[j + 1, j + 1] + \lambda_{t+1,j},$$
$$\mathbf{\Delta}_{t+1,2^t+i}[j, j] = \mathbf{\Delta}_{t,i}[0, 0] + p_{t+1}\mathbf{D}_{t+1}[j, j] = \mathbf{\Delta}_{t,i}[0, 0] + \lambda_{t+1,j}.$$

When $t + 1 + j < r_{\text{BP}}$, by the induction hypothesis, we have $\mathbf{\Delta}_{t,i}[j + 1, j + 1] = 0$ and $\mathbf{\Delta}_{t,i}[0, 0] = 0$, and by (D.15), we have $\lambda_{t+1,j} = 0$. Therefore, $\mathbf{\Delta}_{t+1,i}[j, j] = 0$ and $\mathbf{\Delta}_{t+1,2^t+i}[j, j] = 0$ when $t + 1 + j < r_{\text{BP}}$, which completes the proof of (D.18). When $t + 1 + j \ge r_{\text{BP}}$, by the induction hypothesis, we have $\mathbf{\Delta}_{t,i}[j + 1, j + 1] > \mathbf{\Delta}_{t,i}[j, j]$, and by (D.16) or (D.17), we have $\lambda_{t+1,j} > \lambda_{t+1,j-1}$. Therefore, $\mathbf{\Delta}_{t+1,i}[j, j] > \mathbf{\Delta}_{t+1,i}[j, j]$ and $\mathbf{\Delta}_{t+1,2^t+i}[j, j] > \mathbf{\Delta}_{t+1,2^t+i}[j, j]$ when $t + 1 + j \ge r_{\text{BP}}$, which completes the proof of (D.19).

Now we are ready to prove (i) and (ii) of the theorem. When $t = 0$, by Theorem 7.11 and $\lambda_{0,0} = 0$, we have $P_{\text{stop}}(0|n) = \mathbf{V}_{0,0}[0]\left(1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{0,0}[0, 0]\right)^n = q_0^n$, proving (i). When $1 \le t < r_{\text{BP}}$, by Theorem 7.11 and (D.18), $P_{\text{stop}}(t|n) = (1 - \sum_{\tau=0}^{t} p_\tau)^n \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0]$. To prove (ii), we show that for $t \ge 1$

$$\sum_{i=0}^{2^t-1} \mathbf{V}_{t,i} = \mathbf{0}. \tag{D.20}$$

When $t = 1$, we have

$$\sum_{i=0}^{1} \mathbf{V}_{1,i} = \mathbf{U}_0[0, 1:] - \mathbf{U}_0[0, 0]\mathbf{U}_0[0, 1:] = \mathbf{0}.$$

Suppose that (D.20) holds for certain $t \geq 1$. We have

$$\sum_{i=0}^{2^{t+1}-1} \mathbf{V}_{t+1,i} = \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[1:] - \mathbf{U}_t[0,1:] \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0] = \mathbf{0}.$$

Before proving (iii) of the theorem, we show by induction that for $i = 1, \ldots, 2^t - 1$,

$$\mathbf{\Delta}_{t,0}[j,j] > \mathbf{\Delta}_{t,i}[j,j], \text{ when } t + j \geq r_{\mathrm{BP}}. \tag{D.21}$$

The above inequality holds trivially for $t = 0$. Suppose that (D.21) holds for certain $t \geq 0$. When $t + 1 + j \geq r_{\mathrm{BP}}$, we have for $i = 0, 1, \ldots, 2^t - 1$,

$$\begin{aligned}
\mathbf{\Delta}_{t+1,0}[j,j] &= \mathbf{\Delta}_{t,0}[j+1, j+1] + p_{t+1}\mathbf{D}_{t+1}[j,j] \\
&\geq \mathbf{\Delta}_{t,i}[j+1, j+1] + p_{t+1}\mathbf{D}_{t+1}[j,j] = \mathbf{\Delta}_{t+1,i}[j,j] \\
&> \mathbf{\Delta}_{t,i}[0,0] + p_{t+1}\mathbf{D}_{t+1}[j,j] = \mathbf{\Delta}_{t+1,2^t+i}[j,j],
\end{aligned}$$

where the first inequality follows by the induction hypothesis with equality only when $i = 0$, and the second inequality follows from (D.18) and (D.19).

Now, we prove (iii) for $t \geq r_{\mathrm{BP}} \geq 1$. By (D.21), we know that for $i = 1, \ldots, 2^t - 1$,

$$\mathbf{\Delta}_{t,0}[0,0] > \mathbf{\Delta}_{t,i}[0,0],$$

and hence

$$\frac{1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0]}{q_t} = \frac{1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0]}{1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,0}[0,0]} < 1. \tag{D.22}$$

By Theorem 7.11 and noting that $\mathbf{V}_{t,0}[0] = \mathbf{U}_0[0,t] = \binom{K}{t} > 0$, we write

$$\begin{aligned}
&\lim_{n\to\infty} \frac{-\log P_{\mathrm{stop}}(t|n)}{n} \\
&= \lim_{n\to\infty} \frac{-\log q_t^n \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0](1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0])^n / q_t^n}{n} \\
&= -\log q_t + \lim_{n\to\infty} \frac{-\log \left(\mathbf{V}_{t,0}[0] + \sum_{i=1}^{2^t-1} \mathbf{V}_{t,i}[0](1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0])^n / q_t^n \right)}{n} \\
&= -\log q_t.
\end{aligned}$$

The proof is completed. □

D.4 PROOFS ABOUT POISSON NUMBER OF BATCHES

Proof of Theorem 7.16. Let $\bar{\mathbf{Q}}_t$ be a $(K+1) \times (K+1)$ matrix such that $\bar{\mathbf{Q}}_t[t:, t:] = \mathbf{Q}_t$, and all the other components of $\bar{\mathbf{Q}}_t$ are zero. For integers $n \geq 0$ and $t \geq 0$ define $(n+1) \times (K+1)$ matrix $\bar{\mathbf{\Lambda}}_n^{(t)}$ recursively as follows: (i) $\bar{\mathbf{\Lambda}}_n^{(0)} = \mathbf{\Lambda}_n^{(0)}$, and ii) for $t > 0$,

$$\bar{\mathbf{\Lambda}}_n^{(t)}[c,:] = \sum_{c'=c}^{n} \mathrm{Bi}(c; c', 1-\rho_t) \bar{\mathbf{\Lambda}}_n^{(t-1)}[c',:] \bar{\mathbf{Q}}_t^{c'-c}. \tag{D.23}$$

Note that compared with the iterative formula in Theorem 7.1, $\bar{\mathbf{\Lambda}}_n^{(t-1)}$ in the above formula is not shortened.

We show that

$$\bar{\mathbf{\Lambda}}_n^{(t)}[:,i] = \mathbf{\Lambda}_n^{(i)}[:,0], \quad i = 0,\dots,t, \tag{D.24}$$
$$\bar{\mathbf{\Lambda}}_n^{(t)}[:,t+1:] = \mathbf{\Lambda}_n^{(t)}[:,1:], \tag{D.25}$$

by induction in t. The claim holds for $t = 0$ by definition. Suppose that (D.24) and (D.25) hold for certain $t \geq 0$. We have by the definition that for $0 \leq c \leq n$,

$$\bar{\mathbf{\Lambda}}_n^{(t+1)}[c,:] = \bar{\mathbf{\Lambda}}_n^{(t)}[c,:] + \sum_{c'=c+1}^{n} \mathrm{Bi}(c;c',1-\rho_{t+1})\bar{\mathbf{\Lambda}}_n^{(t)}[c',:]\bar{\mathbf{Q}}_{t+1}^{c'-c}.$$

Since the first $t+1$ columns of $\bar{\mathbf{Q}}_{t+1}$ are all zero, we have for $i = 0,\dots,t$, $\bar{\mathbf{\Lambda}}_n^{(t+1)}[c,i] = \bar{\mathbf{\Lambda}}_n^{(t)}[c,i] = \mathbf{\Lambda}_n^{i}[c,0]$. Since the first $t+1$ rows of $\bar{\mathbf{Q}}_{t+1}$ are all zero, we can write

$$\begin{aligned}
\bar{\mathbf{\Lambda}}_n^{(t+1)}[c,t+1:] &= \bar{\mathbf{\Lambda}}_n^{(t)}[c,t+1:] + \sum_{c'=c+1}^{n} \mathrm{Bi}(c;c',1-\rho_{t+1})\bar{\mathbf{\Lambda}}_n^{(t)}[c',t+1:]\mathbf{Q}_{t+1}^{c'-c} \\
&= \sum_{c'=c}^{n} \mathrm{Bi}(c;c',1-\rho_{t+1})\mathbf{\Lambda}_n^{(t)}[c',1:]\mathbf{Q}_{t+1}^{c'-c} \\
&= \mathbf{\Lambda}_n^{(t+1)}[c,:],
\end{aligned}$$

where the second equality follows from the induction hypothesis and the last equality follows by Theorem 7.1.

