UNIVERSITY OF CALIFORNIA

Los Angeles

Achieving Low-Latency Communication with Feedback: from Information Theory to System Design

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Tsung-Yi Chen

2013

Abstract of the Dissertation

Achieving Low-Latency Communication with Feedback: from Information Theory to System Design

by

Tsung-Yi Chen

Doctor of Philosophy in Electrical Engineering

University of California, Los Angeles, 2013

Professor Richard D. Wesel, Chair

Focusing on the analysis and system design for single-user communication with noiseless feedback, this dissertation consists of two parts. The first part explores the analysis of feedback systems using incremental redundancy (IR) with noiseless transmitter confirmation (NTC). For IR-NTC systems based on *finite-length* codewords and decoding attempts only at *certain specified decoding times*, this dissertation studies the asymptotic expansion achieved by random coding, provides rate-compatible sphere-packing (RCSP) performance approximations, and presents simulation results of tail-biting convolutional codes. The RCSP approximations show great agreement with the convolutional code simulations. Both the approximations and the simulations yield expected throughputs significantly higher than random codes at short latencies.

Motivated by the analyses and optimizations in the first part, the second part of this dissertation proposes a new class of rate-compatible low-density parity-check codes, called Protograph-Based Raptor-Like (PBRL) codes. Similar to Raptor codes, PBRL codes can efficiently produce incremental redundancy, providing extensive rate compatibility. The construction and optimization of PBRL codes suitable for both long-blocklength and short-blocklength applications are discussed. Finally, this dissertation provides examples of constructing PBRL codes with different blocklengths. Extensive simulation results of the PBRL code examples and three other standardized channel codes (3GPP-LTE, CCSDS and DVB-S2) are presented and compared.

The dissertation of Tsung-Yi Chen is approved.

Dariush Divsalar

Lara Dolecek

Mario Gerla

Lieven Vandenberghe

Richard D. Wesel, Committee Chair

University of California, Los Angeles

2013

To Wan-Yi

TABLE OF CONTENTS

List of Figures

ACKNOWLEDGMENTS

I cannot imagine how this dissertation would be possible without the support of my family, teachers, colleagues and friends. First and foremost, I want to thank my parents, Kuo-Chiang and Yi-Jie, and my wife Wan-Yi for their unconditional love, patience and support for me to pursue my degree.

In the Electrical Engineering department, I want to first thank my advisor, Professor Richard D. Wesel, who gave me the opportunity to a five-year journey in research. His passion for research inspired and motivated me. His enthusiasm has always been infectious among people around him. I am very fortunate to have the opportunity to work with Professor Dariush Divsalar. One of my research topics is inspired by Professor Dariush Divsalar and I am very grateful for his insightful guidance. I would also like to thank all my other committee members: Professors Lara Dolecek, Lieven Vandenberghe, and Mario Gerla, for their instructions and help in various aspects.

In the Mathematics department, I want to thank Professor Terrence Tao, Oren Louidor, Sebastien Roch and Marek Biskup for their excellent teaching in real analysis and probability theory. The technical skills that I learned from these courses play a crucial role in the development of my dissertation.

I am also very fortunate to have worked with my excellent colleagues in the CSL group: Bike Xie, Thomas Courtade, Jiadong Wang, and Adam Williamson. This journey would have been much harder without their help. Many ideas in this dissertation also rooted from discussions with my colleagues and I thank them for making this dissertation better. For the junior members in Communication System Laboratory, Kasra Vakilinia and Sudarsan Ranganathan, I wish you the best of luck and hope you will enjoy conducting great research in UCLA.

I also thank all of the friends I met in UCLA, who made my life during the past five years more enjoyable: Zaid Towfic, Tien-Kan Chung, Kuo-Wei Huang, Ni-Chun Huang, Jenny Liu, Chung-Yu Lou, Frank Wang, Lawrence Lee, Yifan Sun, Juntao Li, Jason Pan, Li-Te Chang and

Ayca Balcan, among others.

Outside of UCLA, I would like to thank Nambi Seshadri and Ba-Zhong Shen from Broadcom, who have inspired me and contributed a big part of this dissertation. It is my greatest pleasure to have worked with them and to each of them, I am grateful.

I have benefited in many ways from the teachers and colleagues in National Tsing Hua University. I would like to first thank Professors Chi-Chao Chao, who led me into the research of information theory and channel coding. I also want to thank Jen-Ming Wu, Chung-Ching Lu, Yeong-Luh Ueng and Yao-Wen Peter Hong for teaching me the skills I need to conduct great research. I also thank my colleagues in National Tsing Hua University Wei-De Wu, Kuang-Yu Song, Chi-Hsuang Hsieh and Yu-Shuan Teng for many helpful discussions.

The list of people to thank could go on forever, and surely I have forgotten to include people who deserve my dedication. Please forgive the oversight.

VITA

2007	B.S. (Electrical Engineering), National Tsing Hua University, Hsinchu, Taiwan
2009	M.Sc. (Electrical Engineering), UCLA
2008–2013	Research Assistant, Electrical Engineering Department, UCLA
2010–2011	Teaching Assistant, Electrical Engineering Department, UCLA
2009	Intern Engineer, Broadcom Inc.

SELECTED PUBLICATIONS

T.-Y. Chen, A. D. Williamson, N. Seshadri, and R. D. Wesel. "Feedback Communication Systems with Limitations on Incremental Redundancy." In *IEEE Transaction on Information Theory*, Submitted, Sep. 2013.

J. Wang, K. Vakilinia, *T.-Y. Chen*, T. Courtade, G. Dong, T. Zhang, H. Shankar, and R. D. Wesel. "Enhanced Precision Through Multiple Reads for LDPC Decoding in Flash Memories." In *IEEE Journal on Selected Areas in Communications*, Accepted, Sep. 2013.

T.-Y. Chen, A. D. Williamson, and R. D. Wesel. "Variable-length coding with feedback: Finite-length codewords and periodic decoding." In *Proc. IEEE International Symposium of Information Theory*,Istanbul, Turkey, Jul. 2013.

A. D. Williamson, *T.-Y. Chen*, and R. D. Wesel. “Reliability-based error detection for feedback communication with low latency.” In *IEEE International Symposium of Information Theory*, Istanbul, Turkey, Jul. 2013.

A. D. Williamson, *T.-Y. Chen*, and R. D. Wesel. “Firing the genie: Practical two-phase feedback to approach capacity in the short-blocklength regime.” In *Proc. IEEE Information Theory and Its Applications Workshop*, San Diego, California, Feb. 2013.

T.-Y. Chen, D. Divsalar, and R. D. Wesel. “Chernoff bounds for rate-compatible sphere-packing analysis..” In *Proc. IEEE Information Theory Workshop*, Lausanne, Switzerland, Sep. 2012.

A. R. Williamson, *T.-Y. Chen*, and R. D. Wesel. “A rate-compatible sphere-packing analysis of feedback coding with limited retransmissions”. In *Proc. IEEE International Symposium of Information Theory*, Boston, MA, Jul. 2012.

T.-Y. Chen, D. Divsalar, and R. D. Wesel. “Protograph-based raptor-like LDPC codes with low thresholds.” In *Proc. IEEE International Conference of Communications*, Ottawa, Canada, Jun. 2012.

T.-Y. Chen, D. Divsalar, J. Wang, and R. D. Wesel. “Protograph-Based Raptor-Like LDPC codes for rate-compatibility with short blocklengths.” In *Proc. IEEE Global Communications Conference*, Houston, TX, Dec. 2011.

T.-Y. Chen, N. Seshadri, and R. D. Wesel. 11Incremental redundancy: a comparison of a sphere-packing analysis and convolutional codes.” In *Proc. IEEE Information Theory and Applications Workshop*, San Diego, CA, Feb. 2011.

T.-Y. Chen, N. Seshadri, and B-Z. Shen. “Is feedback a performance equalizer of classic and modern codes?” In *Proc. IEEE Information Theory and Applications Workshop*, San Diego, CA, Feb. 2010.

CHAPTER 1

Introduction

Feedback is ubiquitous in ensuring the reliability of delivering data between two remote terminals, connected via wired or wireless media. Many elegant theories have been proposed to enhance the performance of communication using feedback. In the information theory literature, Burnashev, Elias, Schalkwijk and Kailath have shown in their seminal works that feedback can help increase the reliability even in a class of channels where capacity remains unchanged with feedback. The main goal of this dissertation is to study the implications of these elegant results in practical systems, finding the connections between the feedback theory and system design.

This chapter presents a summary of related previous work on noiseless feedback in single-user communication from both the perspective of information theory and that of system design. A brief review of the design of rate-compatible channel codes is also included. After discussing previous work, we summarize the main contributions of this dissertation.

1.1 Literature Review

1.1.1 Feedback in Information Theory

The first notable appearances of feedback in information theory are in 1956. Shannon showed the surprising result that even in the presence of full feedback (i.e., instantaneous feedback of the received symbol) the capacity for a single-user memoryless channel remains the same as without feedback [Sha56]. That same year Elias and Chang each showed that while feedback does not change the capacity, it can make approaching the capacity much easier [Eli56, Cha56]. Elias showed that feedback makes it possible to achieve the best possible distortion using a simple strategy without any coding when transmitting a Gaussian source over a Gaussian channel if the channel bandwidth is an integer multiple of the source bandwidth. Chang showed that feedback can be used to reduce the equivocation $H(X|Y)$ in a coded system to bring its rate closer to capacity for finite blocklengths.

The next phase in the information-theoretic analysis of feedback showed that it can greatly improve the error exponent. Numerous papers including [SK66, Sch66, Kra69, Zig70, NG08, Gal10] explicitly showed this improvement. A key result in this area is the seminal work of Burnashev [Bur76] which provides an elegant expression of the optimal error-exponent for DMC with noiseless feedback. Burnashev employed a technique that can be considered as a form of active hypothesis testing [NJ12], which uses feedback to adapt future transmitted symbols based on the current state of the receiver.

Although active hypothesis testing provides a performance benefit, practical systems using feedback have thus far primarily used feedback to explicitly decide when to stop transmitting additional information about the intended message. This can happen in two ways: Receiver confirmation (RC) occurs when the receiver decides whether it has decoded with sufficient reliability (e.g. passing a checksum) to terminate communication and feeds this decision back to the transmitter. The alternative to RC is transmitter confirmation (TC), in which the transmitter decides (based on feedback from the receiver) whether the receiver has decoded with sufficient reliability (or even if it has decoded correctly, since the transmitter knows the true message).

TC schemes often use distinct transmissions for a message phase and a confirmation phase. Practical TC and RC systems can usually be assigned to one of two categories based on when confirmation is possible. Single-codeword repetition (SCR) only allows confirmation at the end of a complete codeword and repeats the same codeword until confirmation. In contrast, incremental redundancy (IR) systems [Man74] transmit a sequence of distinct coded symbols with numerous opportunities for confirmation within the sequence before the codeword is repeated. In some cases of IR, the sequence of distinct coded symbols is infinite and is therefore never repeated. If the sequence of symbols is finite and thus repeated, we call this a repeated IR system.

Forney's analysis [For68] provided an early connection between these practical system designs and theoretical analysis by deriving error-exponent bounds for a DMC using an SCR-RC scheme. Following Forney's work, Yamamoto and Itoh [YI79] replaced Forney's SCR-RC scheme with a SCR-TC scheme in which the receiver feeds back its decoding result (based only on the codeword sent during the current message phase). The transmitter confirms or rejects the decoded message during a confirmation phase, continuing with additional message and confirmation phases if needed. This relatively simple SCR-TC scheme allows block codes to achieve the optimal error-exponent of Burnashev for DMCs.

Error-exponent results are asymptotic and do not always provide the correct guidance in the low-latency regime. Polyanskiy et al. [PPV11] analyzed the benefit of feedback in the non-asymptotic regime. They studied random-coding IR schemes under both RC and TC, and showed that capacity can be closely approached in hundreds of symbols using feedback [PPV11] rather than the thousands of symbols required without feedback [PPV10].

In [PPV11] the IR schemes provide information to the receiver one symbol at a time. For the RC approach, the receiver provides confirmation when the belief for a prospective codeword exceeds a threshold. For the TC approach, the transmitter provides confirmation using a special noiseless transmitter confirmation (NTC) allowed once per message when the transmitter observes that the currently decoded message is correct. This setup, referred to as variable-length feedback codes with termination (VLFT) in [PPV11], supports zero-error communication. VLFT codes are defined broadly enough to include active hypothesis testing in [PPV11], but the achievability results of interest use feedback only for confirmation.

When expected latency (average blocklength) is constrained to be short, there is a considerable performance gap shown in [PPV11] between the SCR-TC scheme of Yamamoto and Itoh [YI79] and the superior IR-TC scheme of [PPV11]. This is notable because the SCR-TC scheme of Yamamoto and Itoh achieves the best possible error-exponent, demonstrating that error-exponent results do not always provide the correct guidance in the non-asymptotic regime.

Using [PPV11] as our theoretical starting point, this dissertation explores the information-theoretic analysis of IR-NTC systems at low latencies (i.e., with short average blocklengths) under practical constraints and with practical codes.

1.1.2 Feedback in Practical Systems

The use of feedback in practical communication systems predates Shannon's 1948 paper. By 1940 a patent had been filed for an RC feedback communication system using "repeat request" in printing telegraph systems [Van43]. The first analysis of a retransmission system using feedback in the practical literature appears to be [BF64] in 1964, which analyzed automatic repeat request (ARQ) for uncoded systems that employ error detection (ED) to determine when to request a repeat transmission.

In the 1960s, forward-error correction (FEC) and ED-based ARQ were considered as two separate approaches to enhance transmission reliability. Davida proposed the idea of combining ARQ and FEC [Dav72] using punctured, linear, block codes. The possible combinations of FEC and ARQ came to be known as type-I and type-II hybrid ARQ (HARQ), which first appeared in [LY82] (also see [LC04]). Type-I HARQ is an SCR-RC feedback system that repeats the same set of coded symbols with both ED and FEC until the decoded message passes the ED test. Recent literature and also in this dissertation refer to type-I HARQ as simply ARQ because today's systems rarely send uncoded messages. Type-II HARQ originally referred to systems that alternated between uncoded data with error detection and a separate transmission of parity symbols. Today, type-II HARQ has taken on a wider meaning including essentially all IR-RC feedback systems.

Utilizing the soft information of the received coded symbols, Chase proposed a decoding scheme for SCR-RC [Cha85] that applies maximal ratio combining to the repeated blocks of coded symbols. Hagenauer [Hag88] introduced rate-compatible punctured convolutional (RCPC) codes, which allow a wide range of rate flexibility for IR schemes. Rowitch et al. [RM00] and Liu et al.[LSS03] used rate-compatible punctured turbo (RCPT) codes in an IR-RC system to match the expected throughput to the binary-input AWGN channel capacity.

When implementing an IR system using a family of rate-compatible codes, the IR transmissions will often repeat once all of the symbols corresponding to the longest codeword have been transmitted and the confirmation is not yet received. Chen et al. demonstrated in [CSS10] that a repeated IR system using RCPC codes could deliver bit error rate performance similar to long-blocklength turbo and LDPC codes, but with much lower latency. The demonstration of [CSS10] qualitatively agrees with the error-exponent analysis and the random-coding analysis in [PPV11]. Using [CSS10] as our practical starting point, this dissertation explores the practical design of IR-NTC systems at low latencies (i.e., with short average blocklengths) under practical constraints and with practical codes.

Practical design for an IR scheme is primarily concerned with optimizing the incremental transmissions, which includes design of a rate-compatible code family and optimization of the lengths of the incremental transmissions. Practical constraints include limitations on the blocklength N of the lowest-rate codeword in the rate-compatible code family, the number m of incremental transmissions, and the lengths of the incremental transmissions $I_j, j = 1, \dots, m$.

Several papers have provided inspiration in practical deign of feedback systems. Uhlemann et al. [URG03] studied a similar optimization problem assuming that the error probabilities of each transmission are given. Finding code-independent estimates of appropriate target values for these error probabilities is one goal of this dissertation.

In related work, [VST05] provides expressions relating throughput and latency to facilitate optimization of the lengths of incremental transmissions in real-time for an IR-RC system. In contrast with this dissertation, the receivers in [VST05] must send the optimized transmission lengths in addition to the confirmation message in order to adapt the transmission lengths in real time. Moreover, [VST05] used relatively large transmission lengths in order to accurately approximate mappings from codeword reliability to block error rate, while our focus is on short blocklengths.

Chande et al. [CF98] and Visotsky et al. [VTH03] discussed how to choose the optimal blocklengths for incremental redundancy using a dynamic programming framework. Their approach uses the error probabilities of a given code with specified puncturing to determine the optimal transmission lengths for that code. Their work focused on fading channels and on longer blocklengths than this dissertation.

Freudenberger's work [FS04] differs from the present work in that the receiver requests specific incremental retransmissions by determining which segments of the received word are unreliable. This represents a step in the direction of a practical active hypothesis testing system.

In [FH09] the authors used a reliability-based retransmission criteria in a HARQ setting to show the throughput gains compared to using cyclic redundancy checks for error detection. While [FH09] shows simulation results for several values of the message size k, the results are not discussed in terms of proximity to capacity at short blocklengths.

1.1.3 Design of Rate-Compatible Channel Codes

The incremental redundancy systems with receiver confirmation, i.e. IR-RC, are widely used among modern communication systems, e.g. in 3GPP-LTE standard. A necessary element to achieve high-throughput using IR-RC systems is to use a family of rate-compatible channel codes that provides better error protection as the number of received symbols by the receiver increases. Most modern systems rely on upper layer protocol to provide error detection such as cyclic redundancy check, which is often analyzed separately from the physical layer.

As summarized in the previous chapter, RCPC and RCPT codes are among the most popular rate-compatible channel codes used in IR-RC systems. A collection of rate-compatible puncturing patterns must be carefully designed to ensure that a longer code (hence a lower code rate) gives a better error protection than a shorter one. This incremental strength of error protection is not always the case when puncturing a convolutional code as pointed out by Hagenaur [Hag88]. A rather complicated numerical optimization was performed to ensure this property even when the search is restricted to puncturing patterns that are periodic. Similar to RCPC codes, rate-compatible puncturing patterns for RCPT codes must be designed carefully to prevent undue performance loss.

As shown in [LSS03], a family of randomly punctured turbo codes can yield capacity-approaching performance. A similar approach using pseudo-random puncturing of tail-biting convolutional codes and terminated turbo codes is also adopted in the 3GPP-LTE standard [Gen08]. Random puncturing can often work well when the blocklength is large enough. In the short-blocklength regime, however, random puncturing might not necessarily yield incremental strength of error protection as blocklength increases.

Another ideal candidate to construct a family of rate-compatible codes is Low-Density Parity-Check (LDPC) codes, which were first introduced by Gallager in his dissertation in 1963 [Gal63]. Gallager defines an (n, d_v, d_c) LDPC code as a length-n binary code with a parity-check matrix containing d_v ones in each column and d_c ones in each row. This class of LDPC codes is now referred to as regular LDPC codes. Gallager also proposed both soft-decision and hard-decision iterative decoders that were based on message passing in his dissertation, and he showed simulation results for codes with blocklength around 500 bits using hard-decision decoding. He concluded that the simulation results show great potential of LDPC codes for error correction, yet his work received little attention until decades later.

Tanner [Tan81] proposed the construction of a class of long codes with shorter codes and a bipartite graph. He also generalized the decoding algorithm proposed by Gallager. Tanner's work formed the foundation of representing LDPC codes using bipartite graphs. MacKay et al. [Mac99] later showed by simulation that using message-passing algorithm with soft information, LDPC codes also have capacity-approaching performance similar to turbo codes [BGT93]. They also proposed several heuristics to construct good LDPC codes.

In contrast to regular LDPC codes, irregular LDPC codes have parity-check matrices consist of a range of column weights and row weights. The portion of these weights can be described by the so called "degree distributions". Luby et al. [LMS01] showed that properly constructed irregular LDPC codes can achieve rates even closer to capacity than the regular ones. Richardson, Shokrollahi and Urbanke [RSU01] created a systematic method called "density evolution" to design and analyze the optimal degree distribution of LDPC codes based on the assumption that the blocklength can be infinitely long.

Aiming to achieve high throughput in HARQ systems in various classes of channels, numerous heuristics have been proposed to construct rate-compatible LDPC codes. The first work in the construction of rate-compatible LDPC codes appears to be [LN02]. See also [YB04, JS07, EHB09] and the references therein. A straightforward method of constructing rate-compatible LDPC codes is by rate-compatible puncturing of a good mother code at a relatively low rate to obtain the higher-rate codes, which is a similar approach as in [Hag88]. Ha et al. studied the asymptotic behavior of rate-compatible LDPC codes based on density evolution [HKM04]. They also studied the design of puncturing patterns for LDPC codes with relatively short blocklengths [HKK06]. Many heuristics have been proposed on designing better puncturing pattern to enhance the error-rate performance and/or to allow efficient encoding. See for example, [SHS08, KRM09, VF09].

Obtaining a family of rate-compatible LDPC codes is convenient and simple. However, it is generally observed that finite-length LDPC codes suffer from a larger performance degradation compared to turbo codes at high rates [YB04]. Another way to construct rate-compatible codes is by the method of extending codes. Yazdani et al. studied the construction of RC-LDPC codes based on a combination of extending and puncturing [YB04] and concluded that a combination of the two methods yields better rate-compatible codes than just using puncturing alone. Imposing several structural constraints on the LDPC codes, this dissertation uses the idea of extending a high-rate code to obtain a family of rate-compatible codes.

The analysis of HARQ systems using rate-compatible LDPC codes has been studied by Sesia et al. [SCV04] based on random coding and density evolution of infinite-length LDPC ensemble. The focus in [SCV04] is on wireless fading channels whereas this dissertation focuses on memoryless channels and studies both short-blocklength and long-blocklength regimes.

In recent years, a new class of LDPC codes called protograph LDPC codes, or protograph codes, was introduced by Thorpe [Tho03] and studied extensively in [DDJ09]. The design of protograph codes starts out by constructing a relatively small bipartite graph called the protograph. After properly designing the protograph based on density evolution, the graph is copied many times and the edges are permuted carefully to obtain a bipartite graph with a desired blocklength. As indicated in [DDJ09], this type of construction allows efficient decoder implementation in hardware with a similar reasons as in the class of quasi-cyclic LDPC codes [LC04].

The design of rate-compatible LDPC codes based on protograph first appeared in [Dol05]. Nguyen et al. further optimized the design of rate-compatible protograph by density evolution and a greedy search of all possible rate-compatible protographs [NND12].

Using density evolution, this dissertation also studies the construction of rate-compatible protograph codes. In contrast to [Dol05, NND12], we focus on the design of protograph that has similar structure as a class of rateless codes called Raptor codes [Sho06]. Constraining our design to the structure of Raptor codes makes the construction and optimization protograph codes manageable while providing extensive rate-compatibility.

1.2 Summary of Contributions

This first part of this dissertation is concerned with IR-NTC feedback systems that use a rate-compatible code family to send m incremental transmissions. The jth incremental transmission has length I_j so that after i incremental transmissions the cumulative blocklength is $n_i = \sum_{j=1}^{i} I_j$ and the decoder uses all n_i symbols to decode. We are also concerned with the blocklength of the lowest-rate codeword in the rate-compatible code family, which is $N = \sum_{j=1}^{m} I_j$.

Among its numerous other results, [PPV11] provides achievability result for an IR-NTC system with zero-error probability in the non-asymptotic regime. The achievability uses random-coding analysis for the case where $N = \infty$ (and thus $m = \infty$) and $I_j = 1, \forall j$.

The main contributions of this part of the dissertation are as follows:

1. This dissertation extends the IR-NTC random-coding result of [PPV11] to include the constraint of finite N and of $I > 1$ where $I_j = I \ \forall j$. Though N is finite, our setup still

supports zero-error probability through an ARQ-style retransmission if decoding fails even after the N symbols of the lowest-rate codeword have been transmitted. We refer this type of system as repeated IR-NTC. Our analysis yields two primary conclusions:

- Constraining the difference between N and the expected latency to grow logarithmically with the expected latency negligibly reduces expected throughput as compared with $N = \infty$.
- An uniform incremental transmission length I causes expected latency to increase linearly with I as compared to the $I = 1$ case.

2. This dissertation uses rate-compatible sphere-packing (RCSP) approximation as a tool for analysis and optimization of IR feedback systems as follows:

   - Exact joint error probability computations based on RCSP and bounded-distance (BD) decoding are used to optimize the incremental lengths $I_1, \dots I_m$ for small values of m in a repeated IR-NTC system.
   - Tight bounds on the joint RCSP performance under BD decoding are used to optimize a fixed incremental length I ($I_j = I \; \forall j$) for repeated IR-NTC with larger values of m.
   - For non-repeating IR-NTC, exact joint RCSP computations under BD decoding are used to optimize the incremental lengths $I_1, \dots I_m$ to minimize expected latency under an outage constraint.
   - RCSP approximations yield decoding error trajectory curves that provide design targets for families of rate-compatible codes for IR-NTC feedback systems.

3. This dissertation provides simulations of IR-NTC systems based on randomly-punctured tail-biting convolutional codes that demonstrate the following:

   - For $I = 1$ and large N these simulations exceed the random-coding lower bound of [PPV11] on both the BSC and the AWGN channel. For the BSC, the simulated performance closely matches the RCSP approximation at low latencies. For the AWGN, the simulated performance is also similar, but with a gap possibly due to the constraint of the convolutional codes to a binary-input alphabet.

- Simulations were also performed on the AWGN channel using $m = 5$ incremental transmissions with transmission increments I_j optimized using RCSP under the assumption of BD decoding. These simulations exceeded the random-coding lower bound of [PPV11] at low latencies for the same values of m and I_j. Also at sufficiently low latencies, the simulations closely match the throughput-latency performance predicted by RCSP with ML decoding.

Motivated by the first part of the dissertation, the second part of this dissertation focuses on constructing a family of rate-compatible codes. In specific, we propose a new class of rate-compatible LDPC codes called Protograph-Based Raptor-Like LDPC codes, or PBRL codes, that has excellent performance across a wide range of blocklengths and SNRs. We provide the construction of PBRL codes, optimization of PBRL codes, and examples of constructing PBRL codes with long blocklengths and short blocklengths. Extensive simulations of PBRL codes are presented in this part and the error rates of PBRL codes are compared to other modern channel codes in three communication standards: the RCPT codes in the 3GPP-LTE standard [Gen08], the LDPC codes in the DVB-S2 standard [DVB09] and the LDPC codes in the CCSDS standard [CCS07].

CHAPTER 2

Practical Limitations and Optimizations for IR-NTC

2.1 Overview

2.1.1 Contributions

In [PPV11] Polyanskiy et al. studied the benefit of feedback using incremental redundancy (IR) with noiseless transmitter confirmation (NTC) in the non-asymptotic regime providing achievability results based on an infinite-length random code when decoding is attempted at every symbol. This chapter explores IR-NTC based on *finite-length* codes (with blocklength N) and decoding attempts only at *certain specified decoding times*. For IR-NTC with these constraints, this chapter presents the asymptotic expansion achieved by random-coding, provides a rate-compatible sphere-packing (RCSP) performance approximation, and presents simulation results of tail-biting convolutional codes.

The information-theoretic analysis shows that values of N relatively close to the expected latency yield the same random-coding achievability expansion as with $N = \infty$. However, the penalty introduced in the expansion by limiting decoding times is linear in the interval between decoding times. For binary symmetric channels, the RCSP approximation provides an efficiently-computed approximation of performance that shows excellent agreement with a family of rate-compatible, tail-biting convolutional codes in the short-latency regime. For the additive white Gaussian noise channel, bounded-distance decoding simplifies the computation of the marginal RCSP approximation and produces similar results as analysis based on maximum-likelihood decoding for latencies greater than 200. The efficiency of the marginal RCSP approximation facilitates optimization of the lengths of incremental transmissions when the number of incremental transmissions is constrained to be small or the length of the incremental transmissions is constrained to be uniform after the first transmission. Finally, an RCSP-based decoding error trajectory is introduced that provides target error rates for the design of rate-compatible code families for use in feedback communication systems.

2.1.2 Organization

We briefly summarize the organization of this chapter. Sec. 2.2 investigates the rate penalty incurred by imposing the constraints of finite blocklength and limited decoding times on VLFT codes. Sec. 2.2.3 investigates the penalty incurred by finite blocklengths, Sec. 2.2.4 studies the penalty incurred by limiting decoding attempts, and Sec. 2.2.5 studies the penalty when both limitations are applied. Finally a numerical example for the binary symmetric channel (BSC) is presented in Sec. 2.2.6.

Sec. 2.3 describes the examples (used throughout the rest of the chapter) of IR-NTC systems using families of practical rate-compatible codes based on convolutional and turbo codes. Sec. 2.4 presents the RCSP approximations and applies the approximations to provide numerical examples for the BSC and the additive white Gaussian noise (AWGN) channel. Sec. 2.4 also explores ARQ, Chase code combining, and IR-NTC systems for the AWGN channels using RCSP approximation. Sec. 2.5 continues the RCSP-based exploration of the use of IR-NTC on the AWGN channels studying the optimization of the increments $\{I_j\}_{j=1}^{m}$ based on RCSP approximation. Finally Sec. 2.6 concludes this chapter.

2.2 Practical Constraints on VLFT

This section studies IR-NTC systems using the VLFT framework in [PPV11] with the practical constraints of finite N and uniform increments $I > 1$.

2.2.1 Review of VLFT Achievability

We will consider discrete memoryless channels (DMC) in this section and use the following notation: $x^n = (x_1, x_2, \dots, x_n)$ denotes an n-dimensional vector, x_j the jth element of x^n, and x_i^j the ith to jth elements of x^n. We denote a random variable (r.v.) by capitalized letter, e.g., X^n, unless otherwise stated. The input and output alphabets are denoted as $\mathcal{X}$ and $\mathcal{Y}$ respectively. Let the input and output product spaces be $\mathsf{X} = \mathcal{X}^n, \mathsf{Y} = \mathcal{Y}^n$ respectively. A channel used without feedback is characterized by a conditional distribution $P_{\mathsf{Y}|\mathsf{X}} = \prod_{i=1}^n P_{Y_i|X_i}$ where the equality holds because the channel is memoryless. For codes that use a noiseless feedback link, we consider causal channels $\{P_{Y_i|X_1^i Y^{i-1}}\}_{i=1}^\infty$ and additionally focus on causal, memoryless channels $\{P_{Y_i|X_i}\}_{i=1}^\infty$.

Let the finite dimensional distribution of $(X^n, \bar{X}^n, Y^n)$ be:

$$\begin{aligned} & P_{X^n Y^n \bar{X}^n}(x^n, y^n, \bar{x}^n) \\ & = P_{X^n}(x^n) P_{X^n}(\bar{x}^n) \prod_{j=1}^n P_{Y_j|X^j Y^{j-1}}(y_j|x^j, y^{j-1}), \end{aligned} \tag{2.1}$$

i.e., the distribution of $\bar{X}^n$ is identical to X^n but independent of Y^n. The information density $i(x^n; y^n)$ is defined as

$$i(x^n; y^n) = \log \frac{dP_{X^n Y^n}(x^n, y^n)}{d(P_{X^n}(x^n) \times P_{Y^n}(y^n))} \tag{2.2}$$

$$= \log \frac{dP_{Y^n|X^n}(y^n|x^n)}{dP_{Y^n}(y^n)}. \tag{2.3}$$

In this chapter we only consider channels with essentially bounded information density $i(X; Y)$.

This section extends the results in [PPV11] for VLFT codes. In order to be self-contained, we state the definition of VLFT codes in [PPV11]:

Definition 1. *An (ℓ, M, ϵ) variable-length feedback code with termination (VLFT code) is defined as:*

1. *A common r.v. $U \in \mathcal{U}$ with a probability distribution P_U revealed to both transmitter and receiver before the start of transmission.*

2. *A sequence of encoders $f_n : \mathcal{U} \times \mathcal{W} \times \mathcal{Y}^{n-1} \to \mathcal{X}$ that defines the channel inputs $X_n = f_n(U, W, Y^{n-1})$. Here W is the message r.v. uniform in $\mathcal{W} = \{1, \ldots, M\}$.*

3. *A sequence of decoders $g_n : \mathcal{U} \times \mathcal{Y}^n \to \mathcal{W}$ providing the estimate of W at time n.*

4. *A stopping time $\tau \in \mathbb{N}$ w.r.t. the filtration $\mathcal{F}_n = \sigma\{U, Y^n, W\}$ such that:*

$$\mathbb{E}[\tau] \leq \ell. \tag{2.4}$$

5. *The final decision $\hat{W} = g_\tau(U, Y^\tau)$ must satisfy:*

$$\mathbb{P}[\hat{W} \neq W] \leq \epsilon. \tag{2.5}$$

VLFT represents a TC feedback system because the stopping time defined in item 4) above has access to the message W, which is only available at the transmitter. As observed in [PPV11], the setup of VLFT is equivalent to augmenting each channel with a special use-once input symbol, referred to as the termination symbol, that has infinite reliability. We will refer this concept as noiseless transmitter confirmation (NTC) for the rest of the dissertation.

NTC simplifies analysis by separating the confirmation/termination operation from regular physical channel communication. The assumption of NTC captures the fact that many practical systems communicate control signals in upper protocol layers.

The assumption of NTC increases the non-asymptotic, achievable rate to be larger than the original feedback channel capacity because it noiselessly provides the information of the stopping position. This increases the capacity by the conditional entropy of the stopping position given the received symbols normalized by the average blocklength.

The following is the zero-error VLFT achievability in [PPV11]:

Theorem 1 ([PPV11], Thm. 10)**.** *Fixing $M > 0$, there exists an $(\ell, M, 0)$ VLFT code with*

$$\ell \leq \sum_{n=0}^{\infty} \xi_n \tag{2.6}$$

where ξ_n is the following expectation:

$$\xi_n = \mathbb{E} \min\left\{1, (M{-}1)\mathbb{P}[i(X^n; Y^n) \leq i(\bar{X}^n; Y^n) | X^n Y^n]\right\}. \tag{2.7}$$

The expression in (2.7) *is referred to as the random-coding union (RCU) bound. We take the information density before any symbols are received,* $i(X^0;Y^0)$*, to be* 0 *and hence* $\xi_0 = 1$. *Additionally, from the proof of [PPV11, Thm. 11], we have:*

$$\xi_n \leq \mathbb{E}\left[\exp\left\{-[i(X^n;Y^n) - \log(M-1)]^+\right\}\right]. \tag{2.8}$$

The proof of Thm. 1 is based on a special class of VLFT codes called fixed-to-variable (FV) codes [VS10]. FV codes satisfy the following conditions:

$$f_n(U, W, Y^{n-1}) = f_n(U, W) \tag{2.9}$$

$$\tau = \inf\{n \geq 1 : g_n(U, Y^n) = W\}. \tag{2.10}$$

The condition in (2.9) precludes active hypothesis testing since the feedback is not used by the encoder to determine transmitted symbols. The condition in (2.10) enforces zero-error operation since the stopping criterion is correct decoding.

In the proof of Thm. 1 each codeword is randomly drawn according to the capacity-achieving input distribution $\prod_{j=1}^{\infty} P_X$ on the infinite product space X^{∞}. Given a codebook realization, the encoder maps a message to an infinite-length vector and the transmitter sends the vector over the channel symbol-by-symbol. Upon receiving each symbol, the decoder computes M different information densities between the M different codewords and the received vector. The transmitter sends the noiseless confirmation when the largest information density corresponds to the true message. Averaging over all possible codebooks gives the achievability result.

2.2.2 Introducing Practical Constraints to VLFT

Define a VLFT code with the constraints of finite blocklength N and uniform increment I as follows:

Definition 2. *An* $(\ell, M, N, I, \epsilon)$ *VLFT code modifies 2) and 4) in Definition* 1 *as follows:*

2') A sequence of encoders

$$f_{n+kN} : \mathcal{U} \times \mathcal{W} \times \mathcal{Y}_{kN+1}^{n+kN-1} \mapsto \mathcal{X}, k \in \mathbb{N}$$

that satisfies $f_n = f_{n+N}$.

4') A stopping time $\tau \in \{n_1 + kI : k \in \mathbb{N}\}$ w.r.t. the filtration $\mathcal{F}_n$ s.t. $\mathbb{E}[\tau] \leq \ell$ and n_1 is a given constant such that $n_1 + kI | N$ for some $k \in \mathbb{N}$.

Define the fundamental limit of message cardinality M for an $(\ell, M, N, I, \epsilon)$ VLFT code as follows:

Definition 3. *Let $M_t^*(\ell, N, I, \epsilon)$ be the maximum integer M such that there exists an $(\ell, M, N, I, \epsilon)$ VLFT code. For zero-error codes where $\epsilon = 0$, we denote the maximum M as $M_t^*(\ell, N, I)$ and for zero-error codes with $I = 1$ (i.e., decoding attempts after every received symbol) we denote the maximum M as $M_t^*(\ell, N)$.*

For a feedback system that conveys M messages with expected latency ℓ, the expected throughput R_t is given as $R_t = \log M/\ell$. All of the results that follow assume an arbitrary but fixed channel $\{P_{Y_j|X_j}\}_{j=1}^N$ and a channel-input process $\{X_j\}_{j=1}^N$ taking values in $\mathcal{X}$ where N could be infinity.

2.2.3 The Finite-Blocklength Limitation

This subsection investigates $(\ell, M, N, I, \epsilon)$ VLFT codes with finite N but retains decoding at every symbol ($I = 1$). The achievability results are examples of IR-NTC systems (or FV codes as described in Sec. 2.2.1), so that encoding does not depend on the feedback except that feedback indicates when the transmission should be terminated.

In an IR-NTC system, the expected latency $\mathbb{E}[\tau]$ is given as:

$$\mathbb{E}[\tau] = \sum_{n=1}^{\infty} n\mathbb{P}[\tau = n] \tag{2.11}$$

$$= \sum_{n \geq 0} \mathbb{P}[\tau > n] \tag{2.12}$$

$$= \sum_{n \geq 0} \mathbb{P}[E_n] \tag{2.13}$$

$$\leq \sum_{n \geq 0} \mathbb{P}[\zeta_n] \, , \tag{2.14}$$

where

$$E_n = \cap_{j=1}^{n} \zeta_j \tag{2.15}$$

and ζ_j is the marginal error event at the decoder immediately after the jth symbol is transmitted. Equation (2.14) follows since $E_n \subset \zeta_n$ implies $\mathbb{P}[E_n] \leq \mathbb{P}[\zeta_n]$.

Consider a code $\mathcal{C}_N$ with finite blocklength N and symbols from $\mathcal{X}$. Achievability results for an $(\ell, M, N, 1, \epsilon)$ "truncated" VLFT code follow from a random-coding argument. In particular we have the following:

Theorem 2. *For any $M > 0$ there exists an $(\ell, M, N, 1, \epsilon)$ truncated VLFT code with*

$$\ell \leq \sum_{n=0}^{N-1} \xi_n \tag{2.16}$$

$$\epsilon \leq \xi_N. \tag{2.17}$$

where ξ_n is the same as (2.7).

The proof is provided in Appendix 2.A. Achievability results for $\epsilon = 0$ can be obtained using an $(\ell, M, N, 1, 0)$ "repeated" VLFT code, which repeats the transmission if the blocklength-N codeword is exhausted without successful decoding. The transmission process starts from scratch in this case, discarding the previous received symbols. Using the original N symbols through, for example, Chase code combining would be beneficial but is not necessary for our achievability result. Specifically, for an $(\ell, M, N, I, 0)$ repeated VLFT code we have the following result:

Theorem 3. *For every $M > 0$ there exists an $(\ell, M, N, 1, 0)$ repeated VLFT code such that*

$$\ell \leq \frac{1}{(1 - \xi_N)} \sum_{n=0}^{N-1} \xi_n, \tag{2.18}$$

where ξ_n is the same as (2.7).

The proof is provided in Appendix 2.A. The rate penalty of using a finite-length (length-N) codebook is quantified in the following theorem and its corollary:

Theorem 4. *For an $(\ell, M, N, 1, 0)$ repeated VLFT code with $N = \Omega(\log M)$, we have the following upper bound on ℓ for a stationary DMC with capacity C:*

$$\ell \leq \frac{\log M}{C} + c\,. \tag{2.19}$$

Let $C_\Delta = C - \Delta$ for some $\Delta > 0$ and $N = \log M / C_\Delta$. The $O(1)$ term c due to the finite value of N is upper bounded by the following expression:

$$c \leq \frac{b_2 \log M}{C(M^{b_3/C_\Delta})} + \frac{b_0 \log M}{C_\Delta M^{b_1 \Delta / C_\Delta}} + a \tag{2.20}$$

where a depends on the mean and uniform bound of $i(X;Y)$, and b_j's are constants related to Δ and M. The proof is provided in Appendix 2.A.

This choice of N may have residual terms decaying with M very slowly. However, our non-asymptotic numerical results in Sec. 2.2.6 for a BSC indicate that this decay is fast enough for excellent performance in the short-blocklength regime.

An asymptotic expansion of $\log M_t^*(\ell, N)$ needs to be independent of M and requires N growing with ℓ. However, the components of the correction term c in Thm. 4 depend on both N (as $\log M/C_\Delta$) and M. Indeed for a fixed ℓ, the smallest M satisfying (2.19) and (2.20) is achievable. The argument we make below is that for any fixed constant $c = c_0 > 0$, there is an ℓ_0 that depends logarithmically on c_0^{-1} such that the expansion $\log M_t^*(\ell, N) \geq C\ell - c_0$ is true for all $\ell \geq \ell_0$. We first invoke the converse for an $(\ell, M, \infty, 1, 0)$ VLFT code:

Theorem 5 ([PPV11], Thm. 11)**.** *Given an arbitrary DMC with capacity C we have the following for an $(\ell, M, \infty, 1, 0)$ VLFT code:*

$$\log M_t^* \leq \ell C + \log(\ell + 1) + \log e\,. \tag{2.21}$$

After some manipulation, Thms. 4 and 5 imply the following:

Corollary 1. *For an $(\ell, M, N, 1, 0)$ repeated VLFT code, we can pick $\delta > \frac{\log(\ell+1)+\log e}{C}$ and let $N = (1+\delta)\ell = \ell + \Omega(\log \ell)$ such that the following holds for a stationary DMC with capacity*

C:[1]

$$\log M_t^*(\ell, N) \geq \ell C - O(1)\,. \tag{2.22}$$

The proof is provided in Appendix 2.A.

To conclude this discussion of the penalty associated with finite blocklength, we comment that N only needs to be scaled properly, i.e. $N = \ell + \Omega(\log \ell)$, to obtain the infinite-blocklength expansion of $M_t^*(\ell, \infty)$ provided in [PPV11]. Therefore the restriction to a finite blocklength N does not restrict the asymptotic performance if N is selected properly with respect to ℓ. The constant penalty terms in the expansion are different for infinite and finite N, which might not be negligible in the short-blocklength regime. Still, our numerical results in Sec. 2.2.6 indicate that relatively small values of N can yield good results for short blocklengths.

2.2.4 Limited, Regularly-Spaced, Decoding Attempts

This subsection investigates $(\ell, M, N, I, \epsilon)$ VLFT codes with $N = \infty$ but decoding attempted only at specified, regularly-spaced symbols ($I > 1$). The first decoding time occurs after n_1 symbols (which could be larger than I) so that the decoding attempts are made at the times $n_j = n_1 + (j-1)I$. The relevant information density process $i(X^{n_j}; Y^{n_j})$ is on the subsequence $n_j = n_1 + (j-1)I$. The main result here is that the constant penalty now scales linearly with I:

Theorem 6. *For an $(\ell, M, N, I, 0)$ VLFT code with uniform increments I and $N = \infty$ we have the following expansion for a stationary DMC with capacity C:*

$$\log M_t^*(\ell, \infty, I) \geq \ell C - O(I)\,. \tag{2.23}$$

The proof is provided in Appendix 2.A.