Expending the recursive formula (D.23), we have

$$\begin{aligned}
\bar{\mathbf{\Lambda}}_n^{(t)}[c,:] &= \mathbf{e}_0 \sum \mathrm{Bi}(c_0;n,1-\rho_0)\bar{\mathbf{Q}}_0^{n-c_0}\times \\
&\quad \times \mathrm{Bi}(c_1;c_0,1-\rho_1)\bar{\mathbf{Q}}_1^{c_0-c_1} \times \cdots \times \mathrm{Bi}(c;c_{t-1},1-\rho_t)\bar{\mathbf{Q}}_t^{c_{t-1}-c} \\
&= \mathbf{e}_0 \sum \binom{n}{c_0}(1-\rho_0)^{c_0}(\rho_0\bar{\mathbf{Q}}_0)^{n-c_0}\times \\
&\quad \times \binom{c_0}{c_1}(1-\rho_1)^{c_1}(\rho_1\bar{\mathbf{Q}}_1)^{c_0-c_1} \times \cdots \times \binom{c_{t-1}}{c}(1-\rho_t)^{c}(\rho_t\bar{\mathbf{Q}}_t)^{c_{t-1}-c},
\end{aligned} \tag{D.26}$$

where the summation is over all $(c_0,\dots,c_{t-1})$ such that $n \geq c_0 \geq c_1 \geq \cdots \geq c_{t-1} \geq c$. Reorganizing (D.26) using Lemma 7.4, we obtain

$$\bar{\mathbf{\Lambda}}_n^{(t)}[c,:] = \mathbf{e}_0 \sum \binom{n}{c_0}\binom{c_0}{c_1}\cdots\binom{c_{t-1}}{c}\left(\frac{p_{t+1}}{\rho_{t+1}}\right)^{c}\left(p_0\bar{\mathbf{Q}}_0\right)^{n-c_0}\left(p_1\bar{\mathbf{Q}}_1\right)^{c_0-c_1}\cdots\left(p_t\bar{\mathbf{Q}}_t\right)^{c_{t-1}-c}.$$

Define

$$\check{\mathbf{\Lambda}}_{\bar{n}}^{(t)} = \sum_n \frac{\bar{n}^n}{n!}e^{-\bar{n}}\sum_c \bar{\mathbf{\Lambda}}_n^{(t)}[c,:].$$

By (7.16), (D.24), and (D.25), we have

$$\check{\mathbf{\Lambda}}_{\bar{n}}^{(t)}[t:] = \tilde{\mathbf{\Lambda}}_{\bar{n}}^{(t)}. \tag{D.27}$$

Substituting the expression of $\bar{\mathbf{\Lambda}}_{n}^{(t)}[c,:]$ and using the fact that

$$\binom{n}{c_0}\binom{c_0}{c_1}\cdots\binom{c_{t-1}}{c} = \frac{n!}{(n-c_0)!(c_0-c_1)!\cdots(c_{t-1}-c)!c!},$$

we have

$$\check{\mathbf{\Lambda}}_{\bar{n}}^{(t)} = \mathbf{e}_0 \sum e^{-\bar{n}} \frac{\left(\bar{n}\frac{p_{t+1}}{\rho_{t+1}}\right)^{c}}{c!} \frac{(\bar{n}\,p_0\bar{\mathbf{Q}}_0)^{n-c_0}}{(n-c_0)!} \frac{(\bar{n}\,p_1\bar{\mathbf{Q}}_1)^{c_0-c_1}}{(c_0-c_1)!} \cdots \frac{(\bar{n}\,p_t\bar{\mathbf{Q}}_t)^{c_{t-1}-c}}{(c_{t-1}-c)!},$$

where the summation is over all $(n, c_0, \ldots, c_{t-1}, c)$ such that $n \geq c_0 \geq c_1 \geq \cdots \geq c_{t-1} \geq c$.

Let $x_{t+1} = c$, $x_0 = n - c_0$, $x_t = c_{t-1} - c$ and $x_\tau = c_{\tau-1} - c_\tau$ for $1 \leq \tau \leq t-1$. We can rewrite the above expression as

$$\begin{aligned}
\check{\mathbf{\Lambda}}_{\bar{n}}^{(t)} &= \mathbf{e}_0 \sum_{x_\tau:\tau=0,\ldots,t+1} e^{-\bar{n}} \frac{\left(\bar{n}\frac{p_{t+1}}{\rho_{t+1}}\right)^{x_{t+1}}}{x_{t+1}!} \frac{(\bar{n}\,p_0\bar{\mathbf{Q}}_0)^{x_0}}{x_0!} \frac{(\bar{n}\,p_1\bar{\mathbf{Q}}_1)^{x_1}}{x_1!} \cdots \frac{(\bar{n}\,p_t\bar{\mathbf{Q}}_t)^{x_t}}{x_t!} \\
&= \mathbf{e}_0 e^{-\bar{n}} \sum_{x_{t+1}} \frac{\left(\bar{n}\frac{p_{t+1}}{\rho_{t+1}}\right)^{x_{t+1}}}{x_{t+1}!} \sum_{x_0} \frac{(\bar{n}\,p_0\bar{\mathbf{Q}}_0)^{x_0}}{x_0!} \sum_{x_1} \frac{(\bar{n}\,p_1\bar{\mathbf{Q}}_1)^{x_1}}{x_1!} \cdots \sum_{x_t} \frac{(\bar{n}\,p_t\bar{\mathbf{Q}}_t)^{x_t}}{x_t!} \\
&= \mathbf{e}_0 e^{-\bar{n}} \exp\left(\bar{n}\frac{p_{t+1}}{\rho_{t+1}}\right) \exp\left(\bar{n}\,p_0\bar{\mathbf{Q}}_0\right) \exp\left(\bar{n}\,p_1\bar{\mathbf{Q}}_1\right) \cdots \exp\left(\bar{n}\,p_t\bar{\mathbf{Q}}_t\right) \qquad \text{(D.28)} \\
&= \mathbf{e}_0 \exp\left(-\bar{n}\left(1 - \frac{p_{t+1}}{\rho_{t+1}}\right)\right) \exp\left(\bar{n}\,p_0\bar{\mathbf{Q}}_0\right) \exp\left(\bar{n}\,p_1\bar{\mathbf{Q}}_1\right) \cdots \exp\left(\bar{n}\,p_t\bar{\mathbf{Q}}_t\right) \\
&= \mathbf{e}_0 \exp\left(-\bar{n}\left(\sum_{\tau=0}^{t} p_\tau\right)\right) \exp\left(\bar{n}\,p_0\bar{\mathbf{Q}}_0\right) \exp\left(\bar{n}\,p_1\bar{\mathbf{Q}}_1\right) \cdots \exp\left(\bar{n}\,p_t\bar{\mathbf{Q}}_t\right), \qquad \text{(D.29)}
\end{aligned}$$

where (D.28) is obtained using the definition of matrix exponential, and (D.29) follows from the definition of ρ_t. Thus, we have

$$\check{\mathbf{\Lambda}}_{\bar{n}}^{(t)} = \exp\left(-\bar{n}\,p_t\right) \check{\mathbf{\Lambda}}_{\bar{n}}^{(t-1)} \exp\left(\bar{n}\,p_t\bar{\mathbf{Q}}_t\right)$$

with $\check{\mathbf{\Lambda}}_{\bar{n}}^{0} = \tilde{\mathbf{\Lambda}}_{\bar{n}}^{0}$ given in (7.19). The proof is complete by noting (D.27) and $\exp(\bar{n}\,p_t\bar{\mathbf{Q}}_t) = \begin{bmatrix} \mathbf{I} & \\ & \exp(\bar{n}\,p_t\mathbf{Q}_t) \end{bmatrix}$. □