In view of Thm. 6, the penalty is linear in the increment I. The increment I can grow slowly, e.g., $I = O(\log \ell)$ and still permit an expected rate that approaches C without the dispersion

[1] *As opposed to the expression in [PPV11], we use a minus sign for $O(1)$ term to make the penalty clear.*

penalty incurred when feedback is absence [PPV10]. In the non-asymptotic regime, however, the increment I must be carefully controlled to keep the penalty small. Our numerical results in Sec. 2.2.6 indicate that choosing $I = \lceil \log_2 \log_2 M \rceil$ yields good results for short blocklengths. Also, varying I_j to decrease with j can avoid a substantial penally while keeping the penalty small.

2.2.5 Finite Blocklength and Limited Decoding Attempts

This subsection investigates $(\ell, M, N, I, 0)$ repeated VLFT codes with *both* finite N and $I > 1$. When these two limitations are combined, a key parameter is m, the number of decoding attempts before the transmission process must start from scratch if successful decoding has not yet been achieved. The parameters m, n_1, N and I are related by the following equation:

$$N = n_1 + (m-1)I \tag{2.24}$$

Combining the results of Sec. 2.2.3 and Sec. 2.2.4 yields the following expansion for $(\ell, M, N, I, 0)$ repeated VLFT codes on a stationary DMC with capacity C:

Theorem 7. *For an $(\ell, M, N, I, 0)$ repeated VLFT with $N = \ell + \Omega(\log \ell)$ on a stationary DMC with capacity C and ℓ sufficiently large*

$$\log M_t^*(\ell, N, I) \geq \ell C - O(I)\,. \tag{2.25}$$

Specifically, if we choose $N > \ell + \frac{\log(\ell+1)+\log e}{C}$ and have decoding attempts separated by an increment I, then the expansion is the same as the case with $I > 0$ and $N = \infty$ so that the penalty term is linear in I.

Proof. The proof is based on the following lemma that provides an upper bound on ℓ for an $(\ell, M, N, I, 0)$ repeated VLFT codes :

Lemma 1. *For an $(\ell, M, N, I, 0)$ repeated VLFT code with $N = \Omega(\log M)$, we have the following upper bound on ℓ for a stationary DMC with capacity C:*

$$\ell \leq (1 + \mathbb{P}[\zeta_N])^{-1} \frac{\log M}{C} + \mathbb{P}[\tau_0 \geq m] + O(I) \tag{2.26}$$

$$\leq \frac{\log M}{C} + O(I)\,, \tag{2.27}$$

where τ_0 is the stopping time in terms of the number of decoding attempts up to and including the first success.

The proof of the lemma is similar to Thm. 4 and the details are in Appendix 2.A.

The proof of Thm. 7 now follows. For an $(\ell, M, N, I, 0)$ repeated VLFT code, pick N as follows:

$$N = (1+\delta)\ell, \text{ where } \delta > \frac{\log(\ell+1) + \log e}{\ell C}\,. \tag{2.28}$$

The result follows by a similar argument as for Cor. 1 and by observing that for the condition of δ, $(1+\delta)\ell = \ell + \Omega(\log \ell)$. The restriction on the initial blocklength n_1 only makes a constant difference. □

Our earlier work [CSW11] used an intuitive choice of the finite blocklength: $N = (1+\delta)\ell$ for a fixed $\delta > 0$, i.e., a blocklength that is larger than the target expected latency by the fraction δ. Theorem 7 validates such choice in terms of the optimality of the asymptotic expansion (since $\delta\ell \in \Omega(\log \ell)$), and indicates that the required overhead δ is decreasing as the target expected latency grows.

2.2.6 Numerical Results

For practical applications that apply feedback to obtain reduced latency, non-asymptotic behavior is critical. This section gives numerical examples of non-asymptotic results for a BSC. For a BSC with transition probability p we use the RCU bound in [PPV10, PPV11][2], which gives the following expression:

$$\xi_n \leq \sum_{t=0}^{n} \binom{n}{t} p^t (1-p)^{n-t} \min\left\{1, M \sum_{j=0}^{t} \binom{n}{j} 2^{-n}\right\}\,.$$

[2] We replace $(M-1)$ by M for simplicity.

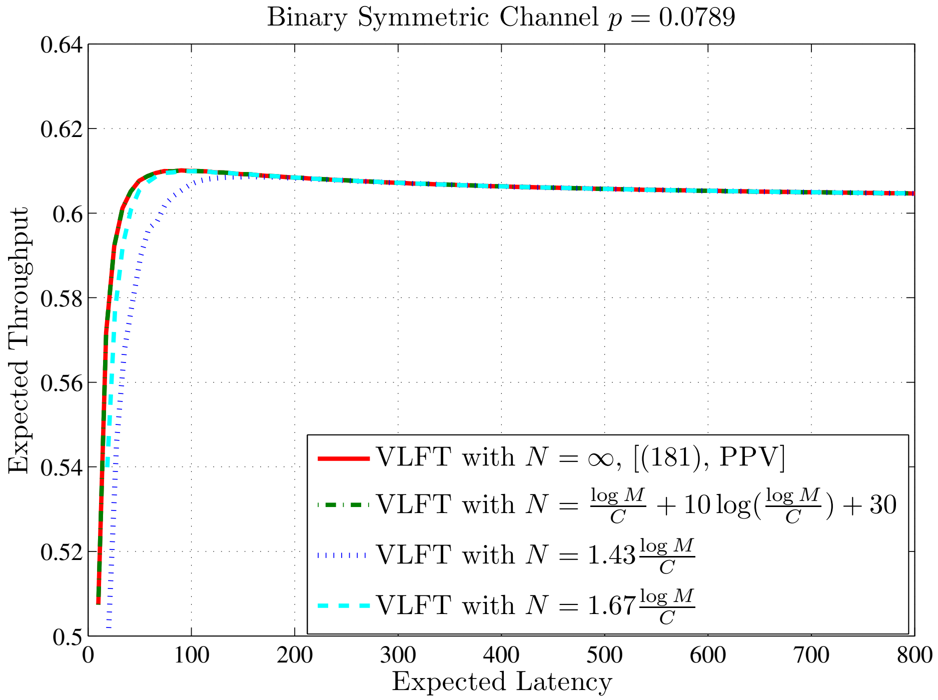

Figure 2.1: Performance comparison of VLFT code achievability based on the RCU bound with different codebook blocklengths.

Fig. 2.1 shows expected throughput (R_t) vs. expected latency (ℓ) performance of a VLFT code with $N = \infty$ and three finite-N repeated VLFT codes over a BSC with $p = 0.0789$. Since ℓ scales linearly with $\frac{\log M}{C}$, for one repeated VLFT code N scales as:

$$N = \frac{\log M}{C} + a \log\left(\frac{\log M}{C}\right) + b\,, \tag{2.29}$$

so that $N = \ell + \Omega(\log \ell)$. The constants $a, b > 0$ were selected experimentally to be $a = 10$, $b = 30$.

For the other two repeated VLFT codes, $N = \log M/C_\Delta$ where $C_\Delta = C - \Delta$. We chose $\Delta = 0.3C$ and $0.4C$, which are 43% and 67% longer, respectively, than the blocklength $N = \log M/C$ that corresponds to capacity. In other words, $N = 1.43 \log M/C$ and $N = 1.67 \log M/C$ respectively.

Expected throughput for the finite-N repeated VLFT codes converges to that of VLFT with $N = \infty$ before expected latency has reached 200 symbols. For expected latency above 75

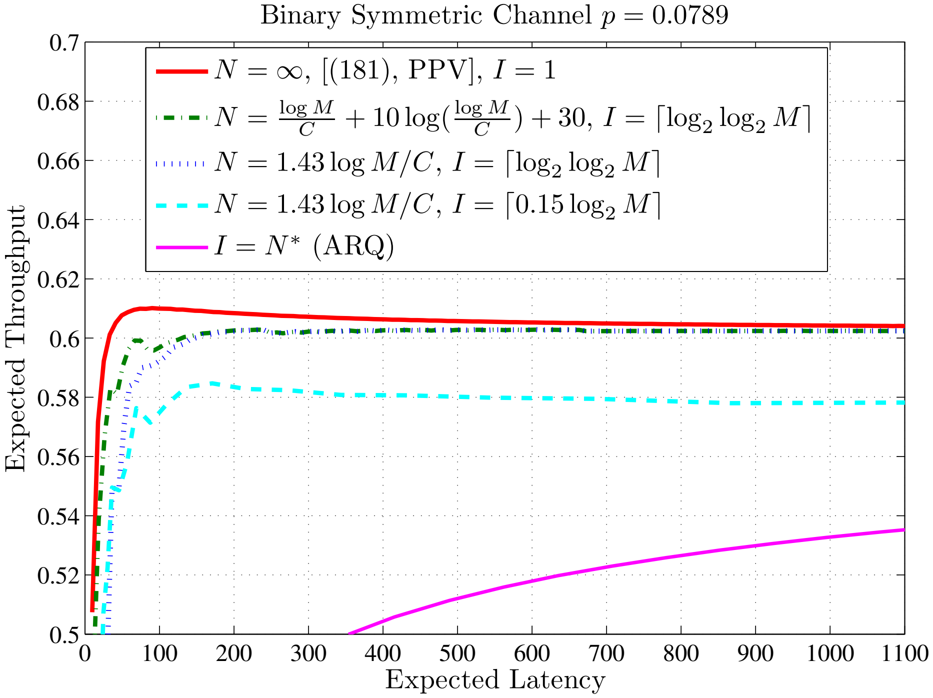

Figure 2.2: Performance comparison of VLFT code achievability based on the RCU bound with uniform increment and finite-length limitations.

symbols, the only repeated VLFT code with visibly different throughput than the $N = \infty$ VLFT code is the $\Delta = 0.3C$ code.

As mentioned in Sec. 2.2.1, VLFT codes can have expected throughput higher than the original BSC capacity of 0.6017 because of the NTC. This effect vanishes as expected latency increases.

Fig. 2.2 shows the R_t vs. ℓ performance of a VLFT code with $N = \infty$ and $I = 1$ and repeated VLFT codes with various decoding-time increments I. As in (2.25), when I grows linearly with $\log M$ (i.e., $\lceil 0.15 \log_2 M \rceil$) then there is a constant gap from the $I = 1$ case. However, if I grows as $\lceil \log_2 \log_2 M \rceil$ then the gap from the $I = 1$ case decreases as expected latency increases. ARQ performance (in which $I = N^*$, where N^* is the optimal blocklength for M) is also shown in the figure, which reveals a considerable performance gap from even the most constrained repeated VLFT implementation in Fig. 2.2.

2.3 IR-NTC with Convolutional and Turbo Codes

The achievability proofs in [PPV11] and 2.2 are based on an IR-NTC scheme using *random* codebooks. This section provides examples of IR-NTC based on *practical* codebooks: rate-compatible families of turbo codes and tail-biting convolutional codes. The constraints of finite N and $I > 1$ studied analytically in the previous section appear naturally in the context of these codes.

2.3.1 Implementation of a repeated IR-NTC with Practical Codes

A practical way to implement repeated IR-NTC is by using a family of rate-compatible codes with incremental blocklengths $\{n_i\}_{i=1}^{m}$ where $n_i = \sum_{j=1}^{i} I_j$. Defining an (n, M) code to be a collection of M length-n vectors taking values in $\mathcal{X}$, we define a family of rate-compatible codes as follows:

Definition 4. *Let $n_1 < n_2 < \cdots < n_m$ be integers. A collection of codes $\{\mathcal{C}_j\}_{j=1}^{m}$ is said to be a family of rate-compatible codes if each $\mathcal{C}_j$ is an (n_j, M) code that is the result of puncturing a common mother code, and all the symbols in the higher-rate code $\{\mathcal{C}_j\}$ are also in the lower rate code $\{\mathcal{C}_{j+1}\}$.*

A family of rate-compatible codes can be constructed by finding a collection of compatible puncturing patterns [Hag88] satisfying Def. 4 for an (N, M) mother code. Note that the puncturing becomes straightforward if we reorder the symbols of the mother code so that the symbols of $\mathcal{C}_1$ are first, followed by the symbols of $\mathcal{C}_2$ and so on. From this perspective, the symbols transmitted by VLFT in [PPV11] can be seen as an infinite family of rate-compatible codes resulting from such an ordered puncturing.

When implemented using a family of rate-compatible codes $\{\mathcal{C}_j\}_{j=1}^{m}$, repeated IR-NTC works as follows. A codeword of $\mathcal{C}_1$ with blocklength $n_1 = I_1$ is transmitted to convey one of the M messages. The decoding result is fed back to the transmitter and an NTC is sent if the decoding is successful. Otherwise the transmitter will send I_2 coded symbols such that the $n_2 = I_1 + I_2$

symbols form the codeword in $\mathcal{C}_2$ representing the same message. The decoder attempts to decode with code $\mathcal{C}_2$ and feeds back the decoding result. If decoding is not successful after the mth transmission where $n_m = N$, the decoder discards all of the previously received symbols and the process begins again with the transmitter resending the I_1 initial coded symbols. This repetition process continues until the decoding is successful.

In the special case where $m = 1$ the repeated IR-NTC reduces to SCR-NTC, which we refer to as ARQ.

2.3.2 Randomly Punctured Convolutional and Turbo Codes

The practical examples of repeated IR-NTC provided in this chapter use tail-biting RCPC codes and RCPT codes. The details of the rate-compatible codes used in this chapter are given as follows:

The two convolutional codes we used in this chapter are a 64-state code and a 1024-state code with generator polynomials $(g_1, g_2, g_3) = (133, 171, 165)$ and $(2325, 2731, 3747)$ in octal, respectively. The 64-state code is from the 3GPP-LTE [Gen08] standard and the 1024-state code is the optimal free distance code from [LC04, Table 12.1b]. Both of the codes are implemented as tail-biting codes [MW86] to avoid rate loss at short blocklengths.

The turbo code used in this chapter is from the 3GPP-LTE standard [Gen08], i.e., the turbo code with generator polynomials $(g_1, g_2) = (13, 15)$ in octal, with a quadratic permutation polynomial interleaver.

Pseudo-random puncturing, also referred to as circular buffer rate matching in [Gen08], provides the rate-compatible families for both the convolutional and the turbo codes. The process is shown in Fig. 2.3: the encoder first generates a rate-$1/3$ codeword. Then the output of each of the encoder's three bit streams passes through a "sub-block" interleaver with a blocklength K. The interleaved bits of each sub-block are concatenated in a buffer, and bits are transmitted sequentially from the buffer to produce the increments I_j as shown in Fig. 2.4. The sub-block interleavers re-order the bits of the mother code so that sequential transmission of the bits produces

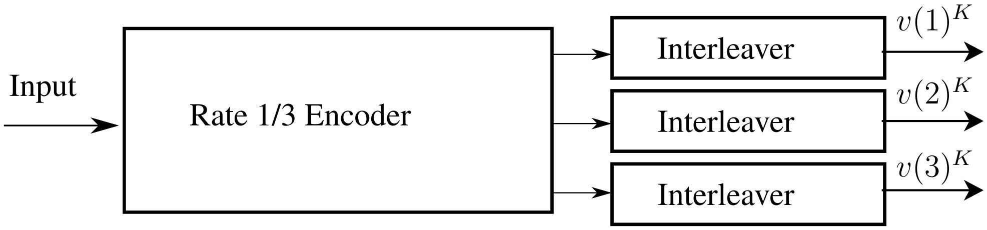

Figure 2.3: Pseudo-random puncturing (or circular buffer rate matching) of a convolutional code. At the bit selection block, a proper amount of coded bits are selected to match the desired code rate.

$$\overbrace{v(1)_1\ldots,v(1)_K,v(2)_1}^{I_1},\underbrace{v(2)_2,\ldots v(2)_9}_{I_2},\ldots,\overbrace{v(2)_K,\ldots,v(3)_K}^{I_m}$$

Figure 2.4: Illustration of an example of transmitted blocks for rate-compatible punctured convolutional codes

an effective family of rate-compatible codes as discussed in Sec. 2.3.1.

We will use simulation results of repeated IR-NTC systems based on these RCPC and RCPT codes to compare with our analysis in the following sections.

2.4 Rate-Compatible Sphere-Packing (RCSP)

The random-coding approach gives a tight achievable bound on expected latency when M is sufficiently large. In the short-latency regime, however, practical codes can outperform random codes, as noted in [WCW12]. To find a code-independent analysis that gives a better prediction of practical code performance, we introduce the rate-compatible sphere-packing (RCSP) approximation. As we will see in Sec. 2.5, RCSP can also facilitate the optimization of the increment lengths I_j and provide a trajectory of target error rates for use in the design of a rate-compatible code family.

Shannon et al. [SGB67] derived lower bounds of channel codes for DMC by packing typical sets into the output space. The typical sets are related to the divergence of the channel and an auxiliary distribution on the output alphabet. For the AWGN channel, Shannon [Sha59] showed both the lower bounds on the error probability by considering optimal codes on a sphere (the surface of the relevant ball). The bound turns out to be tight even in the finite-blocklength regime as shown in [PPV10]. One drawback for considering codes on a sphere is the computational difficulty involved, even for a single fixed-length code.

RCSP is an approximation of the performance of repeated IR-NTC using a family of rate-compatible codes. The idea of RCSP is an extension of the sphere-packing lower bound from a single fixed-length code to a family of rate-compatible codes. For the ideal family of rate-compatible codes, each code in the family would achieve perfect packing. Our analysis will involve two types of packing: 1) perfect packing throughout the volume of the ball whose radius is determined from the signal and noise powers or 2) perfect packing on the surface of the ball whose radius is determined by the signal power constraint. We will also consider both maximum-likelihood (ML) decoding and bounded-distance (BD) decoding.

Let $\{\mathcal{C}_j\}_{j=1}^{m}$ be a family of rate-compatible codes. Let the marginal error event of the code $\mathcal{C}_j$ at blocklength n_j be ζ_{n_j} and let the joint error probabilities $\mathbb{P}[E_{n_j}]$ be defined similar to (2.15):

$$E_{n_j} = \cap_{i=1}^{j} \zeta_{n_i} \, . \tag{2.30}$$

The expected latency for a repeated IR-NTC can be computed as follows:

$$\ell = \frac{I_1 + \sum_{j=2}^{m} I_j \mathbb{P}[E_{n_{j-1}}]}{1 - \mathbb{P}[E_{n_m}]}. \tag{2.31}$$

Applying the ideal of RCSP throughout the volume of the ball to (2.31) leads to the joint RCSP approximation of the repeated IR-NTC performance. Since $\mathbb{P}[\zeta_{n_j}] \geq \mathbb{P}[E_{n_j}]$, replacing $\mathbb{P}[E_{n_j}]$ with $\mathbb{P}[\zeta_{n_j}]$ produces a upper bound on expected latency as follows:

$$\ell \leq \frac{I_1 + \sum_{j=2}^{m} I_j \mathbb{P}[\zeta_{n_{j-1}}]}{1 - \mathbb{P}[\zeta_{n_m}]}. \tag{2.32}$$

Since $\mathbb{P}[\zeta_{n_j}]$ is often a tight upper bound on $\mathbb{P}[E_{n_j}]$ (examples will be shown in Sec. 2.5.3), applying the ideal of RCSP throughout the volume of the ball to (2.32) produces the marginal RCSP approximation of the expected latency in a repeated IR-NTC, which is very similar to the joint RCSP approximation and more easily computed.

Performing ML decoding gives a lower bound on the expected latency for the repeated IR-NTC, but makes the error probabilities difficult to evaluate. We initially analyze using BD decoding to decode the ideal code family $\{\mathcal{C}_j\}_{j=1}^m$. Subsequently we refine the analysis by bounding ML decoding performance. Thus we will be considering both the joint and marginal RCSP approximations and also considering both BD and ML decoding.

2.4.1 Marginal RCSP Approximation for BSC

For the BSC the optimal decoding regions are simply Hamming spheres. RCSP upper bounds IR-NTC performance on the BSC by assuming that each code in the family of rate-compatible codes achieves the relevant Hamming bound.

For a BSC with transition probability p the marginal error probability $\mathbb{P}[\zeta_{n_j}]$ is lower bounded as follows:[3]

$$\mathbb{P}[\zeta_{n_j}] \geq \sum_{t=r_j+1}^{n_j} \binom{n_j}{t} p^t (1-p)^{n-t}, \tag{2.33}$$

where r_j is chosen such that

$$M \sum_{t=0}^{r_j-1} \binom{n_j}{t} + \sum_{i=1}^{M} A_i = 2^{n_j} \tag{2.34}$$

and

$$0 < \sum_{j=1}^{M} A_i < M \binom{n_j}{r_j}. \tag{2.35}$$

We use (2.33) to compute the marginal RCSP approximation for the BSC. Note that for the BSC and an (n, M) code with M uniform decoding regions that perfectly fill the space, ML decoding is also BD decoding with an uniform radius.

[3] Formally, the sphere-packing lower bound for BSC follows from the fact that the tail of a binomial r.v. is convex.

For the BSC with $p = 0.0789$, Fig. 2.5 shows the marginal RCSP approximation with $m = \infty$, the VLFT converse of [PPV11], the random-coding achievability of [PPV11], and simulations of repeated IR-NTC using the 64-state convolutional code from Sec. 2.3.2. All simulation points have increment $I = 1$, finite codeword lengths $N = 3k$, and initial blocklength $n_1 = k$ for $k = 16, 20, 32, 64$, respectively. To compare the choice of $N = 3k$ with the results in Sec. 2.2, take $k = 32$ as an example. For $k = 32$ the expected latency is $\ell = 49.62$ in Fig. 2.5 and $N = 96$ corresponds to choosing a $\delta = 0.93$. As we saw in Fig. 2.1, $\delta = 0.67$ gives RCU performance indistinguishable from $N = \infty$ so that $N = 96$ should be more than sufficient for this example.

The convolutional code simulations give throughput-latency points that are very close to the marginal RCSP approximation for expected latency less than 50. The simulation points for $k = 16, 20, 32$ are significantly higher than the random-coding achievability result of [PPV11]. The simulation point for $k = 64$ falls below the random-coding achievability because the expected latency is larger than the analytic trace-back depth of 60 bits (or 20 trellis state transitions) for the 64-state convolutional code.

Note that in Fig. 2.5, the convolutional code simulation points, the marginal RCSP approximation curve, the VLFT achievability, and the VLFT converse curves are all above the BSC capacity because the NTC assumption provides additional information to the receiver as discussed in Sec. 2.2.1.