Proof of Theorem 7.17. We prove the theorem using

$$\tilde{P}_{\text{stop}}(t|\bar{n}) = \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0] \exp\left(-\bar{n}\left(\sum_{\tau=0}^{t} p_\tau - \mathbf{\Delta}_{t,i}[0,0]\right)\right).$$

When $t = 0$, we have $\tilde{P}_{\text{stop}}(0|\bar{n}) = \mathbf{V}_{0,0}[0] \exp\left(-\bar{n}\left(p_0 - \mathbf{\Delta}_{0,0}[0,0]\right)\right) = \exp\left(-\bar{n}\left(p_0 - \lambda_{0,0}\right)\right) = \exp(-\bar{n}\, p_0)$, where the last equality follows from $\lambda_{0,0} = 0$ (see (D.15)). Hence (i) is proved by noting $q_0 = 1 - p_0$. When $1 \le t < r_{\text{BP}}$, since $\mathbf{\Delta}_{t,i}[0,0] = 0$ (see (D.18)), we have $\tilde{P}_{\text{stop}}(t|\bar{n}) = \exp\left(-\bar{n}\sum_{\tau=0}^{t} p_\tau\right)\sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0] = 0$, where the last equality follows from (D.20), proving (ii). To prove (iii), by (D.22) and $\mathbf{V}_{t,0}[0] = \mathbf{U}_0[0,t] = \binom{K}{t} > 0$, we write

$$\begin{aligned}
&\lim_{\bar{n}\to\infty} \frac{-\log \tilde{P}_{\text{stop}}(t|\bar{n})}{\bar{n}} \\
&= \lim_{\bar{n}\to\infty} \frac{-\log \exp(-\bar{n}(1-q_t)) \sum_{i=0}^{2^t-1} \mathbf{V}_{t,i}[0] \exp\left(-\bar{n}\left(\sum_{\tau=0}^{t} p_\tau - \mathbf{\Delta}_{t,i}[0,0] - 1 + q_t\right)\right)}{\bar{n}} \\
&= 1 - q_t.
\end{aligned}$$

The proof is completed. □

APPENDIX E

Proofs about Inactivation

Proof of Theorem 8.1. First, we have $\mathbf{\Lambda}_n^{(0)} = \mathbf{\Gamma}_n^{(0)}$ by their definitions, proving the formula for $t = 0$. For $t > 0$, define matrices $\mathbf{\Gamma}_n^{t(1)}$ and $\mathbf{\Gamma}_n^{t(2)}$ as

$$\mathbf{\Gamma}_n^{(t1)}[c, r] = \Pr\left\{\hat{C}^{(t)} = c, \hat{R}^{(t)} = r, \hat{R}^{(t-1)} > 0\right\}$$
$$\mathbf{\Gamma}_n^{(t2)}[c, r] = \Pr\left\{\hat{C}^{(t)} = c, \hat{R}^{(t)} = r, \hat{R}^{(t-1)} = 0\right\}.$$

Since

$$\mathbf{\Gamma}_n^{(t)} = \mathbf{\Gamma}_n^{(t1)} + \mathbf{\Gamma}_n^{(t2)},$$

we characterize the two terms on the RHS.

Write

$$\mathbf{\Gamma}_n^{(t1)}[c, r] = \sum_{c'} \sum_{r'>0} \underbrace{\Pr\{\hat{R}^{(t)} = r | \hat{C}^{(t)} = c, \hat{C}^{(t-1)} = c', \hat{R}^{(t-1)} = r'\}}_{(a)} \times \underbrace{\Pr\{\hat{C}^{(t)} = c | \hat{C}^{(t-1)} = c', \hat{R}^{(t-1)} = r'\}}_{(b)} \mathbf{\Gamma}_n^{(t-1)}[c', r'],$$

where term (a) and (b) can be obtained using Lemma D.3 and Lemma D.2, respectively, since only normal BP decoding is applied from time $t-1$ to t when $\hat{R}^{(t-1)} > 0$. Similar to the procedure for obtaining (D.5), we have

$$\mathbf{\Gamma}_n^{(t1)}[c, :] = \sum_{c' \geq c} \mathrm{Bi}(c; c', 1 - \rho_t) \mathbf{\Gamma}_n^{(t-1)}[c', 1:] \mathbf{Q}_t^{c'-c}. \tag{E.1}$$

The components in $\mathbf{\Gamma}_n^{t(2)}$ corresponds to the case that inactivation occurs during from time $t-1$ to time t, where an undecoded input symbol is marked as inactive and is treated as decoded. We write

$$\mathbf{\Gamma}_n^{(t2)}[c, r] = \sum_{c'} \underbrace{\Pr\{\hat{R}^{(t)} = r | \hat{C}^{(t)} = c, \hat{C}^{(t-1)} = c', \hat{R}^{(t-1)} = 0\}}_{(c)} \times \underbrace{\Pr\{\hat{C}^{(t)} = c | \hat{C}^{(t-1)} = c', \hat{R}^{(t-1)} = 0\}}_{(d)} \mathbf{\Gamma}_n^{(t-1)}[c', 0].$$

Since the inactive symbol in the decoding step from time $t-1$ to t can be regarded as the only decodable input symbol in time $t-1$, we can obtain (c) and (d) using Lemma D.3 with $r'=1$ and Lemma D.2 with $r'=1$, respectively. Thus, we have

$$\mathbf{\Gamma}_n^{t(2)}[c,:] = \sum_{c' \geq c} \mathrm{Bi}(c;c',1-\rho_t)\mathbf{\Gamma}_n^{(t-1)}[c',0]\mathbf{e}_0\mathbf{Q}_t^{c'-c}. \tag{E.2}$$

Combining (E.1) and (E.2), the recursive formula of Theorem 8.1 is proved. □

Proof of Theorem 8.2. We first show by induction that for $1 \leq c \leq n$ and $t \geq 0$,

$$\mathbf{\Gamma}_n^{(t)}[c,:] = \frac{n}{c}\prod_{i=0}^{t}(1-\rho_i)\mathbf{\Gamma}_{n-1}^{(t)}[c-1,:]. \tag{E.3}$$

Since $\mathbf{\Gamma}_n^{(0)} = \mathbf{\Lambda}_n^{(0)}$, we have by (D.9) that (E.3) holds with Suppose that (E.3) holds for certain $t \geq 0$. Applying the recursive formula of Theorem 8.1, we can show that

$$\begin{aligned}
\mathbf{\Gamma}_{n+1}^{(t)}[c,:] &= \sum_{c'=c}^{n+1} \mathrm{Bi}(c;c',1-\rho_t)\mathbf{\Gamma}_{n+1}^{(t-1)}[c',:]\mathbf{N}_t\mathbf{Q}_t^{c'-c} \\
&= \sum_{c'=c}^{n+1} \mathrm{Bi}(c;c',1-\rho_t)\frac{n+1}{c'}\prod_{i=0}^{t-1}(1-\rho_i)\mathbf{\Gamma}_n^{(t-1)}[c',:]\mathbf{N}_t\mathbf{Q}_t^{c'-c} \\
&= \frac{n+1}{c}\prod_{i=0}^{t}(1-\rho_i)\sum_{c'=c}^{n+1}\mathrm{Bi}(c-1;c'-1,1-\rho_t)\mathbf{\Gamma}_n^{(t-1)}[c'-1,:]\mathbf{N}_t\mathbf{Q}_t^{c'-c} \\
&= \frac{n+1}{c}\prod_{i=0}^{t}(1-\rho_i)\sum_{c''=c-1}^{n}\mathrm{Bi}(c-1;c'',1-\rho_t)\mathbf{\Gamma}_n^{(t-1)}[c'',:]\mathbf{N}_t\mathbf{Q}_t^{c''-(c-1)} \\
&= \frac{n+1}{c}\prod_{i=0}^{t}(1-\rho_i)\mathbf{\Gamma}_n^{(t)}[c-1,:].
\end{aligned}$$

By expanding (E.3) recursively, we have for $c \geq 0$ and $t \geq 0$,

$$\mathbf{\Gamma}_n^{(t)}[c,:] = \binom{n}{c}\prod_{i=0}^{t}(1-\rho_i)^c\mathbf{\Gamma}_{n-c}^{(t)}[0,:]. \tag{E.4}$$

Substituting (E.4) into (8.2) and by Lemma 7.4, we get

$$P_{\mathrm{inac}}(t|n) = \sum_{c=0}^{n}\binom{n}{c}\prod_{i=0}^{t}(1-\rho_i)^c\mathbf{\Gamma}_{n-c}^{(t)}[0,0] = \sum_{c=0}^{n}\binom{n}{c}\left(1-\sum_{\tau=0}^{t}p_\tau\right)^c\mathbf{\Gamma}_{n-c}^{(t)}[0,0],$$

proving the formula of $P_{\text{inac}}(t|n)$. Further, (8.6) is obtained by (8.3) for $c = 0$. To prove (8.7), we have

$$\begin{aligned}\mathbf{\Gamma}_n^{(t)}[0,:] &= \sum_{c=0}^{n} \rho_t^c \mathbf{\Gamma}_n^{(t-1)}[c,:]\mathbf{N}_t\mathbf{Q}_t^c \\ &= \sum_{c=0}^{n} \binom{n}{c} \rho_t^c \prod_{i=0}^{t-1}(1-\rho_i)^c \mathbf{\Gamma}_{n-c}^{(t-1)}[0,:]\mathbf{N}_t\mathbf{Q}_t^c \\ &= \sum_{c=0}^{n} \binom{n}{c} p_t^c \mathbf{\Gamma}_{n-c}^{(t-1)}[0,:]\mathbf{N}_t\mathbf{Q}_t^c,\end{aligned}$$

where the first equality follows from (8.4) with $c = 0$, the second equality is obtained by substituting (E.4), and the last step is obtained by applying Lemma 7.4. □

Proof of Theorem 8.3. We first show

$$\mathbf{\Gamma}_n^{(t)}[0,:] = \sum_{i=0}^{2^t-1} \mathbf{V}'_{t,i} \mathbf{\Delta}_{t,i}^n \mathbf{U}_t^{-1} \tag{E.5}$$

by induction in t. The claim for $t = 0$ can be shown by replacing $p_0\mathbf{Q}_0$ in (8.6) with the decomposition in Lemma 7.10. Suppose that the claim of the theorem holds for certain $t \geq 0$. Substituting this form of $\mathbf{\Lambda}_n^t$ into (8.7) with $t+1$ in place of t, we obtain

$$\mathbf{\Gamma}_n^{(t+1)}[0,:] = \sum_{i=0}^{2^t-1} \mathbf{V}'_{t,i} \sum_{c=0}^{n} \binom{n}{c} \mathbf{\Delta}_{t,i}^{n-c} \mathbf{U}_t^{-1} \mathbf{N}_{t+1}\mathbf{U}_{t+1}(p_{t+1}\mathbf{D}_{t+1})^c \mathbf{U}_{t+1}^{-1}. \tag{E.6}$$

We can verify that

$$\begin{aligned}\mathbf{U}_t^{-1}\mathbf{N}_{t+1}\mathbf{U}_{t+1} &= (\mathbf{U}_t^{-1}[:,0]\mathbf{e}_0 + \mathbf{U}_t^{-1}[:,1:])\mathbf{U}_{t+1} \\ &= \begin{bmatrix}\mathbf{U}_{t+1}[0,:] - \mathbf{U}_t[0,1:] \\ \mathbf{I}\end{bmatrix} = \begin{bmatrix}\mathbf{U}_{t+1}[0,:] - \mathbf{U}_t[0,1:] \\ \mathbf{0}\end{bmatrix} + \begin{bmatrix}\mathbf{0} \\ \mathbf{I}\end{bmatrix},\end{aligned}$$

where the second equality follows from (D.11). Similar to the steps obtaining (D.13), substituting the above equation into (E.6) and combining the binomial terms, we get

$$\begin{aligned}\mathbf{\Lambda}_n^{(t+1)} = &\sum_{i=0}^{2^t-1} \mathbf{V}'_{t,i}[1:] \left(\mathbf{\Delta}_{t,i}[1:,1:] + p_{t+1}\mathbf{D}_{t+1}\right)^n \mathbf{U}_{t+1}^{-1} \\ &+ \sum_{i=0}^{2^t-1} (\mathbf{U}_{t+1}[0,:] - \mathbf{U}_t[0,1:])\mathbf{V}'_{t,i}[0] \left(\mathbf{\Delta}_{t,i}[0,0]\mathbf{I} + p_{t+1}\mathbf{D}_{t+1}\right)^n \mathbf{U}_{t+1}^{-1}.\end{aligned}$$

The proof of (E.5) is completed by checking the definition of $\mathbf{V}'_{t+1,i}$ and $\mathbf{\Delta}_{t+1,i}$.

Substituting the above formula of $\mathbf{\Gamma}_n^{(t)}$ in (E.5) into (8.5), we obtain the following formula of $P_{\text{inac}}(t|n)$ given in this theorem. □

Proof of Theorem 8.4. When $t < r_{\mathrm{BP}}$, we know that $\mathbf{\Delta}_{t,i}[0,0] = 0$ (see (D.18)). So

$$P_{\mathrm{inac}}(t|n) = \left(1 - \sum_{\tau=0}^{t} p_\tau\right)^n \sum_{i=0}^{2^t-1} \mathbf{V}'_{t,i}[0] = q_t^n \sum_{i=0}^{2^t-1} \mathbf{V}'_{t,i}[0].$$

It can be shown inductively that

$$\sum_{i=0}^{2^t-1} \mathbf{V}'_{t,i} = \mathbf{U}_t[0,:]. \tag{E.7}$$

First, by definition $\mathbf{V}'_{0,0} = \mathbf{U}_0[0,:]$. Suppose that (E.7) holds for certain $t > 0$. We write

$$\begin{aligned}\sum_{i=0}^{2^t+1} \mathbf{V}'_{t+1,i} &= \sum_{i=0}^{2^t-1} \mathbf{V}'_{t,i}[1:] + \sum_{i=0}^{2^t-1} \mathbf{V}'_{t,i}[0](\mathbf{U}_{t+1}[0,:] - \mathbf{U}_t[0,1:]) \\ &= \mathbf{U}_t[0,1:] + \mathbf{U}_t[0,0](\mathbf{U}_{t+1}[0,:] - \mathbf{U}_t[0,1:]) \\ &= \mathbf{U}_{t+1}[0,:],\end{aligned}$$

where the second equality follows by the induction hypothesis and the last equality follows by $\mathbf{U}_t[0,0] = 1$. By (E.7), we have $P_{\mathrm{inac}}(t|n) = q_t^n \mathbf{U}_t[0,0] = q_t^n$.