2.4.2 RCSP Approximation for AWGN Channel

This subsection derives joint and marginal RCSP approximations for the AWGN channel under both BD and ML decoding. Consider an AWGN channel $Y = X + Z$ with an average power constraint P[4]. Let the signal-to-noise ratio (SNR) be $\eta = P/\sigma^2$ where P is the signal power and

[4] We use the expectation average power constraint $\mathbb{E}\left(\sum_{j=1}^{n} X_j^2\right) \leq nP$, which is not as strict as the summation average power constraint $\sum_{j=1}^{n} X_j^2 \leq nP$ since it allows the codeword power be larger than P with low probability. As shown in the Appendix 2.C, the sphere-packing property combined with the expectation average power constraint will satisfy the summation average power constraint if the rate is less than capacity.

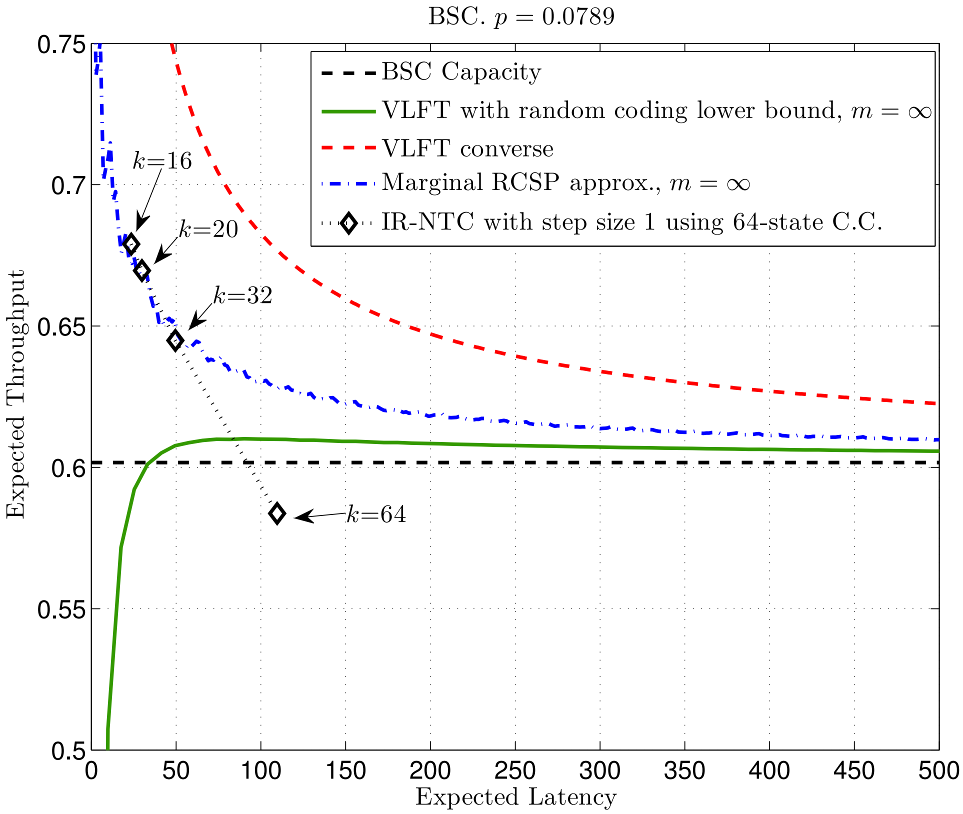

Figure 2.5: In the short-latency regime, the marginal RCSP approximation can be more accurate for characterizing the performance of good codes (such as the convolutional code in this figure) than random-coding analysis. The additional information provided by the error-free termination symbol of NTC leads to a converse and an operational rate for the convolutional code that are above the original BSC capacity.

σ^2 is the noise power. Assume without loss of generality (w.l.o.g.) that each noise sample has a unit variance. The average power of a length-n received word is $\mathbb{E}[\|Y^n\|^2] \leq n(P + \sigma^2) = n(1 + \eta)$. Sphere packing seeks a codebook that has M equally separated codewords within the n-dimensional norm ball with radius $r_{\text{outer}} = \sqrt{n(1 + \eta)}$.

One can visualize a large outer sphere that contains $M = 2^{nR}$ decoding regions, $D_i, i = 1, \ldots, M$, each with the same volume. Considering the spherical symmetry of i.i.d. Gaussian noise Z, our bounded-distance decoding region is an n-dimensional Euclidean ball centered around the message point. By conservation of volume, the largest radius of the decoding region has a volume satisfying

$$\begin{aligned}\text{Vol}(D_i) &= K_n r^n \\ &\leq 2^{-nR} \times \text{Vol(Outer sphere)} \\ &= 2^{-nR} K_n \left(\sqrt{n(1+\eta)}\right)^n\end{aligned}$$

where K_n is the volume of the n-dimensional unit sphere. Solving for the radius of the decoding region yields

$$r \leq 2^{-R}\sqrt{n(1+\eta)}. \tag{2.36}$$

2.4.2.1 RCSP Approximation for AWGN under BD decoding

Using bounded-distance decoding on the AWGN channel gives the following decoding rule: the decoder selects message j if the received word $Y^n \in D_i$ for a unique $i \in \{1, \dots, M\}$ and declares an error otherwise. Regardless of the transmitted codeword, the marginal error event for BD decoding is $\zeta_n = \{Z^n : \|Z^n\|^2 > r^2\}$ where r is the decoding radius. The error probability for decoding region with radius r is then simply given by the tail of a chi-square random variable with n degrees of freedom:

$$\mathbb{P}[\zeta_n] = 1 - F_{\chi_n^2}(r^2) \tag{2.37}$$

$$= G_{\chi_n^2}(r^2) \tag{2.38}$$

where $F_{\chi_n^2}$ is the CDF of a chi-square distribution with n degrees of freedom.

Let $M = 2^k$ for an integer k. Assuming perfect sphere-packing (i.e., achieving (2.36) with equality) at each incremental code rate, the radius at incremental code rate k/n_j is given as:

$$r_j^2 = \frac{n_j(1+\eta)}{2^{2k/n_j}}. \tag{2.39}$$

Note that the probability of a decoding error in the jth transmission depends on previous error events. Conditioning on previous decoding errors $\zeta_{n_j}, j = 1\dots, m-1$ makes the error event

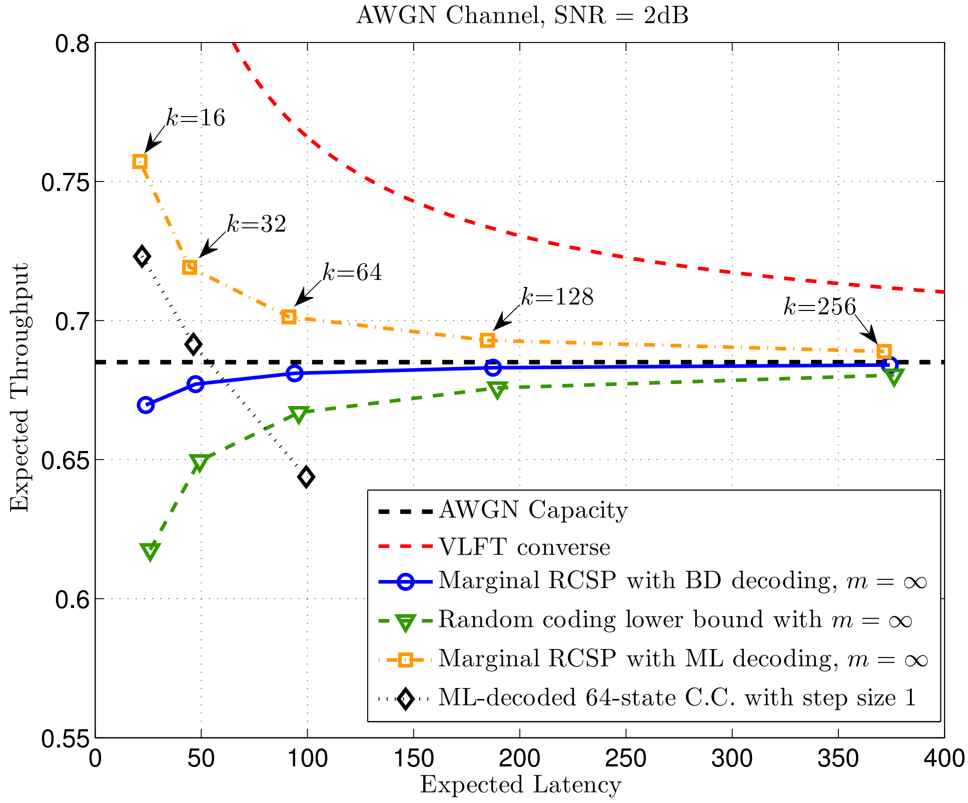

Figure 2.6: The marginal RCSP approximation with BD decoding, and marginal RCSP with ML decoding, for the AWGN channel. Similar to the BSC, the additional information provided by the error-free termination symbol of NTC leads to a converse and an operational rate for the convolutional code that are above the original BSC capacity.

ζ_{n_m} more likely than the marginal distribution would suggest. Recall that E_{n_m} denotes $\cap_{j\le m}\zeta_{n_j}$; the joint probability $\mathbb{P}[E_{n_j}], 1 \le j \le m$ is given as:

$$\mathbb{P}[E_{n_j}] = \int_{r_1^2}^{\infty}\int_{r_2^2-t_1}^{\infty}\dots\int_{r_{j-1}^2-\sum_{i=1}^{j-2}t_i}^{\infty} G_{\chi^2_{I_j}}\left(r_j^2-\sum_{i=1}^{j-1}t_i\right) dF_{\chi^2_{I_1}}(t_1)\dots dF_{\chi^2_{I_{j-1}}}(t_{j-1}), \quad (2.40)$$

where the increments I_j are as defined in Sec. 2.3.1 and $G_{\chi^2_{I_j}}$ is defined in (2.38).

Using $\mathbb{P}[E_{n_j}]$ or $\mathbb{P}[\zeta_{n_j}]$ as derived above, we can compute the expected latency ℓ of the repeated IR-NTC by using the joint BD RCSP approximation as in (2.31) or the marginal BD RCSP approximation as in (2.32). The expected throughput R_t is given by k/ℓ.

2.4.2.2 Marginal RCSP Approximation under ML decoding

To compute the marginal RCSP approximation with ML-decoding, we apply Shannon's sphere-packing lower bound[5] [SGB67]. We use the asymptotic approximation in [SGB67] to estimate the lower bounds of the marginal error probabilities. The approximation formula in [SGB67] does not work well for rates above capacity, providing negative values that are trivial lower bounds. To obtain a good estimate in these cases we replace those trivial probabilities with the ones suggested by the marginal RCSP approximation using BD decoding.

For the 2dB AWGN channel, Fig. 2.6 shows the VLFT converse of [PPV11], the random-coding lower bound of [PPV11], the marginal RCSP with BD decoding, and marginal RCSP with ML decoding, all with $m = \infty$ and $I = 1$. To approximate the behavior of $m = \infty$ we found numerically that $m = 10k$ is sufficient in this example. Also shown in Fig. 2.6 is the simulation of the repeated IR-NTC with step size $I = 1$ and $k = 16, 32$, and 64 using the 64-state convolutional code from Sec. 2.3.2.

Similar to the BSC example, Fig. 2.6 shows that the VLFT converse based on Fano's inequality [PPV11] and the marginal RCSP approximation with ML decoding are both above the asymptotic capacity due to the information provided by NTC. The marginal RCSP approximation with BD decoding is not as accurate as the curve with ML decoding in the short-latency regime. The difference between the two curves, however, becomes small beyond the expected latency of 200.

For $k = 16$ and 32, the convolutional code simulations of repeated IR-NTC (even with a finite N of $3k$) with binary modulation outperform the corresponding points on the curve for VLFT random-coding achievability with $N = \infty$ and unconstrained input to the channel. The two simulation points generally follow the curve of the marginal RCSP approximation with ML decoding, but with a gap. We suspect that this gap is due to the use of binary modulation rather than an unconstrained input in the simulation. As with the BSC example, for $k = 64$

[5] We bound the error probability using Shannon's argument: the optimal error probability for codewords inside a ball with radius $\sqrt{nP}$ is lower bounded by the optimal error probability for codewords with one extra dimension lying on a sphere surface with radius $\sqrt{(n+1)P}$.

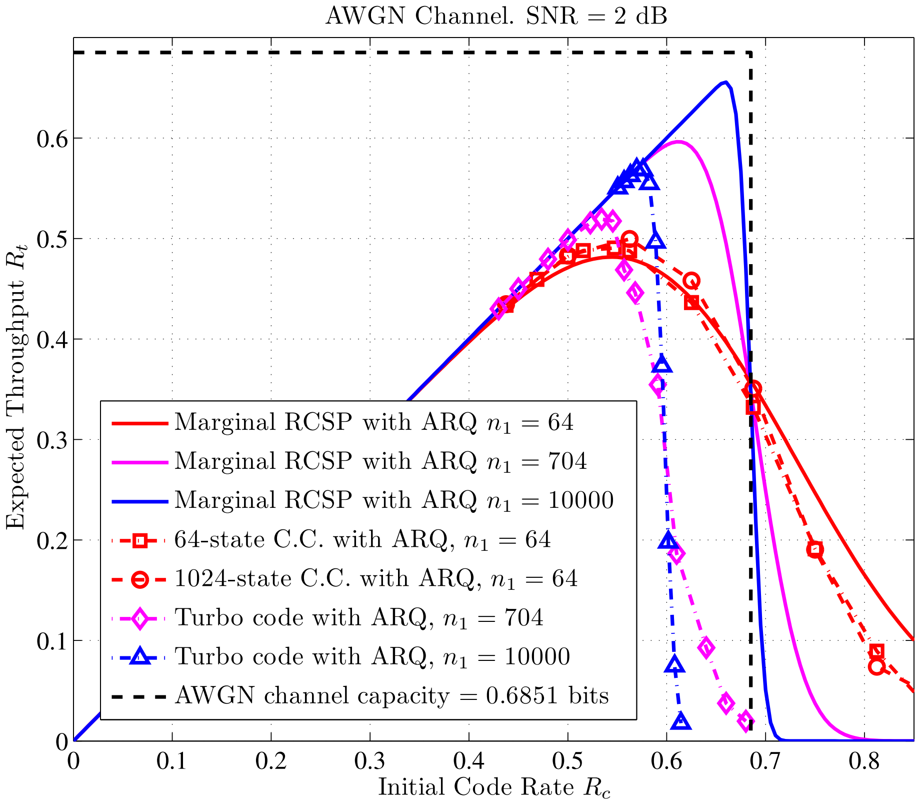

Figure 2.7: ARQ for RCSP, turbo code, 64-state and 1024-state convolutional codes simulations with different initial blocklengths.

the performance of the simulation falls short as the expected latency goes beyond the analytic trace-back depth of the 64-state convolutional code.

2.4.3 Marginal BD RCSP for ARQ and IR-NTC in AWGN

This subsection provides examples of applying the marginal RCSP approximation using BD decoding for the AWGN channel. Note that all examples use BD decoding and we will simply use the term “marginal RCSP approximation”.

2.4.3.1 ARQ

Consider ARQ (SCR-NTC) on an AWGN channel. Based on the marginal RCSP approximation the expected latency ℓ is given as

$$\ell = \frac{n_1}{1 - \mathbb{P}[\zeta_{n_1}]} \tag{2.41}$$

$$= \frac{n_1}{F_{\chi^2_{n_1}}(r_1^2)}, \tag{2.42}$$

and expected throughput $R_t = k/\ell$ is given as

$$R_{t,\text{ARQ}} = R_c F_{\chi^2_{n_1}}(r_1^2) \tag{2.43}$$

where $R_c = k/n_1$ is the initial code rate and r_1 is the decoding radius of the codeword with length n_1. Note that in the case of ARQ (SCR-NTC), the joint RCSP and marginal RCSP approximations are identical since there is only a single transmission before repetition.

Fig. 2.7 shows the R_t vs. R_c (expected throughput vs. initial code rate) curve for the RCSP with ARQ. The SNR is 2 dB and the initial blocklengths are $n_1 = 64$, $n_1 = 704$ and $n_1 = 10,000$. For each curve in Fig. 2.7, the initial blocklength n_1 is fixed as we vary the number of messages $M = 2^k$ and the initial code rate $R_c = k/n_1$ changes accordingly. Note that for ARQ, the initial blocklength is also the lengths of the possible subsequent transmissions.

For blocklengths $n_1 = 704$ and $n_1 = 10000$, Fig. 2.7 compares RCSP with the 3GPP-LTE turbo codes. Each point of the dash-dot turbo-code curves represents a different turbo code with the same blocklength but different code rate (different k). Interestingly, after the initial negligible-codeword-error region where expected throughput equals code rate, the RCSP curve and the turbo code curve are very similar for both $n_1 = 704$ and $n_1 = 10,000$. Furthermore, in this region the difference between the code rate R_c associated with a given throughput for RCSP and for the turbo code is about 0.1 bits for both $n_1 = 704$ and $n_1 = 10,000$ despite the large difference between these two blocklengths.

For blocklength $n_1 = 64$, Fig. 2.7 compares the marginal RCSP approximation to the performance of 64-state and 1024-state tail-biting convolutional codes. The marginal RCSP approx-

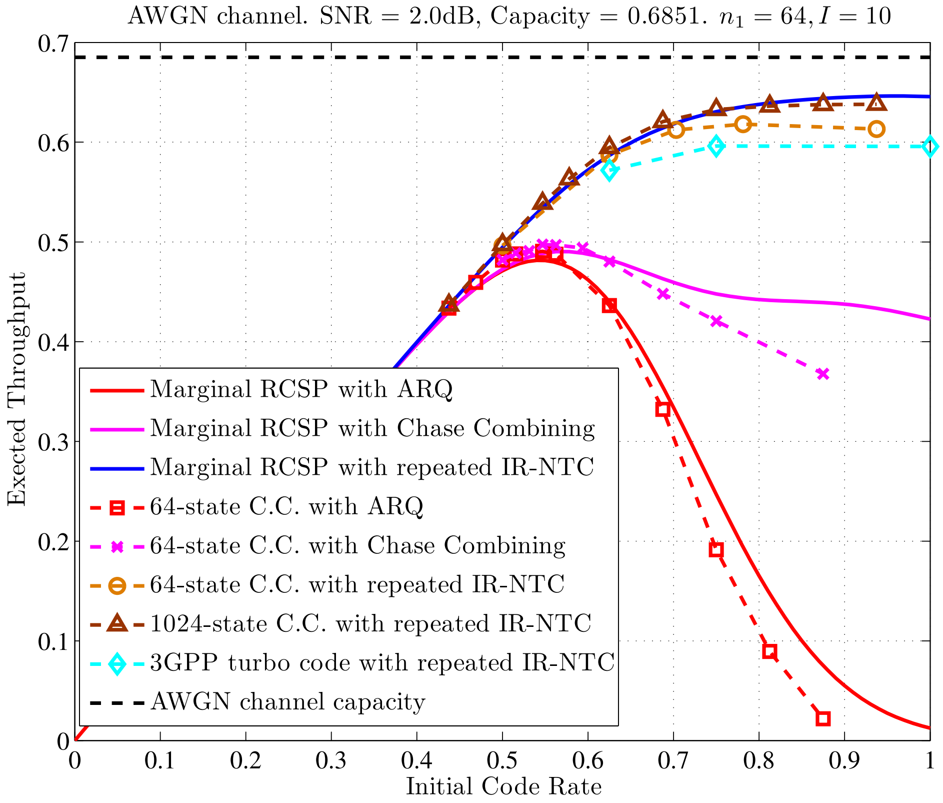

Figure 2.8: R_t vs. R_c for ARQ, ARQ with Chase combining and IR-NTC with $n_1 = 64$.

imation closely predicts the performance of these good tail-biting convolutional codes. Thus, while the turbo codes achieve higher throughputs than the convolutional codes, the convolutional codes perform closer to the RCSP approximation for their (short) blocklengths than do the turbo codes relative to the RCSP approximation for their (longer) blocklengths.

RCSP with its optimistic decoding radius r_1 and suboptimal bounded-distance decoding is a mixture of optimistic and pessimistic assumptions. These plots, however, show that RCSP can provide accurate guidance for good short-blocklength codes such as tail-biting convolutional codes.

2.4.3.2 Chase Combining and IR-NTC

In the ARQ scheme, the same complete codeword is transmitted at each retransmission. One way to utilize these repetitions is to apply the combining scheme proposed by Chase [Cha85]. The Chase scheme uses maximal ratio combining of L repeated codewords at the receiver. If the L codewords are transmitted directly through the AWGN channel with the same SNR, the combining process increases the effective SNR at the receiver by a factor of L.

The side information that the previous block was decoded unsuccessfully (available from the absence of the NTC) implies that the instantaneous noise power in the previous packet is larger than the expected noise power. The error probability is lower bounded (and the throughput is upper bounded) by ignoring this side information.

As shown in Fig. 2.8, Chase combining of all received packets for a message does not significantly increase the highest possible throughput. Note that the RCSP curve for Chase combining uses the throughput upper bound that ignores the side information of previous decoding failures. Chase combining does provide a substantial throughput improvement for higher-than-optimal initial code rates, but these improved throughputs could have also been achieved by using a lower initial code rate and no Chase combining.

We now present an RCSP approximation and code simulations of repeated IR-NTC as described in Sec. 2.3.1. The exact computation of the joint RCSP approximation is challenging when m is large. We use the marginal RCSP approximation here to compute the R_t vs. R_c curve for the repeated IR-NTC.

Fig. 2.8 shows an R_t vs. R_c curve computed based on the approximation of RCSP for the repeated IR-NTC scheme with $n_1 = 64$, $m = 10$ and a uniform increment $I = 10$. We use a uniform increment to avoid the need to optimize each increment although the computation also admits non-uniform increments. We choose $m = 10$ because experimental results show that for $m > 10$ the throughput improvement is diminishing. Note that we include AWGN channel capacity in Fig. 2.8 only as a point of reference since finite-latency capacity with NTC is higher than the asymptotic capacity as shown in Sec. 2.4.2.

Fig. 2.8 also shows repeated IR-NTC simulations of two tail-biting convolutional codes and a turbo code, where the relevant codes are described in Sec. 2.3.2. The simulated turbo code saturates at a lower throughput than the 64-state and 1024-state tail-biting convolutional codes, which are both ML decoded. We expect that turbo codes with better performance at short latencies can be found. Still, the convolutional code performance is outstanding in this short-latency regime.