When $t \geq r_{\mathrm{BP}}$, by (D.22), we know that for $i = 1, \ldots, 2^t - 1$,

$$\frac{1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0]}{q_t} < 1. \tag{E.8}$$

By Theorem 8.3 and noting that $\mathbf{V}'_{t,0}[0] = \mathbf{U}_0[0,t] = \binom{K}{t} > 0$, we write

$$\begin{aligned}&\lim_{n\to\infty} \frac{-\log P_{\mathrm{inac}}(t|n)}{n} \\ &= \lim_{n\to\infty} \frac{-\log q_t^n \sum_{i=0}^{2^t-1} \mathbf{V}'_{t,i}[0](1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0])^n / q_t^n}{n} \\ &= -\log q_t + \lim_{n\to\infty} \frac{-\log\left(\mathbf{V}'_{t,0}[0] + \sum_{i=1}^{2^t-1} \mathbf{V}'_{t,i}[0](1 - \sum_{\tau=0}^{t} p_\tau + \mathbf{\Delta}_{t,i}[0,0])^n / q_t^n\right)}{n} \\ &= -\log q_t.\end{aligned}$$

The proof is completed. □

Proof of Theorem 8.6. The recursive formula in Theorem 8.1 can be rewritten into a form similar to (D.23) as:

$$\bar{\mathbf{\Gamma}}_n^{(t)}[c,:] = \sum_{c'\geq c} \mathrm{Bi}(c;c',1-\rho_t)\bar{\mathbf{\Gamma}}_n^{(t-1)}[c',:]\bar{\mathbf{N}}_t\bar{\mathbf{Q}}_t^{c'-c},$$

where $\bar{\mathbf{\Gamma}}_n^{(t)}[c,:] = (\mathbf{0} \;\; \mathbf{\Gamma}_n^{(t)}[c,:])$ and $\bar{\mathbf{N}}_t$ is a $(K+1)\times(K+1)$ matrix such $\bar{\mathbf{N}}_t[t-1:,t:] = \mathbf{N}_t$ and all the other components are zeros. The proof can be completed by following the steps after (D.23) and using the fact (5.23). □

Proof of Theorem 8.7. When $t < r_{\mathrm{BP}}$, we know that $\mathbf{\Delta}_{t,i}[0,0] = 0$ (see (D.18)). So

$$\tilde{P}_{\mathrm{inac}}(t|\bar{n}) = \exp\left(-\bar{n}\sum_{\tau=0}^{t} p_\tau\right)\sum_{i=0}^{2^t-1}\mathbf{V}'_{t,i}[0],$$

where $\sum_{i=0}^{2^t-1}\mathbf{V}'_{t,i}[0] = \mathbf{U}_t[0,0] = 1$ by (E.7).

When $t \geq r_{\mathrm{BP}}$, by (E.8) and $\mathbf{V}'_{t,0}[0] = \mathbf{U}_0[0,t] = \binom{K}{t} > 0$, we write

$$\begin{aligned}
&\lim_{\bar{n}\to\infty}\frac{-\log\tilde{P}_{\mathrm{inac}}(t|\bar{n})}{\bar{n}}\\
&=\lim_{\bar{n}\to\infty}\frac{-\log\exp(-\bar{n}(1-q_t))\sum_{i=0}^{2^t-1}\mathbf{V}'_{t,i}[0]\exp\left(-\bar{n}\left(\sum_{\tau=0}^{t}p_\tau-\mathbf{\Delta}_{t,i}[0,0]-1+q_t\right)\right)}{\bar{n}}\\
&=1-q_t.
\end{aligned}$$

The proof is completed. □

Bibliography

[1] A. S. Abdullah, M. J. Abbasi, and N. Fisal. Review of rateless-network-coding-based packet protection in wireless sensor networks. *Mobile Information Systems*, 2015. DOI: 10.1155/2015/641027. 15

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4): 1204–1216, July 2000. DOI: 10.1109/18.850663. 12

[3] A. H. Al-Mohy and N. J. Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM Journal on Scientific Computing*, 33(2): 488–511, 2011. DOI: 10.1137/100788860. 119

[4] G. E. Andrews. *The Theory of Partitions*. *Encyclopedia of Mathematics and its Applications*, vol. 2. Addison-Wesley Pub. Co., 1976. DOI: 10.1017/cbo9780511608650. 27, 71

[5] A. Apavatjrut, C. Goursaud, K. Jaffres-Runser, C. Comaniciu, and J.-M. Gorce. Toward increasing packet diversity for relaying LT fountain codes in wireless sensor networks. *IEEE Communications Letters*, 15(1): 52–54, 2011. DOI: 10.1109/lcomm.2010.111910.101692. 15

[6] A. Apavatjrut, K. Jaffres-Runser, C. Goursaud, and J.-M. Gorce. Combining LT codes and XOR network coding for reliable and energy efficient transmissions in wireless sensor networks. In *Sarnoff Symposium (SARNOFF), 35th IEEE*, pages 1–6, 2012. DOI: 10.1109/sarnof.2012.6222736. 15

[7] D. Blackwell. On an equation of Wald. *The Annals of Mathematical Statistics*, 17(1): 84–87, 1946. DOI: 10.1214/aoms/1177731028. 116

[8] F. L. Blasco, G. Liva, and G. Bauch. LT code design for inactivation decoding. In *Information Theory Workshop (ITW), IEEE*, pages 441–445, November 2014. DOI: 10.1109/ITW.2014.6970870. 121

[9] J. W. Byers, M. Luby, M. Mitzenmacher, and A. Rege. A digital fountain approach to reliable distribution of bulk data. In *Proc. ACM SIGCOMM'98*, pages 56–67, New York, NY, 1998. DOI: http://doi.acm.org/10.1145/285237.285258. 7

[10] S. Chachulski, M. Jennings, S. Katti, and D. Katabi. Trading structure for randomness in wireless opportunistic routing. *SIGCOMM Computer Communication Review*, 37(4): 169–180, August 2007. ISSN 0146-4833. DOI: 10.1145/1282427.1282400. 14, 15

[11] M.-L. Champel, K. Huguenin, A.-M. Kermarrec, and N. Le Scouarnec. LT network codes. In *Proc. IEEE ICDCS'10*, June 2010. DOI: 10.1109/icdcs.2010.14. 15

[12] P. A. Chou, Y. Wu, and K. Jain. Practical network coding. In *Proc. Allerton Conference Communication, Control, and Computing*, October 2003. 12, 15, 16

[13] A. F. Dana, R. Gowaikar, R. Palanki, B. Hassibi, and M. Effros. Capacity of wireless erasure networks. *IEEE Transactions on Information Theory*, 52(3): 789–804, 2006. 12, 153

[14] C. Fragouli, D. Lun, M. Medard, and P. Pakzad. On feedback for network coding. In *Proc. CISS'07*, pages 248–252, 2007. DOI: 10.1109/ciss.2007.4298308. 18

[15] C. Fragouli and E. Soljanin. *Network Coding Fundamentals*. Foundations and Trends in Networking. November 2007. DOI: 10.1561/1300000003. 12

[16] M. Gadouleau and Z. Yan. Packing and covering properties of subspace codes for error control in random linear network coding. *IEEE Transactions on Information Theory*, 56(5): 2097–2108, May 2010. DOI: 10.1109/TIT.2010.2043780. 27, 71

[17] C. Gkantsidis and P. Rodriguez. Network coding for large scale content distribution. In *Proc. IEEE INFOCOM'05*, 2005. DOI: 10.1109/infcom.2005.1498511. 15

[18] R. Gummadi and R. S. Sreenivas. Relaying a fountain code across multiple nodes. In *Proc. IEEE ITW'08*, pages 149–153, May 2008. DOI: 10.1109/ITW.2008.4578640. 9, 15

[19] A. Heidarzadeh and A. H. Banihashemi. Overlapped chunked network coding. In *Proc. ITW'10*, pages 1–5, 2010. DOI: 10.1109/ITWKSPS.2010.5503153. 16, 17

[20] N. J Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Journal on Matrix Analysis and Applications*, 26(4): 1179–1193, 2005. DOI: 10.1137/04061101x. 118, 119

[21] T. Ho, B. Leong, M. Medard, R. Koetter, Y. Chang, and M. Effros. The benefits of coding over routing in a randomized setting. In *Proc. IEEE ISIT'03*, June 2003. DOI: 10.1109/isit.2003.1228459. 12, 13, 17, 23

[22] T. Ho and D. Lun. *Network Coding: An Introduction*. Cambridge, 2008. DOI: 10.1017/cbo9780511754623. 12

[23] K.-C. Huang and Z. Wang. *Millimeter Wave Communication Systems*, vol 29. John Wiley & Sons, 2011. DOI: 10.1002/9780470889886. 2

[24] Q. Huang, K. Sun, X. Li, and D. O. Wu. Just fun: A joint fountain coding and network coding approach to loss-tolerant information spreading. In *Proc. of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc'14*, ACM, pages 83–92, New York, NY, 2014. DOI: 10.1145/2632951.2632965. 23, 156

[25] M. Jafari, L. Keller, C. Fragouli, and K. Argyraki. Compressed network coding vectors. In *Proc. IEEE ISIT'09*, pages 109–113, 2009. DOI: 10.1109/isit.2009.5206041. 18

[26] S. Jaggi, P. A. Chou, and K. Jain. Low complexity optimal algebraic multicast codes. In *Proc. IEEE ISIT'03*, June 2003. 12

[27] S. Jaggi, Y. Cassuto, and M. Effros. Low complexity encoding for network codes. In *Proc. IEEE ISIT'06*, pages 40–44, 2006. DOI: 10.1109/isit.2006.261594. 18

[28] A. Kamra, V. Misra, J. Feldman, and D. Rubenstein. Growth codes: Maximizing sensor network data persistence. In *Proc. of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM'06*, ACM, pages 255–266, New York, NY, 2006. DOI: 10.1145/1159913.1159943. 67

[29] R. Karp, M. Luby, and A. Shokrollahi. Finite length analysis of LT codes. In *Proc. of the International Symposium on Information Theory (ISIT)*, 2004. DOI: 10.1109/ISIT.2004.1365074. 105, 110

[30] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. XORs in the air: Practical wireless network coding. In *Proc. SIGCOMM'06*, New York, NY, 2006. DOI: 10.1145/1159913.1159942. 15

[31] L. Keller, E. Drinea, and C. Fragouli. Online broadcasting with network coding. In *Proc. NetCod'08*, pages 1–6, 2008. DOI: 10.1109/netcod.2008.4476183. 18

[32] R. Koetter and M. Medard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11(5): 782–795, October 2003. DOI: 10.1109/tnet.2003.818197. 12, 17

[33] R. Koetter and F. R. Kschischang. Coding for errors and erasures in random network coding. *IEEE Transactions on Information Theory*, 54(8): 3579–3591, August 2008. DOI: 10.1109/isit.2007.4557321. 18, 31

[34] B. A. LaMacchia and A. M. Odlyzko. Solving large sparse linear systems over finite fields. In *Proc. CRYPTO'90*, pages 109–133, Springer, 1991. DOI: 10.1007/3-540-38424-3_8. 129

[35] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49(2): 371–381, February 2003. DOI: 10.1109/tit.2002.807285. 12, 17

[36] Y. Li, E. Soljanin, and P. Spasojevic. Effects of the generation size and overlap on throughput and complexity in randomized linear network coding. *IEEE Transactions on Information Theory*, 57(2): 1111–1123, February 2011. DOI: 10.1109/TIT.2010.2095111. 16, 17

[37] A. Liau, S. Yousefi, and I. M. Kim. Binary soliton-like rateless coding for the y-network. *IEEE Transactions on Communications*, 59(12): 3217–3222, December 2011. DOI: 10.1109/TCOMM.2011.091911.100189. 15

[38] A. Limmanee and W. Henkel. A cooperative scheme for shaping degree distribution of LT-coded symbols in network coding multicast. In *Source and Channel Coding (SCC), International ITG Conference on*, pages 1–6, IEEE, 2010. 15

[39] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder. RaptorQ forward error correction scheme for object delivery—RFC 6330, 2011. `http://datatracker.ietf.org/doc/rfc6330/`. DOI: 10.17487/rfc6330. 30

[40] M. Luby. LT codes. In *Foundations of Computer Science (FOCS), Proc. the 43rd Annual IEEE Symposium on*, pages 271–280, November 2002. DOI: 10.1109/SFCS.2002.1181950. 8

[41] M. G. Luby, M. Mitzenmacher, and M. A. Shokrollahi. Analysis of random processes via and-or tree evaluation. In *Proc. of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'98*, pages 364–373, Philadelphia, PA, 1998. 77

[42] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. Efficient erasure correcting codes. *IEEE Transactions on Information Theory*, 47(2): 569–584, 2001. DOI: 10.1109/18.910575. 69, 76

[43] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman. Improved low-density parity-check codes using irregular graphs. *IEEE Transactions on Information Theory*, 47(2): 585–598, 2001. DOI: 10.1109/18.910576. 79

[44] D. S. Lun, P. Pakzad, C. Fragouli, M. Medard, and R. Koetter. An analysis of finite-memory random linear coding on packet streams. In *The 2nd Workshop on Network Coding*, pages 1–6, 2006. DOI: 10.1109/wiopt.2006.1666502. 18

[45] D. S. Lun, M. Médard, R. Koetter, and M. Effros. On coding for reliable communication over packet networks. *Physical Communication*, 1(1): 3–20, 2008. DOI: 10.1016/j.phycom.2008.01.006. 12, 153

[46] D. J. C. MacKay. Fountain codes. *IEE Proc. on Communication*, 152(6): 1062–1068, 2005. DOI: 10.1049/ip-com:20050237. 8

[47] K. Mahdaviani, M. Ardakani, H. Bagheri, and C. Tellambura. Gamma codes: A low-overhead linear-complexity network coding solution. In *Network Coding (NetCod), International Symposium on*, pages 125–130, June 2012. DOI: 10.1109/NETCOD.2012.6261896. 17, 18

[48] K. Mahdaviani, R. Yazdani, and M. Ardakani. Overhead-optimized gamma network codes. In *Proc. NetCod'13*, 2013. 17

[49] E. Maneva and A. Shokrollahi. New model for rigorous analysis of LT-codes. In *Proc. IEEE ISIT'06*, pages 2677–2679, July 2006. DOI: 10.1109/isit.2006.262139. 105

[50] P. Maymounkov. Online codes. Technical report, NYU, November 2002. 8, 67, 79, 81

[51] P. Maymounkov, N. J. A. Harvey, and D. S. Lun. Methods for efficient network coding. In *Proc. Allerton Conference on Communication, Control, and Computing*, September 2006. 16

[52] M. Medard and A. Sprintson. *Network Coding: Fundamentals and Applications*. Academic Press, 2011. 12

[53] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1): 3–49, 2003. DOI: 10.1137/s00361445024180. 118

[54] T.-C. Ng and S. Yang. Finite-length analysis of BATS codes. In *Network Coding (NetCod), International Symposium on*, pages 1–6, June 7–9, 2013. DOI: 10.1109/NetCod.2013.6570815. xvi, 105, 110, 130

[55] T. C. Ng and S. Yang. Finite-length analysis of BATS codes, 2013. `http://arxiv.org/abs/1312.4811v1`. DOI: 10.1109/netcod.2013.6570815. 110

[56] U. Niesen, C. Fragouli, and D. Tuninetti. On capacity of line networks. *Information Theory, IEEE Transactions on*, 53(11): 4039–4058, 2007. DOI: 10.1109/tit.2007.907513. 18

[57] J. Nutini, M. W. Schmidt, I. H. Laradji, M. P. Friedlander, and H. A. Koepke. Coordinate descent converges faster with the Gauss-Southwell rule than random selection. In *ICML*, pages 1632–1641, 2015. 122

[58] P. Pakzad, C. Fragouli, and A. Shokrollahi. Coding schemes for line networks. In *Proc. IEEE ISIT'05*, pages 1853–1857, 2005. DOI: 10.1109/ISIT.2005.1523666. 4, 9, 10, 14

[59] C. Pomerance and J. W. Smith. Reduction of huge, sparse matrices over finite fields via created catastrophes. *Experimental Math*, 1(89–94), 1992. DOI: 10.1080/10586458.1992.10504250. 129