Fig. 2.8 shows that ARQ achieves a throughput of less than 0.5 bits. and Chase combining provides little improvement over ARQ, with less than 0.01 bits increase in expected throughput. In contrast, the 1024-state convolutional code simulation of repeated IR-NTC achieves a throughput of 0.638 bits. This coincides with the observation in [PPV11] that IR is essential in achieving high throughput with feedback in the finite-blocklength regime, even though the SCR scheme proposed by Yamamoto and Itoh [YI79] achieves the optimal error-exponent shown by Burnashev [Bur76].

We conclude this section by comparing Fig. 2.7 and Fig. 2.8. As shown in Fig. 2.7, the expected throughput of an ARQ using the blocklength-$10,000$ turbo code is 0.56 bits with an expected latency of $10,016$ bits. Fig. 2.8 shows that the repeated IR-NTC using the 1024-state convolutional code achieves a higher expected throughput of 0.64 bits with an expected latency of only 100 bits!

2.5 Optimization of Increments

Sec. 2.4.3 studied throughput by fixing the initial blocklength n_1 and varying the number of information symbols k, which correspondingly varied the initial code rate R_c. This produces curves of expected throughput R_t vs. initial code rate R_c. In contrast, this section fixes k and studies throughput by optimizing the set of increments $\{I_j\}_{j=1}^{m}$ for the repeated IR-NTC scheme as presented in Sec. 2.3.1. This produces curves of expected throughput R_t vs. expected latency ℓ. The optimization uses BD decoding to compute the expected throughput for both joint and marginal RCSP approximation. For the rest of the chapter we will assume BD decoding unless otherwise stated. However, numerical results for ML decoding are also presented and compared to BD decoding.

We first consider exact computations of the joint RCSP approximation for relatively small values of m. We begin with the special case of $m = 1$, which is ARQ. We then provide results for cases where $m > 1$ and study how increasing m improves performance. We introduce Chernoff bounds on the joint RCSP approximation and compare these bounds with the marginal RCSP approximation. We then use the marginal RCSP approximation to optimize the performance of repeated IR-NTC for large m but constrained to have uniform increments after the initial transmission. Then we optimize increments for non-repeating IR-NTC constraining both the maximum number of incremental transmissions and the probability of outage. Finally, we introduce the concept of the decoding error trajectory, which provides the RCSP approximation of error probability at each incremental transmission. It is a useful guide for rate-compatible code design for feedback systems.

2.5.1 Choosing I_1 for the $m = 1$ Case (ARQ)

Recall that for repeated IR-NTC the special case of $m = 1$ is ARQ. In this case, when the message is fixed to be k bits, the RCSP approximation[6] of expected throughput is a quasi-concave function of the code rate R_c in (2.43). Thus a unique optimal code rate R_c^* for the repeated codewords can be found numerically [BV04] to maximize the RCSP approximation of R_t for a given k.

Table 2.1 presents the optimal code rates $R_c^* = k/I_1^*$ and transmission lengths $n_1 = I_1^*$ for ARQ. Fig. 2.9 plots the maximum RCSP approximation of throughput vs. expected latency ℓ for ARQ as the red curve with diamond markers. Both Table 2.1 and Fig. 2.9 apply the constraint that the lengths I_1^* must be integers. These results for ARQ are discussed together with the results of $m > 1$ in the next subsection.

[6] Note that for the $m = 1$ case there is no distinction between the joint RCSP approximation and the marginal RCSP approximation.

Table 2.1: Optimized transmission lengths $n_1 = I_1^*$ and initial code rates R_c^* for ideal-sphere-packing ARQ with information lengths k and SNR $\eta = 2$dB.

k	16	32	64	128	256
$n_1 = I_1^*$	31	60	116	222	429
R_c^*	0.516	0.533	0.552	0.577	0.597

2.5.2 Choosing Increments $\{I_j\}$ to Maximize Throughput

In Sec. 2.4.3.2 we demonstrated one repeated IR-NTC scheme with $m = 10$ transmissions that could approach capacity with low latency based on RCSP. Specifically, the transmission lengths were fixed to $I_1 = 64$ and $I_2, \ldots, I_{10} = 10$, while k was varied to maximize throughput.

This subsection presents optimization results based on exact numerical integrations computing the joint RCSP approximation. Both k and the number of transmissions m are fixed, and a search identifies the set of transmission lengths $\{I_j\}_{j=1}^m$ that maximizes the joint RCSP approximation of expected throughput using BD decoding. Based on the optimized increments, this subsection also provides the marginal RCSP approximation of the expected throughput using ML decoding.

For $m > 1$, identifying the transmission lengths I_j which minimize the latency ℓ is not straightforward due to the joint decoding error probabilities in (2.40). Restricting to a small m allows exact computation of (2.40) in Mathematica, avoiding the marginal approximation. We study the cases when $m \leq 6$ based on numerical integration.

The computational complexity of numerical integration for (2.40) increases with the transmission index j. Because of this increasing complexity we limited attention to a well-chosen subset of possible transmission lengths. Thus our results based on numerical integration may be considered as lower bounds to what is possible with a fully exhaustive optimization.

Table 2.2 shows the results of the $m = 5$ optimization (i.e., the set of lengths I_j found to achieve the highest throughput) and the corresponding throughput for the joint RCSP approx-

Table 2.2: Optimized RCSP transmission lengths for $m = 5$ and SNR$= 2$ dB using non-uniform increments.

k	I_1	I_3	I_3	I_4	I_5	R_t Opt.
16	19	4	4	4	8	0.6019
32	38	8	8	8	12	0.6208
64	85	12	8	12	16	0.6363
128	176	14	14	14	28	0.6494
256	352	24	24	24	48	0.6593

imation. Table 2.2 also shows that for every value of k the initial code rate k/I_1 is above the channel capacity of 0.6851. These high initial code rates indicate that feedback is allowing the decoder to capitalize on favorable noise realizations by attempting to decode and terminate early.

Fig. 2.9 shows the optimized joint RCSP approximation of R_t vs. ℓ for $m = 1, 2, 5, 6$ on an AWGN channel with SNR 2 dB. As m increases, each additional retransmission increases the expected throughput but the amount of that increase diminishes. The points on each curve in Fig. 2.9 represent values of k ranging from 16 to 256 information bits. Fig. 2.9 shows, for example, that by allowing up to four retransmissions ($m = 5$) with $k = 64$, the joint RCSP approximation has a throughput $R_t = 0.636$ bits or 93% of the original AWGN capacity[7] with an expected latency of 101 symbols. Similar results are obtained for other SNRs.

Fig. 2.9 also shows the R_t vs. ℓ curves for marginal RCSP approximation using ML decoding. The increments in Table 2.2 are used in the computation. The curves for ML decoding shows that the effect of NTC starts to manifest at low latencies as m increases. Moreover, the differences between the BD decoding curves and ML decoding curves are negligible for $m > 1$ and expected latencies larger than 200. This observation motivates us to focus on the expected throughput optimization using BD decoding, which simplifies the computation.

[7] As shown in Sec. 2.4 the finite-latency capacity with NTC is higher than the asymptotic capacity. However, for small m the capacity increase due to NTC is small, and so we include the AWGN capacity as a point of reference.

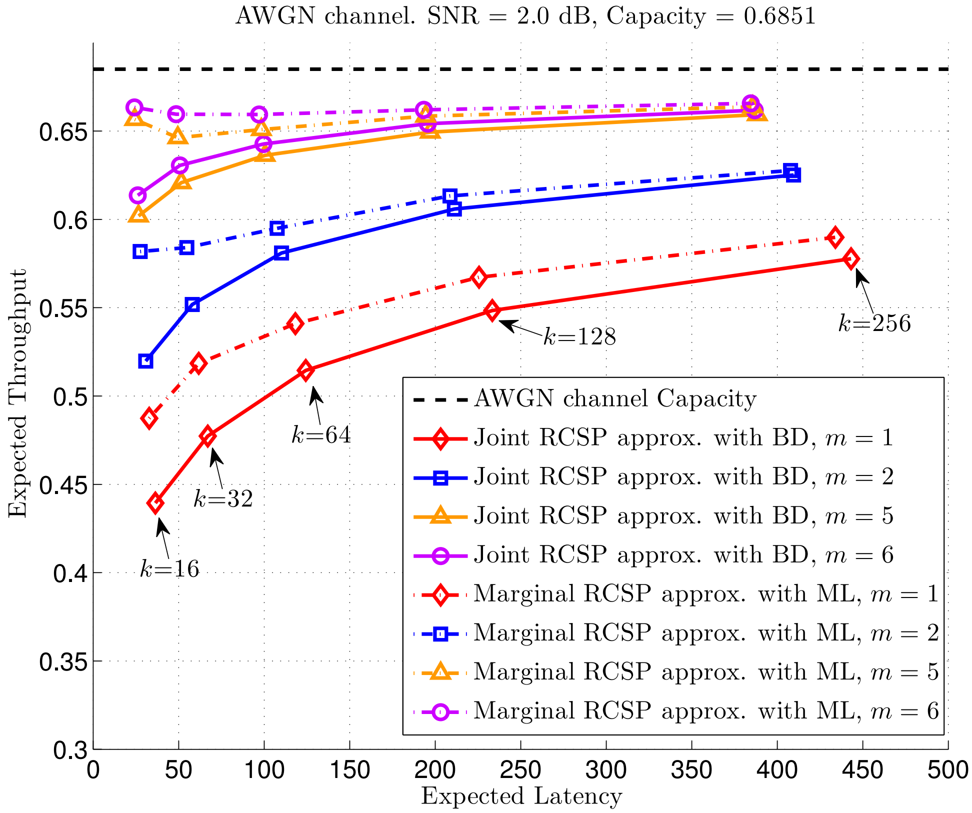

Figure 2.9: The R_t vs. ℓ curves using the joint RCSP approximation with BD decoding and the marginal RCSP approximation with ML decoding for $m = 1, 2, 5, 6$. The transmission lengths $\{I_j\}$ are identified by joint RCSP approximation.

2.5.3 Chernoff Bounds for RCSP over AWGN Channel

Even using the less complex BD decoding, the computation of the joint RCSP approximation based on numerical integration becomes unwieldy for $m > 6$. In this subsection we study upper and lower bounds based on the Chernoff inequality and compare these bounds with the marginal RCSP approximation of throughput, which is itself a lower bound on the joint RCSP approximation. Similar to previous subsection we assume BD decoding unless otherwise stated.

Assume w.l.o.g. that the noise has unit variance. Let r_j and n_j be the decoding radius and blocklength for the jth decoding attempt. As in earlier sections let ζ_{n_j} be the marginal error event and $E_{n_i} = \cap_{j=1}^{i} \zeta_{n_j}$ is the joint error event. The main result of applying the Chernoff inequality to bound the joint RCSP approximation is the following theorem.

Theorem 8. *Using the joint RCSP approximation with BD decoding for AWGN channel we have for all* $1 < i \leq m$ *that*

$$\mathbb{P}[E_{n_i}] \geq \max\left\{0, \mathbb{P}[\zeta_{n_i}] - \sum_{j=1}^{i-1} \mathbb{P}\left[\zeta_{n_i} \cap \zeta_{n_j}^c\right]\right\} , \tag{2.44}$$

$$\mathbb{P}[E_{n_i}] \leq \min\{P_1, P_2, 1\} , \tag{2.45}$$

where $P_1 = \mathbb{P}[\zeta_{n_i}]$ *and* $P_2 = \mathbb{P}[\zeta_{n_i} \cap \zeta_{n_{i-1}}]$. *The pairs of joint events* $\zeta_{n_m} \cap \zeta_{n_j}, j = 1, \ldots, m-1$ *can be bounded as follows:*

$$\mathbb{P}[\zeta_{n_j} \cap \zeta_{n_m}] \leq \inf_{0 \leq u < 1/2} \frac{\mathbb{P}\left[\chi^2_{n_j} > (1-2u)r_m^2\right]}{e^{ur_m^2}(1-2u)^{n_m/2}} , \tag{2.46}$$

$$\mathbb{P}[\zeta_{n_j}^c \cap \zeta_{n_m}] \leq \inf_{0 \leq u < 1/2} \frac{\mathbb{P}\left[\chi^2_{n_j} \leq (1-2u)r_m^2\right]}{e^{ur_m^2}(1-2u)^{n_m/2}} . \tag{2.47}$$

The bounds on pairs of joint events (2.46) *and* (2.47) *can be extended to joints of more than two events which leads to a slightly tighter upper bound for* $\mathbb{P}[E_{n_i}]$. *The proof of this extension and the proof of Thm. 8 are provided in Appendix 2.B.*

We observed numerically that the marginal probability $\mathbb{P}[\zeta_{n_m}]$ in (2.45), which can be evaluated directly via the tail of a single chi-square random variable, is surprisingly tight for short blocklengths. The tightness of the marginal was used in [PPV11] for (2.6), where the upper bound on the error probability of each time instance is the marginal.

Using the marginal $\mathbb{P}[\zeta_{n_i}]$ as an upper bound on $\mathbb{P}[E_{n_i}]$, the lower bound (2.44) shows that the gap between the joint and marginal probability is upper bounded by applying (2.47) to $\sum_{j=1}^{i-1} \mathbb{P}\left[\zeta_{n_i} \cap \zeta_{n_j}^c\right]$. We found numerically that setting $u = 1/2 - n_m/(2r_m^2 + 2k)$ in (2.47) where $k = \log_2 M$, gives a tight upper bound on the gap, although this convenient choice of u is not the optimal value.

Fig. 2.10 shows the R_t vs. ℓ curves for $m = 5$ using the optimized step sizes provided in Table 2.2 of Sec. 2.5 and the values of k are shown in the figure. The channel SNR is 2 dB and the

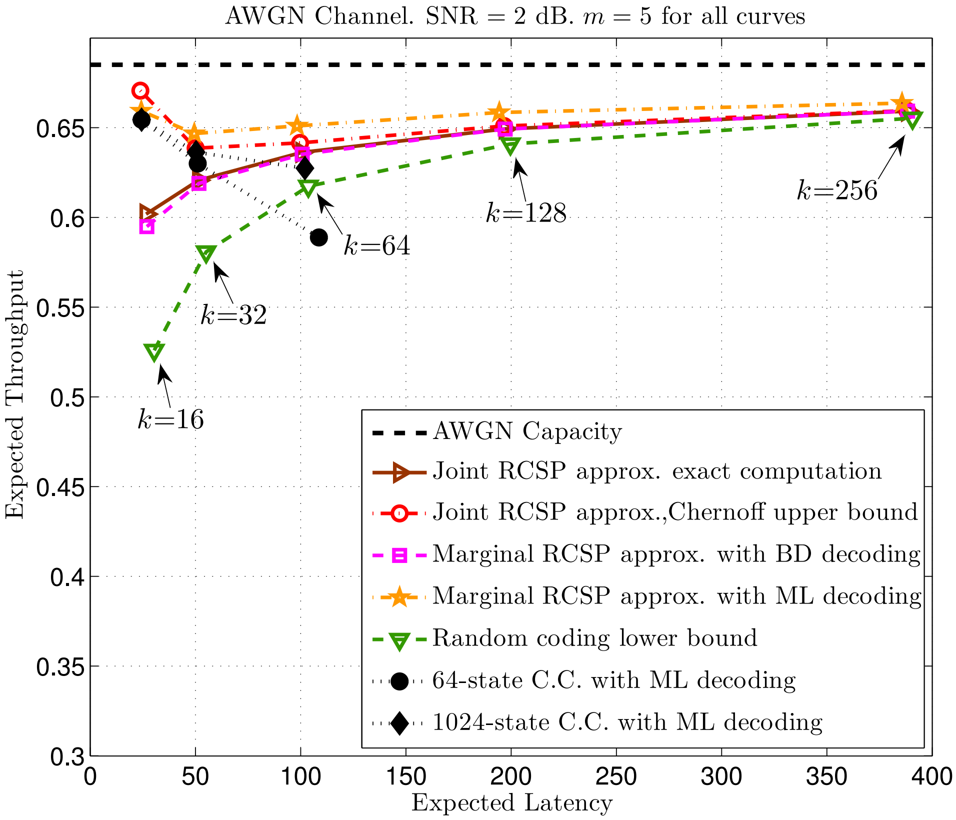

Figure 2.10: The R_t vs. ℓ curves for the joint RCSP approximation with $m = 5$ over the 2dB AWGN channel. All curves use the optimized increments in Table 2.2.

asymptotic capacity is 0.6851 bits. The $m = 5$ curves shown include exact numerical integration of the joint RCSP approximation, the upper bound on the joint RCSP approximation using (2.44) and (2.47), the marginal RCSP approximation using both ML decoding and BD decoding, and the random-coding lower bound using (2.8). Evaluating (2.6) would give a slightly better bound than (2.8) for random coding but is very time-consuming to compute.

The throughput upper bound using (2.44) and (2.47) becomes tight for latencies larger than 100. The lower bound on the joint RCSP approximation using (2.45) and (2.46) is not shown separately because it turns out to be identical to the marginal RCSP approximation. This is because the Chernoff bound of the pairwise joint probabilities are often larger than the marginal probabilities.

Fig. 2.10 also plots the R_t vs. ℓ points of the 1024-state and 64-state convolutional codes presented in Sec. 2.3.2. The simulation results demonstrate that both codes achieve throughputs higher than the random-coding lower bound for $k = 16$ and $k = 32$. The more complex 1024-state code gives expected throughput higher than random coding even for $k = 64$. For $k = 16$ and $k = 32$ the simulation points of the 1024-state code closely approach the marginal RCSP approximation of the expected throughput using ML decoding.

Note that the lower bound of the expected throughput using random coding is significantly below the RCSP approximations (joint RCSP approximation using BD decoding and marginal RCSP using ML or BD decoding) for low expected latencies. However, for expected latencies above 350 symbols the RCSP approximations and the lower bound based on random coding produce very similar expected throughputs.

For random coding, an i.i.d. codebook is drawn using a Gaussian distribution with a zero mean and a variance equal to the power constraint η. This type of random codebook generation will sometimes produce a codeword that violates the power constraint. To address this, the average power should be slightly reduced or codewords violating the power constraint should be purged, either of which will lead to a slight performance degradation.

To conclude this subsection, we summarize two relevant observations to motivate the next subsection: (1) When using BD decoding, the difference in expected throughput between the joint RCSP approximation and the marginal RCSP approximation is negligible. (2) The difference between the joint RCSP approximation using BD decoding and marginal RCSP approximation using ML decoding is small for reasonably large expected latencies, e.g., 200 for $m > 1$. These two observations allow us to focus on efficient optimizations based on marginal RCSP approximation using BD decoding.

2.5.4 RCSP with Uniform Increments

For a specified k and for a fixed finite number of transmissions m, there are m variables $\{I_j\}_{j=1}^{m}$ that can be varied to optimize the throughput. The number of possible combinations of I_j's

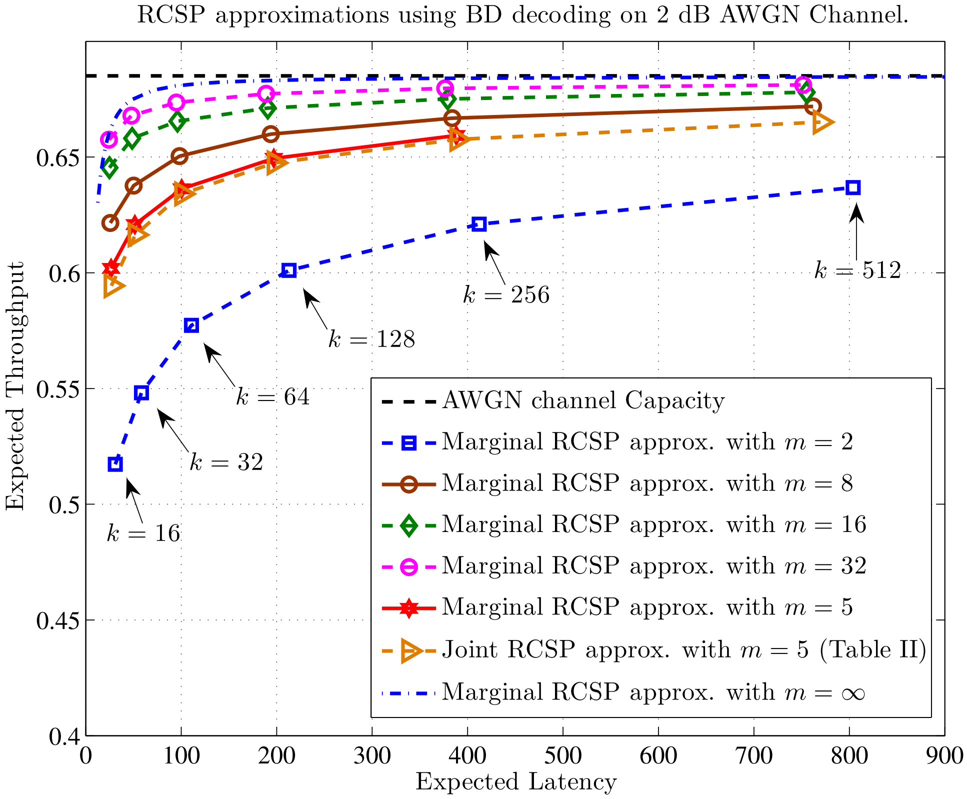

Figure 2.11: Comparing marginal RCSP approximation with optimized uniform increment $I_j = I$, $m = 2, 8, 16, 32, \infty$, and joint RCSP approximation with optimized $\{I_j\}_{j=1}^{m}$, $m = 5$.

increases rapidly as m increases. Sec. 2.5.2 addressed the optimization problem for $m \leq 6$ by using the joint RCSP approximation. Motivated by the pattern seen in Table 2.2, this subsection considers the large m case by restricting the transmissions to use uniform increments $I_j = I$ for $j > 1$. This yields a two parameter optimization: the initial blocklength n_1 and the increment I. To reflect practical constraints, we restrict the increment I to be an integer.