[60] S. Puducheri, J. Kliewer, and T. E. Fuja. The design and performance of distributed LT codes. *IEEE Transactions on Information Theory*, 53(10): 3740–3754, October 2007. DOI: 10.1109/TIT.2007.904982. 15

[61] S. Puducheri and T. E. Fuja. Capacity and coding for two common wireless erasure relay networks with optimal bandwidth allocation. *IEEE Transactions on Wireless Communications*, 11(12): 4308–4317, December 2012. DOI: 10.1109/TWC.2012.092512111404. 5

[62] T. Richardsan and R. Urbanke. *Modern Coding Theory*. Cambridge University Press, Cambridge, UK, 2008. DOI: 10.1017/cbo9780511791338. 69, 76, 157

[63] T. J. Richardson and R. L. Urbanke. Efficient encoding of low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2): 638–656, 2001. DOI: 10.1017/cbo9780511791338.010. 129

[64] P. Sanders, S. Egner, and L. Tolhuizen. Polynomial time algorithms for network information flow. In *Proc. ACM SPAA'03*, pages 286–294, New York, NY, 2003. DOI: 10.1145/777412.777464. 12

[65] S. Sanghavi. Intermediate performance of rateless codes. In *Information Theory Workshop, ITW'07. IEEE*, pages 478–482, September 2007. DOI: 10.1109/ITW.2007.4313121. 67

[66] D. Sejdinovic, R. J. Piechocki, and A. Doufexi. And-or tree analysis of distributed LT codes. In *IEEE Information Theory Workshop on Networking and Information Theory*, pages 261–265, June 2009. DOI: 10.1109/ITWNIT.2009.5158583. 15

[67] A. Shokrollahi, S. Lassen, and R. Karp. Systems and processes for decoding chain reaction codes through inactivation, February 2005. 129

[68] A. Shokrollahi. Raptor codes. *Information Theory, IEEE Transactions on*, 52(6): 2551–2567, June 2006. DOI: 10.1109/TIT.2006.874390. 8, 29, 121

[69] A. Shokrollahi and M. Luby. *Raptor Codes*, volume 6 of *Foundations and Trends in Communications and Information Theory*. November 2011. DOI: 10.1109/itwitwn.2007.4318034. 8, 30, 129, 140, 142, 143

[70] B. Shrader and A. Ephremides. A queueing model for random linear coding. In *MILCOM—IEEE Military Communications Conference*, pages 1–7, October 2007. DOI: 10.1109/MILCOM.2007.4454981. 13

[71] M. J. Siavoshani, S. Mohajer, C. Fragouli, and S. Diggavi. On the capacity of noncoherent network coding. *IEEE Transactions on Information Theory*, 57(2): 1046–1066, February 2011. DOI: 10.1109/TIT.2010.2094813. 18

[72] D. Silva, F. R. Kschischang, and R. Koetter. A rank-metric approach to error control in random network coding. *IEEE Transactions on Information Theory*, 54(9): 3951–3967, September 2008. DOI: 10.1109/TIT.2008.928291. 18

[73] D. Silva, W. Zeng, and F. R. Kschischang. Sparse network coding with overlapping classes. In *Proc. NetCod'09*, pages 74–79, 2009. DOI: 10.1109/NETCOD.2009.5191397. 16, 17

[74] D. Silva, F. R. Kschischang, and R. Koetter. Communication over finite-field matrix channels. *IEEE Transactions on Information Theory*, 56(3): 1296–1305, March 2010. DOI: 10.1109/tit.2009.2039167. 31

[75] N. A. Smith and R. W. Tromble. Sampling uniformly from the unit simplex, 2004. `http://www.cs.cmu.edu/~nasmith/papers/smith+tromble.tr04.pdf`. 93

[76] J. K. Sundararajan, D. Shah, and M. Medard. ARQ for network coding. In *Proc. IEEE ISIT'08*, pages 1651–1655, 2008. DOI: 10.1109/isit.2008.4595268. 18

[77] B. Tang and S. Yang. An improved design of overlapped chunked codes. In *Communications Proceedings (ICC), IEEE International Conference on*, pages 1–6, May 23–27, 2016. DOI: 10.1109/ICC.2016.7511109. 17

[78] B. Tang, S. Yang, Y. Yin, B. Ye, and S. Lu. Expander graph based overlapped chunked codes. In *Information Theory Proceedings (ISIT), IEEE International Symposium on*, pages 2451–2455, Cambridge, MA, July 1–6 2012. DOI: 10.1109/ISIT.2012.6283956. 17, 18

[79] B. Tang, S. Yang, Y. Yin, B. Ye, and S. Lu. Expander chunked codes. *EURASIP Journal on Advances in Signal Processing*, (1): 1–13, 2015. DOI: 10.1186/s13634-015-0297-8. 16, 17

[80] B. Tang, S. Yang, B. Ye, S. Guo, and S. Lu. Near-optimal one-sided scheduling for coded segmented network coding. *Computers, IEEE Transactions on*, 65(3): 929–939, March 2016. DOI: 10.1109/TC.2015.2435792. 64, 65

[81] N. Thomos and P. Frossard. Degree distribution optimization in Raptor network coding. In *Proc. IEEE ISIT'11*, August 2011. DOI: 10.1109/isit.2011.6034070. 15

[82] B. N. Vellambi, N. Torabkhani, and F. Fekri. Throughput and latency in finite-buffer line networks. *IEEE Transactions on Information Theory*, 57(6): 3622–3643, June 2011. DOI: 10.1109/TIT.2011.2137070. 18

[83] S. Von Solms and A. S. Helberg. The implementation of LT network coding in resource limited RLNC networks. In *Proc. of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC'11)*, 2011. 15

[84] A. Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2): 117–186, 1945. DOI: 10.1214/aoms/1177731118. 116

[85] M. Wang and B. Li. Lava: A reality check of network coding in peer-to-peer live streaming. In *Proc. IEEE INFOCOM'07*, 2007. DOI: 10.1109/infcom.2007.130. 15

[86] N. C. Wormald. Differential equations for random processes and random graphs. *Annals of Applied Probability*, 5(4): 1217–1235, 1995. DOI: 10.1214/aoap/1177004612. 76, 157, 161

[87] N. C. Wormald. The differential equation method for random graph processes and greedy algorithms. Karonsky and Proemel, Eds., *Lectures on Approximation and Randomized Algorithms PWN*, pages 73–155, Warsaw, 1999. 69, 76, 157, 158, 161

[88] Y. Wu. A trellis connectivity analysis of random linear network coding with buffering. In *Proc. IEEE ISIT'06*, Seattle, July 2006. DOI: 10.1109/isit.2006.261636. 12, 153

[89] X. Xu, M. S. G. Praveen Kumar, Y. L. Guan, and P. H. Joo Chong. Two-phase cooperative broadcasting based on batched network code. *IEEE Transactions on Communications*, 64(2): 706–714, February 2016. DOI: 10.1109/TCOMM.2015.2512584. 156

[90] X. Xu, Y. Zeng, Y. L. Guan, and L. Yuan. Bats code with unequal error protection. In *IEEE International Conference on Communication Systems (ICCS)*, pages 1–6, December 2016. DOI: 10.1109/ICCS.2016.7833563. xvi

[91] X. Xu, Y. L. Guan, Y. Zeng, and C. C. Chui. Quasi-universal bats code. *IEEE Transactions on Vehicular Technology*, 66(4): 3497–3501, April 2017. DOI: 10.1109/TVT.2016.2594051. xvi

[92] X. Xu, Y. L. Guan, Y. Zeng, and C. C. Chui. Spatial-temporal network coding based on bats code. *IEEE Communications Letters*, 21(3): 620–623, March 2017. DOI: 10.1109/LCOMM.2016.2636818. 153

[93] S. Yang. Superposition coding for linear operator channels over finite fields. In *Information Theory Workshop (ITW), IEEE*, pages 502–506, Lausanne, Switzerland, Sepember 3–7, 2012. DOI: 10.1109/ITW.2012.6404724. 18