We also reduce computational burden by replacing the joint RCSP approximation with the marginal RCSP approximation, and we only use BD decoding. Note that in Sec. 2.5.3 we saw that the marginal RCSP approximation is operationally identical to lower bound on the joint RCSP approximation using (2.45) and (2.46) and is a tight lower bound to the joint RCSP approximation. It is also relatively simple to compute using BD decoding since the relevant probability of error is simply the tail of a chi-square.

Table 2.3: Optimized RCSP n_1 and n_m for $m = 5$ and uniform increments.

k	n_1	n_5	I	R_t
16	17	47	6	0.5944
32	39	79	8	0.6164
64	83	143	12	0.6341
128	172	262	18	0.6475
256	353	483	26	0.6576

Fig. 2.11 presents the optimized performance with uniform increments for various m ranging from 2 to ∞. In the optimization that produced this figure, the longest possible blocklength was constrained to be less than $\lceil 6k/C \rceil$ where C is the capacity of the AWGN channel.

For the $m = 5$ case, it is instructive to compare optimized uniform increments with the unconstrained optimal increments of Table 2.2. Table 2.3 shows the numerical results of the uniform-increment optimization for $m = 5$ on the 2 dB AWGN channel. Comparing the $m = 5$ curves in Fig. 2.11 and the parameters n_1, n_m, I, and R_t in Tables 2.2 and 2.3 shows that the constraint of constant increments $I_j = I$ for $j > 1$ negligibly reduces expected throughput in this case.

Looking at the uniform-increment curves in Fig. 2.11, we observe diminishing returns even for m increasing exponentially. This implies that for a practical system it suffices to consider an m smaller than 16.

2.5.5 Performance across a range of SNRs

To allow easy comparison across the various plots above, we have focused attention on the specific case of the 2 dB AWGN channel. The uniform-increment approach of the previous subsection allows us to efficiently explore performance across a range of SNRs. For expected latencies constrained to be close to (but not greater than) 200 symbols, Fig. 2.12 plots the marginal RCSP approximation of R_t vs. η for $m = 1, 4, 8$ and η ranging from 0 to 5 dB. The expected throughput

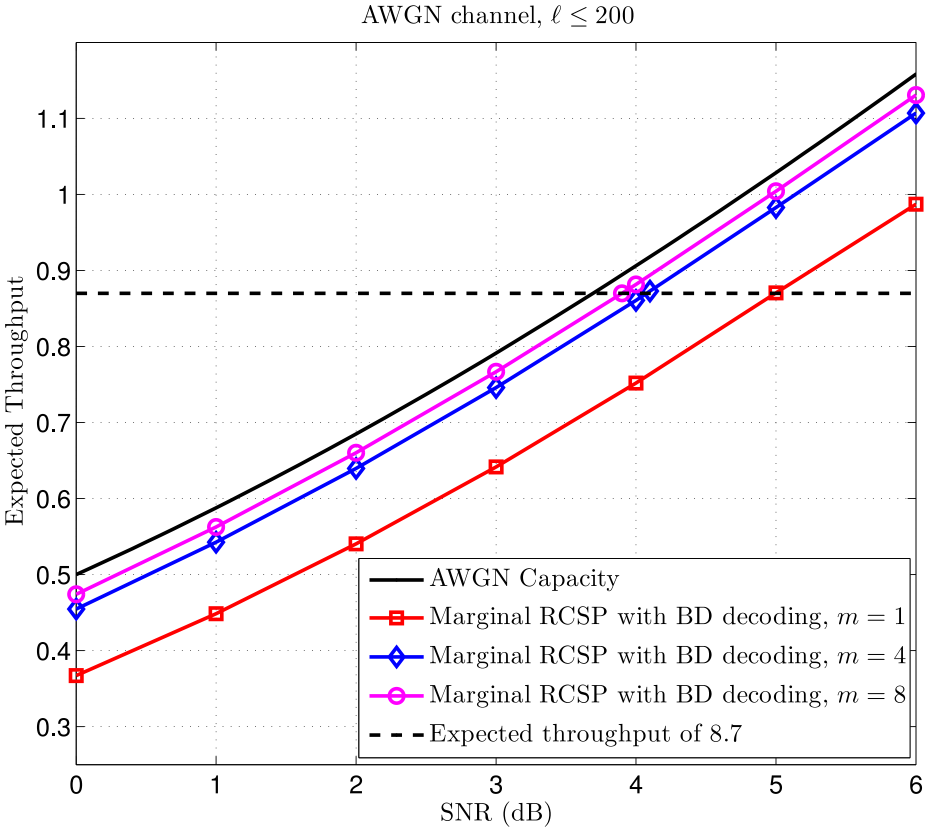

Figure 2.12: R_t vs. η for $\ell \leq 200$, $\eta = 0, 1, \ldots, 5$dB and $m = 1, 4, 8$.

R_t is obtained by finding the largest integer k such that the optimized initial blocklength n_1 and the uniform increment I yield expected latency $\ell \leq 200$. The actual expected latencies ranged only between 197 and 200. We chose the constraint to be 200 since the difference between ML decoding and BD decoding for the marginal RCSP approximation is small.

This plot shows the significant benefit of even limited IR as compared to ARQ over a range of SNRs. For example, at 4 dB the curve for $m = 1$ (ARQ) is 0.155 bits from the original AWGN capacity, but this gap reduces to 0.046 bits for $m = 4$ and 0.025 bits for $m = 8$. To see these gaps from an SNR perspective, the horizontal line at expected throughput 8.7 bits shows that $m = 1$ (ARQ) performs within 1.3 dB of the original AWGN capacity while $m = 4$ is within 0.4 dB and $m = 8$ is within 0.2 dB.

Recall that the marginal RCSP curves in Fig. 2.12 are for repeated IR-NTC, which generally can have throughputs above capacity because of the extra information communicated by the NTC. We saw this in Figs. 2.5 and 2.6. However, this extra information is quite limited for small values of m. A simple upper bound on the extra information per transmitted symbol to communicate τ for repeated IR-NTC is $\log_2(m+1)/n_1$, where n_1 is the initial blocklength. We examine this upper bound for the case of 4 dB. For $m=1$, n_1 is 192 symbols and the upper bound is 0.0052 bits. For $m=4$ and $m=8$ the values of n_1 are 182 and 178 respectively, and the upper bounds on the NTC per-symbol extra information are 0.0127 and 0.0177 bits respectively.

Thus, the small values of m along with practically reasonable expected latencies of around 200 symbols considered in Fig. 2.12 cause the extra information provided by NTC to be negligible. Note that for larger expected latencies, the per-symbol extra information of NTC will become even smaller.

2.5.6 Optimizing Increments for Non-Repeating IR-NTC

Repeated IR-NTC has an outage probability of zero because it never stops trying until a message is decoded correctly. However, this leads to an unbounded maximum latency. Using a non-repeating IR-NTC scheme, optimization of transmission lengths using the joint RCSP approximation can incorporate a strict constraint on the number of incremental transmissions so that the transmitter gives up after m transmissions. This optimization can also include a constraint on the outage probability, which is nonzero for non-repeating IR-NTC.

To handle these two new constraints, we fix m and restrict $\mathbb{P}[E_{n_m}]$ to be less than a specified p_{outage}. Without modifying the computations of $\mathbb{P}[E_{n_j}]$, the optimization is adapted to pick the set of lengths that yields the maximum throughput s.t. $\mathbb{P}[E_{n_m}] \leq p_{\text{outage}}$. When there is a decoding error after the mth transmission, the transmitter declares an outage event and proceeds to encode the next k information bits. This scheme is suitable for delay-sensitive communications, in which data packets are not useful to the receiver after a fixed number of transmission attempts.

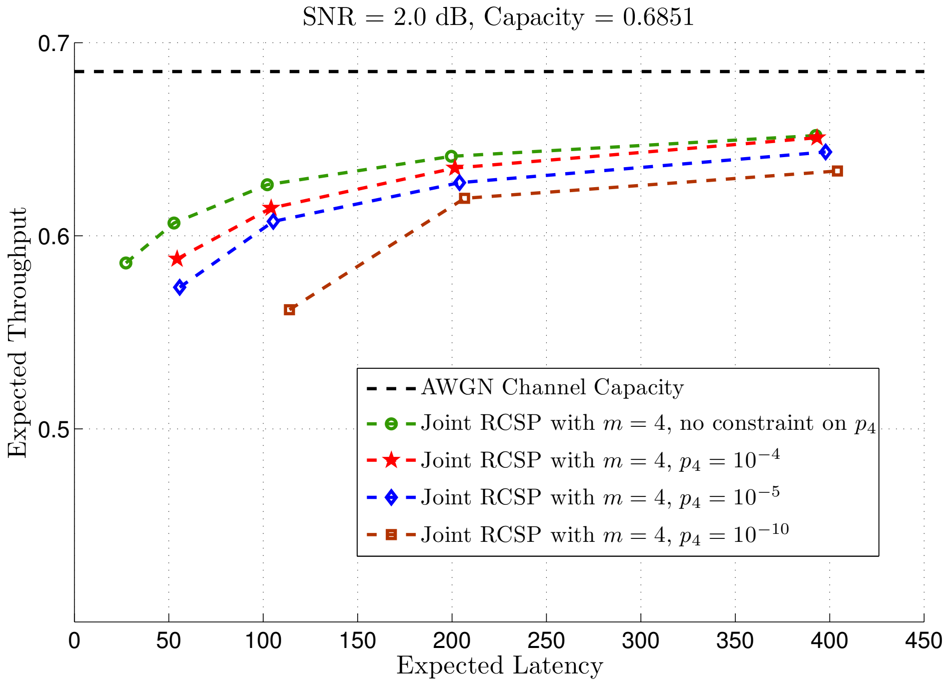

Figure 2.13: The effect of specifying a constraint on the outage probability p_4 on the R_t vs. ℓ.

The expected number of channel uses ℓ is given by

$$\ell = I_1 + \sum_{j=2}^{m} I_j \mathbb{P}[E_{n_{j-1}}]. \tag{2.48}$$

The expected throughput R_t is again given by k/ℓ and the outage probability is $\mathbb{P}[E_{n_m}]$.

Fig. 2.13 shows how the outage probability constraint affects the R_t vs. ℓ curve. The maximum number of transmissions is fixed to be $m = 4$. The constraint values we considered for the outage probability (error probability of the fourth transmission $\mathbb{P}[E_{n_4}] = p_4$) are 1 (no constraint), $10^{-4}, 10^{-5}$ and 10^{-10}. Stricter constraints on outage probability increase the average latency ℓ. According to the joint RCSP approximation, however, it is exciting to see that the loss in the expected throughput is only 0.022 bits around latency of 200 symbols compared to the unconstrained R_t even with the outage constraint $p_4 = 10^{-10}$.

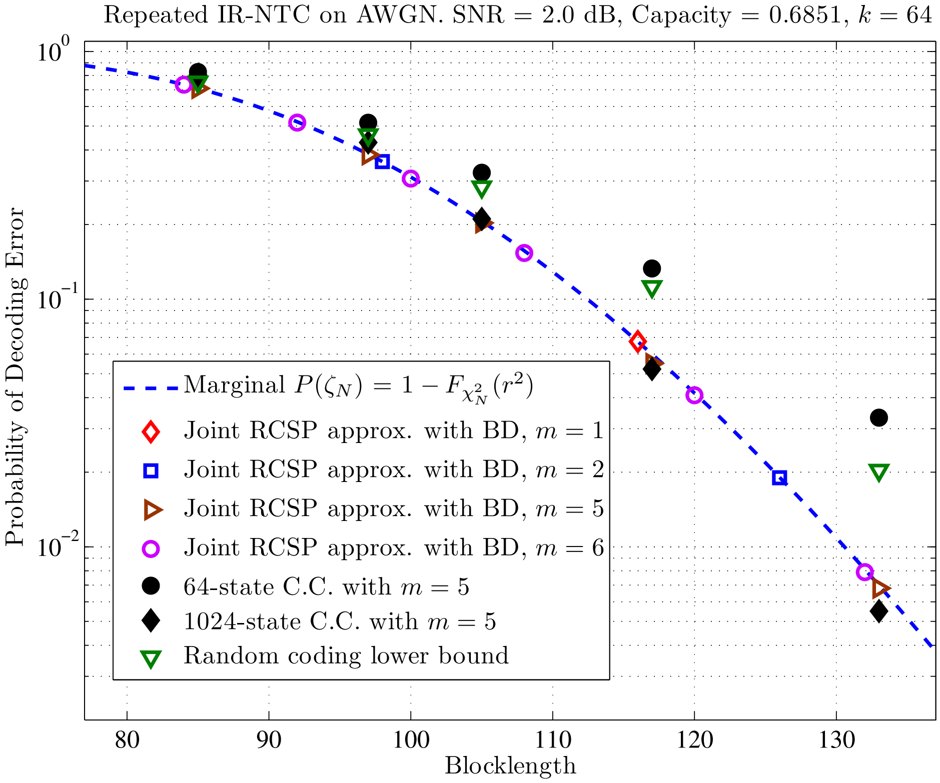

Figure 2.14: A comparison of the decoding error trajectories of joint RCSP approximation, marginal RCSP approximation, simulated ML-decoded convolutional codes and random-coding lower bound for $k = 64$.

2.5.7 Decoding Error Trajectory with BD Decoding

The optimization of increments in Sec. 2.5.2 uses the joint RCSP approximation[8] to find the highest expected throughput R_t. The joint RCSP approximation provides a set of joint decoding error probabilities $\mathbb{P}[E_{n_j}], j = 1, \ldots, m$, which we call the "decoding error trajectory". If we can find a family of rate-compatible codes that achieves this decoding error trajectory, then we can match the throughput performance suggested by the joint RCSP approximation. For the short-latency regime, e.g. $k = 16$ and $k = 32$, one should use the marginal RCSP approximation with ML decoding to study the decoding error trajectory for a better approximation. To demonstrate

[8] Note that for joint RCSP approximation we only use BD decoding.

an example of the decoding error trajectory we focus our attention on the case of $k = 64$ and use BD decoding throughout this subsection.

Fig. 2.14 presents the decoding error trajectories for $k = 64$ and $m = 1, 2, 5, 6$ using the joint RCSP approximation. Each trajectory corresponds to a $k = 64$ point on the R_t vs. ℓ curve in Fig. 2.9. For example, the decoding error trajectory for $k = 64$ and $m = 2$ consists of the two blue square markers in Fig. 2.14 and corresponds to the point on the blue solid curve in Fig. 2.9 with $k = 64, m = 2$.

Fig. 2.14 also shows the decoding error trajectories for the random-coding lower bound using (2.8) with $m = 5$, as well as the simulations of the two tail-biting convolutional codes presented in Sec. 2.3.2 with $m = 5$. The dashed line is the decoding error trajectory using the marginal RCSP approximation. The marginal RCSP approximation provides a good estimate that can serve as a performance goal for practical rate-compatible code design across a wide range of blocklengths.

While the 64-state code is not powerful enough to match the trajectory suggested by the joint RCSP approximation, the 1024-state code closely follows the trajectory for $m = 5$ and therefore has a performance very close to the joint RCSP (c.f. Fig. 2.10). Thus there exist practical codes, at least in some cases, that achieve the idealized performance of RCSP.

Fig. 2.15 shows how the outage probability constraints affect the decoding error trajectory for $k = 128$ and $m = 4$ using the joint RCSP approximation. Curves are shown with no constraint on the outage probability p_4 and for p_4 constrained to be less than $10^{-4}, 10^{-5}$ and 10^{-10}. For ease of comparison, the x-axis is labeled with the transmission index rather than blocklength as in Fig. 2.14.

At each index, the blocklengths corresponding to the curves with different constraints are different. For example, at transmission index two, the curve with $p_4 = 10^{-4}$ has blocklength 191 whereas the curve with $p_4 = 10^{-10}$ has blocklength 211. An important observation is that even for relatively low outage probability constraints such as $p_4 = 10^{-10}$, the initial transmission should still have a relatively high decoding error rate in order to take advantage of instantaneous

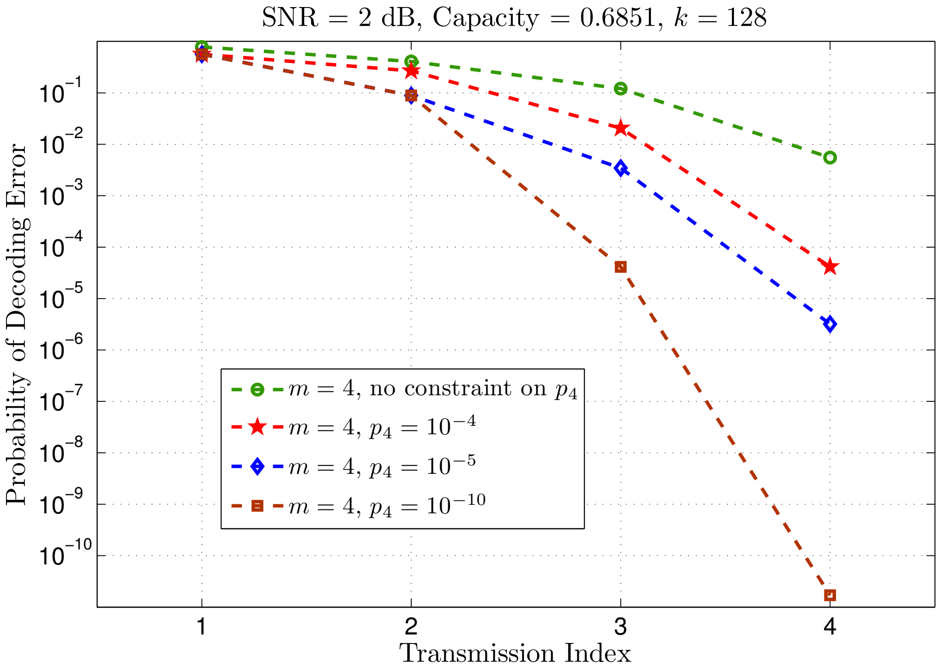

Figure 2.15: The effect of specifying a constraint on the outage probability $\mathbb{P}[E_4] = p_4$ on the decoding error trajectory for $k = 128$.

information densities that may be significantly higher than capacity.

2.6 Concluding Remarks

Inspired by the achievability and converse results in [PPV11] and practical simulation results in [CSS10], this chapter studies feedback communication systems that use incremental redundancy. We focus on the convenient model of IR-NTC, in which a stream of incremental redundancy concludes when a noiseless confirmation symbol is sent by the transmitter once the receiver has successfully decoded.

VLFT achievability in [PPV11] uses a non-repeating IR-NTC system with an infinite-length mother code and decoding attempted after each received symbol. The first part of this chapter shows that a finite-length mother code (implying repeated IR-NTC to achieve zero-error communication) with decoding attempted only at certain specified times can still approach the VLFT achievability curve. The finite-length constraint introduces only a slight penalty in expected latency as long as the additional length of the mother code beyond the blocklength corresponding to capacity grows logarithmically with the expected latency. This is a requirement that is easily met by practical systems. In contrast, the expected latency penalty associated with decoding time limitations is linear in the interval between decoding times. This forces the intervals to grow sub-linearly in the expected latency for systems to approach capacity.

The second part of this chapter introduces rate-compatible sphere-packing (RCSP) and uses this tool to analyze and optimize IR-NTC systems for the AWGN channel. The joint RCSP approximation with BD decoding optimizes the incremental lengths $I_1, \dots, I_m$ for small values of m in a repeated IR-NTC system. We found that under BD decoding, the marginal RCSP approximation is a tight lower bound of the joint RCSP approximation of expected throughput and simplifies the computation. This simplification allows optimization of the uniform incremental length I for repeated IR-NTC with larger values of m. The marginal RCSP approximation can also be computed for ML decoding, and the difference between ML decoding and BD decoding is significant for short expected latencies. For expected latencies larger than 200 symbols, however, we observed that the difference between ML and BD decoding becomes small.

For relatively small values of m and N, a repeated IR-NTC system can approach the capacity of the original AWGN channel with expected latencies around 200 symbols. We applied the marginal RCSP approximation assuming BD decoding across a range of SNRs to an IR-NTC system with $m = 8$ and expected latencies at or below 200 symbols. The results showed throughputs consistently within about 0.2 dB of the performance corresponding to the original AWGN capacity. The NTC introduces additional information that can generally cause IR-NTC achievable rates to be above capacity. When m is less than 8 and the expected latency is above 200 symbols, however, this increase in throughput is limited to negligible values (less than 0.02 bits) so that comparisons with the original AWGN channel capacity are reasonable.

For non-repeating IR-NTC, we can use the joint RCSP approximation with BD decoding to optimize the incremental lengths $I_1, \dots, I_m$ under an outage constraint. Numerical result shows that for an expected latency above 200 symbols, strict outage probability constraints can be met with minimal loss in throughput.

From a practical code design perspective, this chapter demonstrates an IR-NTC system for $m = 5$ incremental transmissions based on a 1024-state, randomly punctured, tail-biting convolutional code with optimized transmission increments. At short expected latencies, the resulting IR-NTC system exceeds the random-coding lower bound of [PPV11] and closely matches the throughput-latency performance predicted by RCSP for the AWGN channel at low latency.

Rate-compatible codes for IR-NTC systems that match the performance predicted by RCSP remain to be identified for expected latencies between 200 and 600 symbols. This chapter demonstrates that the decoding error trajectory based on the marginal RCSP approximation can provide the target error probabilities for designing such rate-compatible codes. Approximations based on both ML decoding and BD decoding can be used. BD decoding is easier to compute and we showed that the difference between ML and BD decoding becomes small for $m > 1$ and expected latencies larger than 200. The design of rate-compatible codes matching the marginal-RCSP decoding error trajectory for expected latencies between 200 and 600 symbols is a challenging open problem in channel code design.

2.A Proofs for VLFT with Practical Constraints

Proof of Thm. 2. Consider a random codebook $\mathcal{C}_N = \{\mathsf{C}_1, \ldots, \mathsf{C}_M\}$ with M codewords of length-N and codeword symbols independent and identically distributed according to P_X. To construct a VLFT code consider the following (U, f_n, g_n, τ): The common random variable

$$U \in \mathcal{U} = \overbrace{\mathcal{X}^N \times \cdots \times \mathcal{X}^N}^{M \text{times}} . \tag{2.49}$$

is distributed as:

$$U \sim \prod_{j=1}^{M} P_{X^N} . \tag{2.50}$$

A realization of U corresponds to a deterministic codebook $\{\mathsf{c}_1, \ldots, \mathsf{c}_M\}$. Let $\mathsf{C}_W(n)$ denote the nth symbol of the codeword C_W, and let $[\mathsf{C}_j]^n$ denote the first n symbols of the codeword C_j. The sequence (f_n, g_n) is defined as

$$f_n(U, W) = \mathsf{C}_W(n) \tag{2.51}$$

$$g_n(U, Y^n) = \arg \max_{j=1,\ldots,M} i([\mathsf{C}_j]^n; Y^n), \tag{2.52}$$

and the stopping time τ is defined as:

$$\tau = \inf\{n : g_n(U, Y^n) = W\} \wedge N\,. \tag{2.53}$$

The nth marginal error event ζ_n is given as:

$$\zeta_n = \left\{ \bigcup_{j \neq W} i(\mathsf{C}_j^n; Y^n) > i(\mathsf{C}_W^n; Y^n) \right\}. \tag{2.54}$$

Following (2.11)-(2.14) we have

$$\begin{aligned} \mathbb{E}[\tau] &= \sum_{n=0}^{N-1} \mathbb{P}[\tau > n] & (2.55)\\ &\leq \sum_{n=0}^{N-1} \mathbb{P}[\zeta_n]. & (2.56) \end{aligned}$$

As in [PPV11, (151)-(153)], the union bound $\mathbb{P}(\zeta_n) \leq \xi_n$ provides an upper bound on (2.14) as follows:

$$\mathbb{E}[\tau] \leq \sum_{n=0}^{N-1} \xi_n\,, \tag{2.57}$$

where ξ_n is given in (2.7).

With a similar bounding technique, the error probability can be upper bounded as:

$$\begin{aligned} \mathbb{P}[g_\tau(U, Y^\tau) \neq W] &= \mathbb{P}[g_N(U, Y^N) \neq W, \tau = N] & (2.58)\\ &= \mathbb{P}\left[\bigcap_{j=1}^{N} \zeta_j\right] & (2.59)\\ &\leq \mathbb{P}[\zeta_N] & (2.60) \end{aligned}$$

$$\leq \xi_N \,. \tag{2.61}$$

In other words, the error probability is upper bounded by the error probability of the underlying finite-length code $\mathcal{C}_N$. □

Proof of Thm. 3. The proof follows from random coding and the following modification of the triplet (f_n, g_n, τ) of Thm. 2: For $k = 1, 2, \dots$ let (f'_n, g'_n) be defined as:

$$f'_n(U, W) = \begin{cases} f_n(U, W) & \text{if } n \leq N \\ f_{n-kN}(U, W) & \text{if } kN < n \leq (k+1)N \end{cases}$$

$$g'_n(U, Y^n) = \begin{cases} g_n(U, Y^n) & \text{if } n \leq N \\ g_{n-kN}(U, Y^n_{kN+1}) & \text{if } kN < n \leq (k+1)N \end{cases}$$

Let the new stopping time τ' be defined as:

$$\tau' = \inf\{n : g'_n(U, Y^n) = W\} \,. \tag{2.62}$$

The error probability is zero because the definition of the stopping time τ' ensures that decoding stops only when the decision is correct. As mentioned above, the new encoder/decoder sequence (f'_n, g'_n) is simply an extension of the VLFT code in Thm. 2 by performing an ARQ-like repetition. The expectation of τ' is thus given as:

$$\mathbb{E}[\tau'] = \sum_{n=0}^{N-1} \mathbb{P}\left[\bigcap_{j=1}^{n} \zeta_j\right] + \mathbb{P}\left[\bigcap_{j=1}^{N} \zeta_j\right] \mathbb{E}[\tau'] \tag{2.63}$$

$$\leq \sum_{n=0}^{N-1} \mathbb{P}[\zeta_n] + \mathbb{P}[\zeta_N]\mathbb{E}[\tau'] \,, \tag{2.64}$$

which implies that:

$$\mathbb{E}[\tau'] \leq (1 - \mathbb{P}[\zeta_N])^{-1} \sum_{n=0}^{N-1} \mathbb{P}[\zeta_n] \,. \tag{2.65}$$

Applying the RCU bound to replace each $\mathbb{P}[\zeta_n]$ with ξ_n completes the proof. □

Proof of Thm. 4. We define a pair of random walks to simplify the proofs:

$$S_n \triangleq i(X^n; Y^n) \tag{2.66}$$

$$\bar{S}_n \triangleq i(\bar{X}^n; Y^n)\,. \tag{2.67}$$

Referring to (2.2), note that for any measurable function f we have the property:

$$\mathbb{E}[f(\bar{X}^n, Y^n)] = \mathbb{E}[f(X^n, Y^n)\exp\{-S_n\}]. \tag{2.68}$$

Letting P_X to be a capacity-achieving input distribution, observe that S_n and $\bar{S}_n$ are sums of i.i.d. r.v.s with positive and negative means:

$$\mathbb{E}[i(X;Y)] = C \tag{2.69}$$

$$\mathbb{E}[i(\bar{X};Y)] = -L\,, \tag{2.70}$$

where C is the channel capacity and L is the lautum information [PV08]. The sequence $\{S_n - nC\}_n$ is a bounded martingale based on our assumption that the information density of each symbol is essentially bounded. Hence, by Doob's optional stopping theorem we have for a stopping time τ:

$$\mathbb{E}[S_\tau] = C\mathbb{E}[\tau]\,. \tag{2.71}$$

Properties (2.68)-(2.71) are used in the rest of this appendix.

Using the definitions of (2.66) and (2.67) in (2.7) produces (2.72). Weakening the RCU bound using (2.8) and replacing $M-1$ with M in (2.8) for simplicity produces (2.73):

$$\xi_n = \mathbb{E}\left[\min\left\{(1, (M-1)\mathbb{P}[\bar{S}_n \geq S_n | X^n Y^n]\right\}\right] \tag{2.72}$$

$$\leq \mathbb{E}\left[\exp\{-[S_n - \log M]^+\}\right]\,. \tag{2.73}$$

Applying (2.73) to Thm. 3 yields the following:

$$\ell \leq \frac{1}{(1-\xi_N)} \sum_{n=0}^{N-1} \mathbb{E}\left[\exp\left\{-[S_n - \log M]^+\right\}\right]\,. \tag{2.74}$$

Consider an auxiliary stopping time $\tilde{\tau}$ w.r.t. the filtration $\mathcal{F}_n = \sigma\{X^n, \bar{X}^n, Y^n\}$:

$$\tilde{\tau} = \inf\{n \geq 0 : S_n \geq \log M\} \wedge N \, . \tag{2.75}$$

For a specified set E, use $\mathbb{E}[X; E]$ to denote $\mathbb{E}[X 1_E]$ where 1_E is the indicator function of the set E. We now turn our attention to computing the summation in (2.74). Letting E be the set $\{\tilde{\tau} < N\}$, we have the following:

$$\sum_{n=0}^{N-1} \mathbb{E}\left[\exp\{-[S_n - \log M]^+\}\right] = \mathbb{E}\left[\tilde{\tau} - 1 + \sum_{k=0}^{N-1-\tilde{\tau}} \exp\left\{-[S_{\tilde{\tau}+k} - \log M]^+\right\}; E\right] + N\mathbb{P}[E^c] \, . \tag{2.76}$$

On E we have $i(X^{\tilde{\tau}}; Y^{\tilde{\tau}}) \geq \log M$ and hence:

$$\begin{aligned} [S_{\tilde{\tau}+k} - \log M]^+ &= [S_{\tilde{\tau}+k} - S_{\tilde{\tau}} + S_{\tilde{\tau}} - \log M]^+ && (2.77) \\ &\geq [S_{\tilde{\tau}+k} - S_{\tilde{\tau}}]^+ && (2.78) \\ &\overset{d}{=} [S_k]^+ && (2.79) \end{aligned}$$

where the last equality is equality in distribution and is true almost surely by the strong Markov property of random walks. Applying (2.77)-(2.79) to (2.76) yields:

$$\sum_{n=0}^{N-1} \mathbb{E}\left[\exp\{-[S_n - \log M]^+\}\right] \leq \mathbb{E}\left[\tilde{\tau} - 1 + \sum_{k=0}^{N-1-\tilde{\tau}} \exp\{-[S_k]^+\}; E\right] + N\mathbb{P}[E^c] \, . \tag{2.80}$$

Using (2.68) we have the following:

$$\mathbb{E}\left[\exp\{-[S_k]^+\}\right] = \mathbb{P}\left[\bar{S}_k > 0\right] + \mathbb{P}\left[S_k \leq 0\right] . \tag{2.81}$$

S_k and $\bar{S}_k$ are sums of i.i.d. r.v.s with positive and negative means, respectively. Thus by the Chernoff inequality, both terms decay exponentially in k, yielding

$$\mathbb{P}\left[\bar{S}_k > 0\right] + \mathbb{P}\left[S_k \leq 0\right] \leq a_1 e^{-k a_2} \, , \tag{2.82}$$

for some positive constants a_1 and a_2. Thus there is a constant $a_3 > 0$ such that:

$$\sum_{k=0}^{N-1-\tilde{\tau}} \mathbb{E}\left[\exp\{-[S_k]^+\}\right] \leq \sum_{k=0}^{N-1} \mathbb{E}\left[\exp\{-[S_k]^+\}\right] \tag{2.83}$$

$$
\begin{aligned}
&\leq \sum_{k=0}^{N-1} a_1 e^{-ka_2} && (2.84)\\
&= \frac{a_1 e^{-a_2}(1 - e^{-(N-1)a_2})}{1 - e^{-a_2}} && (2.85)\\
&= a_3 \,. && (2.86)
\end{aligned}
$$

We assume that S_n has bounded jumps, and hence on the set E there is a constant a_4 such that

$$S_{\tilde{\tau}} - \log M \leq a_4 C \,. \tag{2.87}$$

Therefore from (2.71) we have that on the set E:

$$\mathbb{E}[\tilde{\tau}] \leq \frac{\log M}{C} + a_4 \,. \tag{2.88}$$

We are now ready to provide a bound on (2.74). Letting $a_5 = a_3 + a_4$ and applying (2.83)-(2.86) and (2.88) to (2.76) we have:

$$\ell \leq (1 - \mathbb{P}[\zeta_N])^{-1} \left(\frac{\log M}{C} + a_5 + N\mathbb{P}[E^c] \right) \,. \tag{2.89}$$

For a fixed M with random coding, there is a constant $\Delta > 0$ such that with $C_\Delta = C - \Delta$ and $N = \log M / C_\Delta$ we have the following bound on error probability:

$$\mathbb{P}[\zeta_N] \leq b_2 \exp(-N b_3) \,, \tag{2.90}$$

for some constants $b_2 > 0$ and $b_3 > 0$.

Recalling that S_n is a sum of i.i.d. r.v.'s with mean C, let $N = \log M / C_\Delta$ we have by the Chernoff inequality that:

$$
\begin{aligned}
\mathbb{P}[E^c] &= \mathbb{P}[S_N < \log M] && (2.91)\\
&\leq b_0 \exp\left\{ -b_1 N \left(C - \frac{\log M}{N} \right) \right\} && (2.92)\\
&= b_0 \exp\left\{ -b_1 \frac{\log M}{C_\Delta} \Delta \right\} \,. && (2.93)
\end{aligned}
$$

Combining (2.89) and (2.93) we have the following for ℓ:

$$\ell \leq (1 - \mathbb{P}[\zeta_N])^{-1} \left(\frac{\log M}{C} + a_5 + \frac{\log M}{C_\Delta} \frac{b_0}{M^{b_1 \Delta / C_\Delta}} \right) \tag{2.94}$$

Now applying (2.90), notice that we are only interested in the first two terms of the expansion $(1-\mathbb{P}[\zeta_N])^{-1} = 1 + \mathbb{P}[\zeta_N] + \mathbb{P}[\zeta_N]^2 + \dots$ on $[0,1)$. Thus

$$\ell \leq \frac{\log M}{C} + \frac{b_0 \log M}{C_\Delta M^{b_1 \Delta / C_\Delta}} + \frac{b_2 \log M}{C(M^{b_3/C_\Delta})} + a_6 \tag{2.95}$$

for some $a_6 > 0$. Hence for M large enough we have (2.19). □

Proof of Cor. 1. We first choose N to scale with ℓ with a factor δ to be chosen later:

$$N = (1+\delta)\ell\,. \tag{2.96}$$

Then by the converse (Thm. 5) we have:

$$\frac{\log M}{N} \leq C + \frac{\log(\ell+1) + \log e - \delta\ell C}{(1+\delta)\ell} \tag{2.97}$$

$$\leq C - \delta'\,. \tag{2.98}$$

The term δ' on the right is positive by setting:

$$\delta > \frac{\log(\ell+1) + \log e}{\ell C}\,. \tag{2.99}$$

Again by the Chernoff inequality we have:

$$\mathbb{P}[\tau' \geq N] = \mathbb{P}[S_N < \log M] \tag{2.100}$$

$$\leq \mathbb{P}[S_N - NC < -N\delta'] \tag{2.101}$$

$$\leq b_0' \exp\left\{-\ell(1+\delta)b_1'\delta'\right\}\,. \tag{2.102}$$

Since δ is chosen such that $\log M/N$ is less than capacity, we also have (2.90). By reordering (2.89) we have for some $b_2', b_3' > 0$ such that:

$$\frac{\log M}{C} \geq \ell\left[1 - b_2 e^{-\ell(1+\delta)b_3} - (1+\delta)b_0' e^{-b_2'\ell}\right] - b_3'\,, \tag{2.103}$$

which implies $\log M_t^*(\ell, N) \geq \ell C - O(1)$ for $N = (1+\delta)\ell$ and large enough ℓ. □

Proof of Thm. 6. Consider the same random-coding scheme as in Thm. 3 with encoders $f'_n(U, W)$ and decoders $g'_n(U, Y^n)$. The auxiliary stopping time $\tilde{\tau}$ of Thm. 4 is altered to reflect the limitation on decoding times as follows:

$$\tilde{\tau} = n_1 + (\tilde{j} - 1)I, \tag{2.104}$$

where $\tilde{j}$ is also a stopping time given as:

$$\tilde{j} = \inf\{j > 0 : S_{n_j} = i(X^{n_j}; Y^{n_j}) \geq \log M\}\,. \tag{2.105}$$

The rest is similar to the proof of Thm. 4, but without the complication of a finite N:

$$\begin{aligned}
\ell &\leq n_1 + I \sum_{j=1}^{\infty} \mathbb{P}[\zeta_{n_j}] && (2.106)\\
&\leq n_1 + I \sum_{j=1}^{\infty} \mathbb{E}\left[\exp\left\{-\left[S_{n_j} - \log M\right]^+\right\}\right] && (2.107)\\
&\leq n_1 + I\mathbb{E}[\tilde{j} - 1] + I \sum_{k=0}^{\infty} \mathbb{E}\left[\exp\left\{-\left[S_{n_{\tilde{j}+k}} - \log M\right]^+\right\}\right] && (2.108)\\
&\leq \mathbb{E}[\tilde{\tau}] + I \sum_{k=0}^{\infty} \mathbb{E}\left[\exp\left\{-[S_{n_k}]^+\right\}\right] && (2.109)\\
&\leq \mathbb{E}[\tilde{\tau}] + Ia_3 && (2.110)\\
&\leq \frac{\log M}{C} + Ia_4 + Ia_3\,, && (2.111)
\end{aligned}$$

for some $a_3 > 0$ and some $a_4 > 0$, where (2.110) follows by applying the Chernoff inequality and (2.111) is a consequence of the jumps S_{n_k} being bounded by Ia_4C for some $a_4 > 0$. Reordering the equations gives the result. □

Proof of Lem. 1. Consider the same random-coding scheme as in Thm. 6 with encoders $f'_n(U, W)$ and decoders $g'_n(U, Y^n)$, an initial blocklength n_1, and a uniform increment I. However, the number of increments is now limited to a finite integer m. The finite block-length is then given by $N = n_m$ where $n_j = n_1 + (j - 1)I$. Similar to Thm. 4, define the auxiliary stopping time as:

$$\tilde{j} = \inf\{j > 0 : S_{n_j} \geq \log M\} \wedge m\,. \tag{2.112}$$

Similar to Thm. 4 we have (2.113) to (2.117), shown as follows:

$$
\begin{aligned}
(1-\xi_N)\ell &\le n_1 + I\sum_{j=1}^{m-1}\mathbb{E}[\exp\{-[S_{n_j}-\log M]^+\}] && (2.113)\\
&= \mathbb{E}\left[n_1 + (\tilde{j}-1)I + I\sum_{k=0}^{m-1-\tilde{j}}\exp\left\{-\left[S_{n_{k+\tilde{j}}}-\log M\right]^+\right\};E\right] && (2.114)\\
&\quad + \mathbb{E}[n_1+(m-1)I;E^c]\\
&\le \mathbb{E}[\tilde{\tau};E] + I\mathbb{E}\left[\sum_{k=0}^{m-1-\tilde{j}}\exp\left\{-\left[S_{n_{k+\tilde{j}}}-\log M\right]^+\right\};E\right] + N\mathbb{P}[E^c] && (2.115)\\
&\le \mathbb{E}[\tilde{\tau};E] + I\mathbb{E}\left[\sum_{k=0}^{m-1}\exp\left\{-\left[S_{n_k}\right]^+\right\};E\right] + N\mathbb{P}[E^c] && (2.116)\\
&\le \frac{\log M}{C} + N\mathbb{P}[E^c]] + O(I). && (2.117)
\end{aligned}
$$

In (2.113) to (2.117), E is the set $\{\tilde{j} < m\}$.

Let the scaling of m be

$$
m = \left\lceil\left(\frac{\log M}{IC_\Delta} - \frac{n_1}{I}\right) + 1\right\rceil \tag{2.118}
$$

which yields a similar choice of N as in the proof of Thm. 4:

$$
N = n_1 + (m-1)I \ge \frac{\log M}{C_\Delta}\,. \tag{2.119}
$$

The rest of the proof follows as in the proof of Thm. 4. □

2.B Chernoff Bounds for RCSP

This appendix gives the proofs of the upper and lower bounds provided in Sec. 2.5.3. Let $f_{\chi_n^2} = dF_{\chi_n^2}$ be the density function of a chi-square distribution with n degrees of freedom. The following lemma gives upper and lower bounds on the joint of a pair or error events.

Lemma 2. *For an AWGN channel, let $\{\zeta_{n_j}\}_{j=1}^m$ be the error events of RCSP with bounded-distance decoding. We have the following upper and lower bounds on the probability of a pair* *of joint error events ζ_{n_j} and ζ_{n_i} where $j < i \leq m$:*

$$\mathbb{P}[\zeta_{n_j} \cap \zeta_{n_i}] \leq \inf_{0 \leq u < 1/2} \frac{\mathbb{P}\left[\chi^2_{n_j} > (1-2u)r_i^2\right]}{e^{ur_i^2}(1-2u)^{n_i/2}}, \tag{2.120}$$

$$\mathbb{P}[\zeta^c_{n_j} \cap \zeta_{n_i}] \leq \inf_{0 \leq u < 1/2} \frac{\mathbb{P}\left[\chi^2_{n_j} \leq (1-2u)r_i^2\right]}{e^{ur_i^2}(1-2u)^{n_i/2}}, \tag{2.121}$$

$$\mathbb{P}[\zeta_{n_j} \cap \zeta_{n_i}] \geq \max\left(\mathbb{P}[\zeta_{n_j}] - w_1, \mathbb{P}[\zeta_{n_i}] - w_2\right)\ , \tag{2.122}$$

where w_1 and w_2 are given as:

$$w_1 = \inf_{v \geq 0} \frac{e^{vr_i^2}\mathbb{P}\left[\chi^2_{n_j} > (1+2v)r_j^2\right]}{(1+2v)^{n_i/2}}, \tag{2.123}$$

$$w_2 = \inf_{0 \leq v \leq 1/2} \frac{e^{-vr_i^2}\mathbb{P}\left[\chi^2_{n_j} \leq (1-2v)r_j^2\right]}{(1-2v)^{n_i/2}}. \tag{2.124}$$

Proof. It suffices to show the bounds for $\zeta_{n_1} \cap \zeta_{n_2}$. Let the set $A_i^j(r_k^2)$ be defined as:

$$A_i^j(r_k^2) = \left\{z_{i+1}^j : \|z_{i+1}^j\|^2 > r_k^2\right\}. \tag{2.125}$$

Applying the Chernoff inequality to (2.40) for $m = 2$, we have

$$\mathbb{P}[\zeta_{n_1} \cap \zeta_{n_2}] \leq \int_{r_1^2}^{\infty} \frac{\mathbb{E}e^{u\chi^2_{I_2}} f_{\chi^2_{I_1}}(t_1)}{e^{u(r_2^2 - t_1)}} dt_1 \tag{2.126}$$

$$= \int_{r_1^2}^{\infty} (1-2u)^{-I_2/2} e^{-u(r_2^2 - t_1)} f_{\chi^2_{I_1}}(t_1) dt_1 \tag{2.127}$$

$$= (1-2u)^{-I_2/2} e^{-ur_2^2} \int_{A_0^{I_1}(r_1^2)} \frac{e^{-\frac{(1-2u)}{2}\sum_{i=1}^{I_1} z_i^2}}{(2\pi)^{I_1/2}} dz_1^{I_1} \tag{2.128}$$

$$= \frac{\int_{A_0^{I_1}\left((1-2u)r_1^2\right)} \frac{e^{-\sum_{i=1}^{I_1} z_i'^2/2}}{(2\pi)^{I_1/2}} dz_1'^{I_1}}{(1-2u)^{I_2/2}(1-2u)^{I_1/2}e^{ur_2^2}} \tag{2.129}$$

$$= \frac{e^{-ur_2^2}\mathbb{P}\left[\chi^2_{I_1} > (1-2u)r_1^2\right]}{(1-2u)^{N_2/2}}, \tag{2.130}$$

where (2.129) follows from a change of variable $z_i' = (1+2u)^{1/2} z_i$. Taking the infimum over $u < 1/2$ gives the result.