[94] S. Yang and B. Tang. From LDPC to chunked network codes. In *Information Theory Workshop (ITW), IEEE*, pages 406–410, November 2014. DOI: 10.1109/ITW.2014.6970863. 17, 18

[95] S. Yang and R. W. Yeung. Coding for a network coded fountain. In *Information Theory Proceedings (ISIT), IEEE International Symposium on*, pages 2647–2651, Saint Petersburg, Russia, July 31–August 5, 2011. DOI: 10.1109/ISIT.2011.6034050. xvi, 17, 33

[96] S. Yang and R. W. Yeung. Further results on finite-length analysis of bats codes. In *Global Communications Conference Proceedings (Globecom), IEEE*, December 4–8, 2016. DOI: 10.1109/GLOCOMW.2016.7848914. xvi, 105, 130

[97] S. Yang and R. W. Yeung. Batched sparse codes. *Information Theory, IEEE Transactions on*, 60(9): 5322–5346, Sepember 2014. DOI: 10.1109/TIT.2014.2334315. xvi, 17, 26, 33, 69, 123

[98] S. Yang and Q. Zhou. Tree analysis of BATS codes. *IEEE Communications Letters*, 20(1): 37–40, January 2016. DOI: 10.1109/LCOMM.2015.2499192. xvi, 77

[99] S. Yang, J. Meng, and E.-H. Yang. Coding for linear operator channels over finite fields. In *Information Theory Proceedings (ISIT), IEEE International Symposium on*, pages 2413–2417, Austin, June 13–18, 2010. DOI: 10.1109/ISIT.2010.5513770. 31

[100] S. Yang, S.-W. Ho, J. Meng, and E.-H. Yang. Capacity analysis of linear operator channels over finite fields. *Information Theory, IEEE Transactions on*, 60(8): 4880–4901, August 2014. DOI: 10.1109/TIT.2014.2326976. 18

[101] S. Yang, R. W. Yeung, H. F. Cheung, and H. H. F. Yin. BATS: Network coding in action. In *Communication, Control, and Computing (Allerton), 52nd Annual Allerton Conference on*, pages 1204–1211, October 2014. DOI: 10.1109/ALLERTON.2014.7028592. xvi, 38

[102] S. Yang, T. C. Ng, and R. W. Yeung. Finite-length analysis of BATS codes, 2016. `http://arxiv.org/abs/1312.4811v2`. DOI: 10.1109/netcod.2013.6570815. xvi, 105, 110, 130

[103] R. W. Yeung. *Information Theory and Network Coding*. Springer, 2008. 12, 152

[104] R. W. Yeung. Network coding: A historical perspective. *Proc. of IEEE*, 99(3): 366–371, March 2011. DOI: 10.1109/jproc.2010.2094591. 12

[105] H. H. F. Yin, S. Yang, Q. Zhou, and L. M. L. Yung. Adaptive recoding for BATS codes. In *Information Theory Proceedings (ISIT), IEEE International Symposium on*, pages 2349–2353, July 10–15, 2016. DOI: 10.1109/ISIT.2016.7541719. 64, 65

[106] M. Zelen and N. C. Severo. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. New York, Dover, 1972. DOI: 10.2307/1266136. 165

[107] H. Zhang, K. Sun, Q. Huang, Y. Wen, and D. Wu. Fun coding: Design and analysis. *IEEE/ACM Transactions on Networking*, (99): 1–1, 2016. DOI: 10.1109/TNET.2016.2516819. 23, 156

[108] Y. Zhang. Lecture notes for CAAM 378—a quick introduction to linear programming (draft). http://www.caam.rice.edu/~yzhang/caam378/Notes/lec_notes.pdf. 171

[109] H. Zhao, G. Dong, and H. Li. Simplified bats codes for deep space multihop networks. In *IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pages 311–314, May 2016. DOI: 10.1109/ITNEC.2016.7560372. 156

[110] H. Zhao, S. Yang, and G. Dong. A polynomial formula for finite-length BATS code performance. *IEEE Communications Letters*, 2017. DOI: 10.1109/LCOMM.2016.2619698. 105

[111] H. Zhao, S. Yang, and G. Feng. Fast degree-distribution optimization for bats codes. *Science China Information Sciences*, 60(10): 102301, July 2017. DOI: 10.1007/s11432-016-9011-8. 105

[112] Q. Zhou, H. H. F. Yin, S. Yang, and B. Tang. On BATS codes with variable batch sizes. *IEEE Communication Letters*, 2017. DOI: 10.1109/LCOMM.2017.2713813.

Authors' Biographies

SHENGHAO YANG

Shenghao Yang was born in China on March 19, 1978. He received his B.S. degree from Nankai University in 2001, an M.S. degree from Peking University in 2004, and a Ph.D. degree in Information Engineering from The Chinese University of Hong Kong in 2008.

He was a visiting student at the Department of Informatics, University of Bergen, Norway in Spring 2017. He was a Postdoctoral Fellow in the University of Waterloo from 2008 to 2009 and in the Institute of Network Coding, The Chinese University of Hong Kong from 2010 to 2012. He was with the Tsinghua University from 2012 to 2015 as an Assistant Professor. He is currently a Research Assistant Professor at The Chinese University of Hong Kong, Shenzhen.

His research interests include network coding, information theory, coding theory, network computation, big data processing, and quantum information. He has published more than 40 papers in international journals and conferences. He is a co-inventor of BATS code and has two U.S. patents granted.

RAYMOND W. YEUNG

Raymond W. Yeung was born in Hong Kong on June 3, 1962. He received B.S., M.Eng., and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY, in 1984, 1985, and 1988, respectively.

He was on leave at École Nationale Supérieure des Télécommunications, Paris, France, during Fall 1986. He was a Member of Technical Staff of AT&T Bell Laboratories from 1988 to 1991. Since 1991, he has been with the Department of Information Engineering at The Chinese University of Hong Kong, where he is now Choh-Ming Li Professor of Information Engineering. Since January 2010, he has been serving as Co-Director of the Institute of Network Coding at The Chinese University of Hong Kong. He was a Consultant at the Jet Propulsion Laboratory in Pasadena, CA, on a project to salvage the malfunctioning Galileo Spacecraft, and a Consultant for NEC, USA.

He is the author of the textbooks *A First Course in Information Theory* (Kluwer Academic/Plenum 2002) and its revision *Information Theory and Network Coding* (Springer 2008), which have been adopted by over 100 institutions around the world. In Spring 2014, he gave the first MOOC in the world on information theory that reached over 25,000 students. His research interests include information theory and network coding.

Dr. Yeung was a member of the Board of Governors of the IEEE Information Theory Society from 1999 to 2001. He has served on the committees of a number of information theory symposiums and workshops. He was General Chair of the First and the Fourth Workshop on Network, Coding, and Applications (NetCod 2005 and 2008), a Technical Co-Chair for the 2006 IEEE International Symposium on Information Theory, and a Technical Co-Chair for the 2006 IEEE Information Theory Workshop, Chengdu, China. He currently serves as an Editor-at-Large of *Communications in Information and Systems*, an Editor of *Foundation and Trends in Communications and Information Theory* and of *Foundation and Trends in Networking*, and was an Associate Editor for Shannon Theory of the *IEEE Transactions on Information Theory* from 2003 to 2005.

He was a recipient of the Croucher Foundation Senior Research Fellowship for 2000/2001, the Best Paper Award (Communication Theory) of the 2004 International Conference on Communications, Circuits and System, the 2005 IEEE Information Theory Society Paper Award, the Friedrich Wilhelm Bessel Research Award of the Alexander von Humboldt Foundation in 2007, and the 2016 IEEE Eric E. Sumner Award (for "pioneering contributions to the field of network coding"). In 2015, he was named an Outstanding Overseas Chinese Information Theorist by the China Information Theory Society. He is a Fellow of the IEEE, Hong Kong Academy of Information Sciences, and Hong Kong Institution of Engineers.

Index