We provide a sketch of the proof for (2.47) and the lower bound follows from (2.47). Observe that $\mathbb{P}[\zeta_{n_1} \cap \zeta_{n_2}] = \mathbb{P}[\zeta_{n_1}] - \mathbb{P}[\zeta_{n_1} \cap \zeta_{n_2}^c] = \mathbb{P}[\zeta_{n_2}] - \mathbb{P}[\zeta_{n_1} \cap \zeta_{n_2}^c]$. Let $w_1 = \mathbb{P}[\zeta_{n_1} \cap \zeta_{n_2}^c]$, $w_2 = \mathbb{P}[\zeta_{n_1}^c \cap \zeta_{n_2}]$ and finding the upper bounds of them yield the lower bound. The upper bound on w_2 and also (2.47) follows from the above derivation by changing the integration interval from (r_1, ∞) to $[0, r_1]$. For the upper bound on w_1, apply the Chernoff inequality with the form $\mathbb{P}[X \leq r] \leq \mathbb{E}[e^{-vX}]e^{vr}$. Taking the infimum over $v \geq 0$ gives the result. □

The following theorem extends the upper bound of the above lemma for the m-transmission joint error probability given in (2.40), which depends on the initial blocklength $n_1 = I_1$ and the $m - 1$ step sizes $I_2, \ldots, I_m$:

Theorem 9. *Let $u_i < 1/2, i = 1, 2, \ldots, m$ be the parameters for each use of Chernoff bound in the integral. Define $h_i, g_i(u_1^m)$ by the following recursion:*

$$\begin{aligned}
h_1 &= u_1 \\
g_1 &= e^{-u_1 r_m^2}(1 - 2u_1)^{\frac{-I_m}{2}}, \\
h_i &= h_{i-1} + u_i(1 - 2h_{i-1}), \\
g_i &= g_{i-1}e^{-u_i(1-2h_{i-1})r_{m-i+1}^2}\,(1 - 2h_{i-1})^{\frac{-I_{m-i+1}}{2}}.
\end{aligned}$$

Note the property that $1 - 2h_i = \prod_{j \leq i}(1 - 2u_j)$. We have

$$\mathbb{P}\left[E_{n_m}\right] \leq \inf_{u_1^m} \frac{g_{m-1}(u_1^m)\mathbb{P}\left[\chi_{I_1}^2 > (1 - 2h_{m-1})r_1^2\right]}{(1 - 2h_{m-1})^{I_1/2}}. \tag{2.131}$$

Proof. The proof follows from changing the variables iteratively similar to the proof of Lem. 2. □

Several versions of lower bounds of the joint error probability with m transmissions can be obtained by different expansions of the joint events, and the recursion formulas follow closely to those in Thm. 9 and Lem 2. The following corollary gives an example of one specific expansion that yields a lower bound in a recursive fashion and the other formulas are omitted.

Corollary 2. *With the same recursion as in Thm. 9, the lower bound is given as:*

$$\mathbb{P}\left[E_{n_m}\right] \geq \max\left\{0, p\right\} \,, \tag{2.132}$$

where p is given as

$$\mathbb{P}\left[\bigcap_{j=2}^{m} \zeta_{n_j}\right] - \inf_{u_1^m} \frac{g_{m-1}(u_1^m)\mathbb{P}\left[\chi_{I_1}^2 \leq (1-2h_{m-1})r_1^2\right]}{(1-2h_{m-1})^{I_1/2}} . \tag{2.133}$$

Proof. Expand E_m as

$$\bigcap_{1\leq j\leq m} \zeta_{n_j} = \bigcap_{2\leq j\leq m} \zeta_{n_j} \setminus \left(\bigcap_{2\leq j\leq m} \zeta_{n_j} \cap \zeta_{n_1}^c \right) \tag{2.134}$$

and the proof follows from applying Thm. 9 except for the last event ζ_{n_1}, which gives

$$\mathbb{P}\left[\chi_{I_1}^2 \leq (1-2h_{m-1})r_1^2\right]$$

instead of

$$\mathbb{P}\left[\chi_{I_1}^2 \geq (1-2h_{m-1})r_1^2\right] .$$

□

Applying Thm. 9 for the case of $m = 2$ gives a proof of Thm. 8 as shown in the following.

Proof of Thm. 8. We first show that both (2.44) and (2.45) follow immediately from properties of probability. For any two sets A, B we can write a disjoint union of A as

$$(A \cap B) \cup (A \cap B^c) = A \,. \tag{2.135}$$

Letting $A = \zeta_{n_m}$ and $B = E_{n_m} \setminus \zeta_{n_m} = E_{n_{m-1}}$, we can rewrite the expression of $\mathbb{P}[E_{n_m}]$ as

$$\mathbb{P}[E_{n_m}] + \mathbb{P}[\zeta_{n_m} \cap E_{n_{m-1}}^c] = \mathbb{P}[\zeta_{n_m}] \,. \tag{2.136}$$

We therefore have the following:

$$\mathbb{P}\left[E_{n_m}\right] = \mathbb{P}[\zeta_{n_m}] - \mathbb{P}\left[\zeta_{n_m} \cap \bigcup_{i=1}^{m-1} \zeta_{n_i}^c\right] \tag{2.137}$$

$$\geq \max \left[0, \mathbb{P}[\zeta_{n_m}] - \sum_{j=1}^{m-1} \mathbb{P}\left\{ \zeta_{n_m} \cap \zeta_{n_j}^c \right] \right\}, \tag{2.138}$$

where the last inequality can be seen as the union bound on the second term of the first equality.

To show the upper bound, a straightforward probability upper bound of E_{n_m} gives

$$\mathbb{P}\left[E_{n_j}\right] \leq \mathbb{P}[\zeta_{n_j}], \tag{2.139}$$

which can be computed with the tail of the chi-square CDF directly. Since $\mathbb{P}\left[E_{n_j}\right] \leq \mathbb{P}[\zeta_{n_{j-1}} \cap \zeta_{n_j}]$, applying Lem. 2 for upper bounds on $\mathbb{P}[\zeta_{n_{j-1}} \cap \zeta_{n_j}^c]$ finishes the proof. □

Note that alternatively, we can rewrite the joint error probability to obtain different upper bounds. For example, write $\mathbb{P}[E_{n_j}]$, $j \leq m$ as

$$\mathbb{P}\left[\bigcap_{i=1}^{j} \zeta_{n_i}\right] = \mathbb{P}\left[\bigcap_{i=1}^{j} \zeta_{n_i} \cap \zeta_{n_m}\right] + \mathbb{P}\left[\bigcap_{i=1}^{j} \zeta_{n_i} \cap \zeta_{n_m}^c\right] \tag{2.140}$$

$$\leq \mathbb{P}[\zeta_m, \zeta_j] + \mathbb{P}[\zeta_j, \zeta_{j-1}] - \mathbb{P}[\zeta_j, \zeta_{j-1}, \zeta_m]. \tag{2.141}$$

Applying Thm. 9 to the first two terms and Cor. 2 to the last term gives another version of the upper bound. Also note that for rates above capacity, only (2.139) is active since the Chernoff bounds give trivial results.

2.C The AWGN Power Constraint for RCSP

Recall that for an n-dimensional input X^n for the AWGN channel with unit noise variance and SNR η, the power constraint is given as $\sum_{j=1}^{n} X_j^2 \leq n\eta$. Assuming perfect packing of $M = 2^k$ identical Euclidean balls D_i with radii r into the outer sphere with radius $r_{\text{outer}} = \sqrt{n(1+\eta)}$, the radii for D_i is $r = \frac{\sqrt{n(1+\eta)}}{2^{k/n}}$. The codeword point, which is located at the center of the decoding region D_i, is at least distance r from the outer sphere surface assuming perfect packing and within the outer sphere.

Let $S_n(r) = \{y^n : \sum_{i=1}^{n} y_i^2 = r^2\}$ be the sphere surface with radius r. The set of $x \in \mathbb{R}^n$ that is at least distance r away from the sphere surface with radius $r_{\text{outer}} = \sqrt{n(1+\eta)}$ is given as

$$H(r) = \left\{ x^n : \sum_{i=1}^{n} (x_i - y_i)^2 \geq r^2, \forall y \in S_n\left(r_{\text{outer}}\right) \right\}. \tag{2.142}$$

Assuming $R = k/n \leq \frac{1}{2}\log_2(1+\eta)$ we have $2^{2k/n} \leq 1+\eta$ and hence

$$\begin{aligned} r^2 &= \frac{n(1+\eta)}{2^{2k/n}} && (2.143) \\ &\geq n\,. && (2.144) \end{aligned}$$

Therefore by setting $R = 1/2\log(1+\eta)$ the set $H(r)$ becomes

$$H(r) = \left\{ x^n : \sum_{i=1}^{n} (x_i - y_i)^2 \geq n, y^n \in S_n\left(\sqrt{n(1+\eta)}\right) \right\}. \tag{2.145}$$

Let $B_n(r)$ be a ball with radius r: $B_n\left(r\right) = \{x^n : \sum_i x_i^2 \leq r^2\}$. With an additional constraint that x^n must also be in the ball with radius $\sqrt{n(1+\eta)}$, we conclude that the codeword points must be in the ball $B_n\left(\sqrt{n\eta}\right)$. The maximum energy of a codeword is therefore within the power constraint $\sum_{j=1}^{n} X_j^2 \leq n\eta$ if $k/n \leq \frac{1}{2}\log_2(1+\eta)$, the capacity of the AWGN channel.

CHAPTER 3

Protograph-Based Raptor-Like LDPC Codes

In the previous chapter we studied the analyses and optimizations of repeated IR-NTC systems using a family of rate-compatible codes. We showed that if one can construct a family of rate-compatible codes that has marginal error probabilities hitting the decoding error trajectory suggested by the RCSP approximation, the same capacity-approaching expected throughput can be achieved with low latencies.

Motivated by the previous chapter, this chapter constructs a family of rate-compatible LDPC codes called Protograph-Based Raptor-Like LDPC codes, or PBRL codes. PBRL codes allow extensive rate-compatibility and efficient encoding of incremental redundancy. After a brief introduction to LDPC codes, protograph-based LDPC codes and Raptor codes, this chapter discusses the construction and optimization of PBRL codes. To provide a thorough study of the proposed PBRL codes, we give examples of constructing both long-blocklength and short-blocklength codes. We present extensive simulation results of these examples and compare the performance between PBRL codes and other standardized codes.

3.1 Introduction

This section briefly reviews the preliminaries of LDPC codes. We also briefly introduce protograph-based LDPC codes, Luby-Transform (LT) codes and Raptor codes. These codes are the inspirations of our PBRL codes.

3.1.1 LDPC Codes and Protograph-Based LDPC Codes

Denote a binary linear block code that encodes k bits to n bits as an $[n, k]$ code. A length-n parity-check code is a binary linear block code whose codewords are a collection of row vectors $c_j \in \{0, 1\}^n, j = 1, 2, \dots, 2^k$ all satisfy m linear parity-check constraints. These parity-check constraints can be expressed as an equation in **GF**(2): $Hc^T = 0$ where H is an $m \times n$ binary-valued matrix. If the rank of H is exactly m and we let $m = n - k$, then there are 2^k codewords and the parity-check code is also an $[n, k]$ code.

As the name suggests, the 1s in the parity-check matrix of an LDPC code is sparse. LDPC codes are most commonly represented by a bipartite graph, also known as Tanner graph. We give a simple example of representing a parity-check code by a bipartite graph. Let H be a parity-check matrix given as follows:

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \tag{3.1}$$

i.e. the $[7, 4]$ Hamming code. In the bipartite graph representation, there are two groups of vertices (we will use the name node rather than vertex for the rest of the dissertation): the check nodes and the variable nodes. Each row of the represents a check node and each column represents a variable node. There is an edge between the ith check node and the jth variable node if there is a 1 in the ith row and the jth column of H. The bipartite graph representation is given as in Fig. 3.1 where the squares with "+" signs represent check nodes and black circles represent variable nodes. The first variable node from the left of the graph represents the first column of the parity-check matrix H, and the first check node from the left represents the first row of H.

Regarding the $[7, 4]$ Hamming code as an LDPC code[1], we note that it is in fact an "irregular" LDPC codes where the variable nodes have different node degrees. The degree profile of an irregular LDPC code can be denoted by the "degree distribution" either from the node

[1] Though the example of $[7, 4]$ Hamming does not manifest the "low-density" property of LDPC codes, it is a simple example to illustrate the main ideas of representing an LDPC code using a bipartite graph.

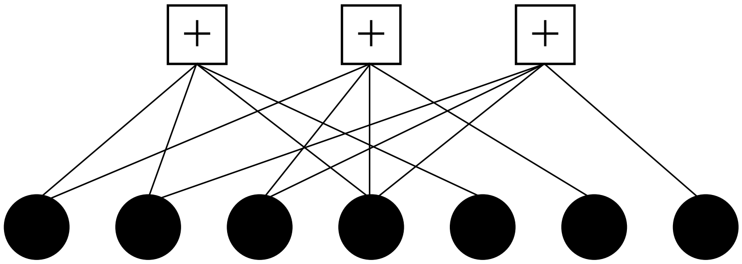

Figure 3.1: Bipartite graph representation of the $[7,4]$ Hamming code.

perspective or the edge perspective, which can be mapped from one to the other for a given parity-check code. A degree distribution from the edge perspective is denoted by a pair of polynomials $\lambda(x) = \sum_j \lambda_j x^{j-1}$ and $\rho(x) = \sum_j \rho_j x^{j-1}$, both with non-negative coefficients such that $\lambda(1) = 1, \rho(1) = 1$. The coefficients λ_j and ρ_j denote the *fraction of edges* that connects to variable nodes and check nodes with degree j, respectively. The details of the relations between the two perspectives can be found in [RU08]. In the $[7,4]$ Hamming code example, the degree distribution is given as

$$\lambda(x) = \frac{1}{4} + \frac{1}{2}x + \frac{1}{4}x^2 \,, \tag{3.2}$$

$$\rho(x) = x^3 \,. \tag{3.3}$$

The decoding of an LDPC code is based on message-passing algorithm, or belief propagation (BP), on the bipartite graph of the code. A large volume of work has been dedicated to the study of the decoding algorithm for LDPC codes. We refer readers to [MMC98, Ana01, KFL01, VGW10] and the references therein. We will use the standard BP decoding for the binary-input AWGN channels.

In order to facilitate efficient hardware implementation of the decoder, LDPC codes are often constructed with certain structures, e.g. quasi-cyclic LDPC codes and regular LDPC codes [LC04]. Protograph-based LDPC codes, or simply protograph codes, also allows efficient hardware implementation [DDJ09]. A protograph, or projected graph, is a bipartite graph with a relatively small number nodes. A copy-and-permute operation, often referred to as the "lifting" process, can be applied to the protograph to obtain larger graphs of various sizes, resulting in a

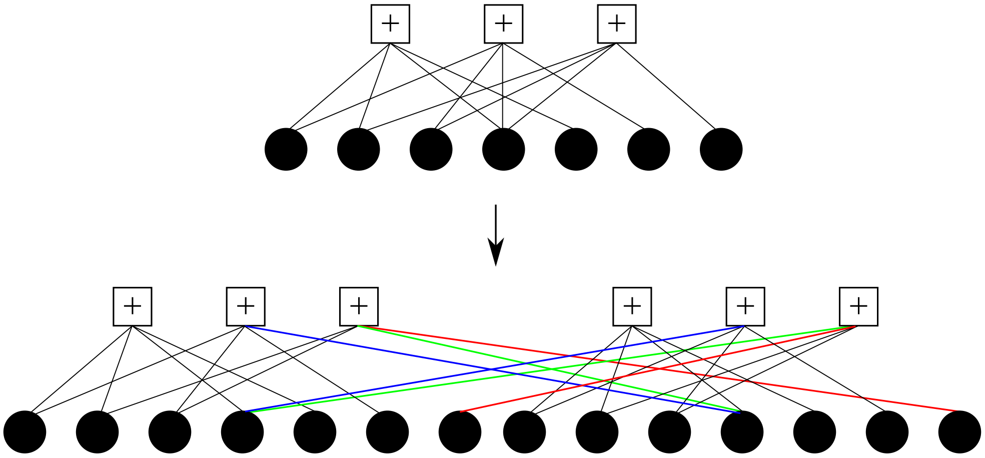

Figure 3.2: Lifting process of the $[7, 4]$ Hamming code.

larger LDPC code. After the lifting operation, the edges of the same type are permuted among the protograph replicas. Fig. 3.2 shows an example of the lifting process for the $[7, 4]$ Hamming code. For clarity only three types of edges are permuted and each type is color coded with the same color. Note that the protograph can also have parallel edges, i.e. multiple edges connecting the same pair of variable node and check node. Parallel edges in a protograph can later be removed through the lifting process. A detailed discussion on parallel edges in protographs can be found in [DDJ09].

3.1.2 LT Codes and Raptor Codes

This section reviews the preliminaries of LT codes [Lub02] and Raptor codes [Sho06]. Introduced by Luby [Lub02] and Shokrollahi [Sho06], LT codes and Raptor codes share many similarities with LDPC codes and are shown to achieve binary erasure channel (BEC) capacity universally. Etesami et al. [ES06] explore the application of Raptor codes to binary memoryless symmetric channels and derive various results, including the fact that Raptor codes are not universal except in BEC. Results on Raptor codes such as [Sho06] and [ES06] rely heavily on the assumption of large information blocks.

An LT code is described by the output degree distribution Ω on its output symbols, which is the node perspective of the degree distribution. Let n be a positive integer that denotes the number of the input symbols. Let $\mathbf{\Omega} = [\Omega_1, \Omega_2, \dots, \Omega_n]$ be a distribution on a set of integers $\{1, 2, \dots, n\}$ such that Ω_j denotes the probability of the value j being chosen. For the ith output bit, the encoder first chooses an integer d_i randomly according to the distribution $\mathbf{\Omega}$. It then chooses d_i input symbols uniformly (without replacement) from $\{1, 2, \dots, n\}$, and taking exclusive-or of these chosen input bits yields the output bit. This encoding process continues indefinitely for $i = 1, 2, \dots,$ often concluding only when all interested receivers have been able to decode the message.

Let $\mathcal{C}$ be an $[n, k]$ linear code. A Raptor code is a serial concatenation of a code $\mathcal{C}$, which is also called the "precode," and an LT code. A Raptor code is described by the parameters $(k, C, \Omega(x))$, where $\Omega(x) = \sum_{i=k}^{n} \Omega_i x^i$ is the generator polynomial of the output degree distribution of the LT code.

The decoding of the Raptor code is performed in two stages: the decoder first decodes the LT code and recovers a fraction of the precoded symbols (or provides the soft information of the precoded symbols in the case of AWGN channel). The decoder then attempts to recover the remaining symbols by decoding the precoded symbols with the precode $\mathcal{C}$.

Raptor codes can also be represented as a bipartite graph and BP decoding can be applied. This is the inspiration for us to consider the bipartite graph structure of Raptor codes to build our rate-compatible LDPC codes.

3.1.3 Organization

We briefly summarize the organization of this chapter. Sec. 3.2 introduces the structure of PBRL codes. The encoding and decoding are also discussed in this section. Sec. 3.3 reviews the density evolution process and discusses the optimization of PBRL codes. Sec. 3.4 studies a variant of PBRL codes, called Punctured-Node PBRL (PN-PBRL) codes with an emphasis in the short-blocklength regime. Examples of constructing PBRL and PN-PBRL codes with short blocklengths and the corresponding simulation results are provided in Sec. 3.5. Extending the result in

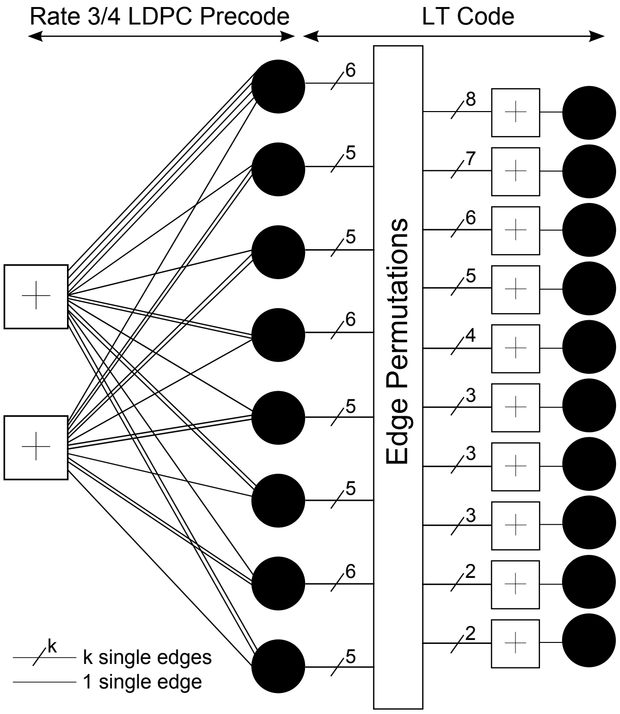

Figure 3.3: Protograph for a PBRL code with a rate-$3/4$ precode. The matrices shown in (3.6) and (3.7) in Sec. 3.5 give the details of the edge connections in the LT code part.

Sec. 3.4, Sec. 3.6 investigates the construction and optimization of PN-PBRL codes in the long-blocklength regime. An example of long-blocklength PN-PBRL codes and the corresponding simulation results are presented in Sec. 3.7. Finally Sec. 3.8 concludes this chapter.

3.2 Protograph-Based Raptor-Like LDPC Code

This section introduces the structure of PBRL codes. The encoding and decoding of PBRL codes are also discussed.

3.2.1 Introducing PBRL Codes

The structure of a PBRL code can be best illustrated by its protograph. Fig. 3.3 shows the protograph of a PBRL code. This protograph consists of two parts: (1) a relatively simple protograph code (on the left) representing the protograph of the precode and (2) a number of check nodes (on the right) that are each connected to several variable nodes of the first part and an additional degree-one variable node. The second part represents the protograph of an LT code. The highest rate of the code is rate $3/4$, and subsequent lower-rate codes are obtained by transmitting the variable nodes in the LT code protograph starting from the top node.

After the lifting operation, the first part can be seen as an LDPC precode, and the degree-one variable nodes of the second part can be efficiently encoded with the precoded symbols in a manner similar to the LT code. The structure of this protograph code resembles a Raptor code, but with a deterministic (rather than random) encoding rule for combining the precoded symbols. The rate of the precode in this example is $3/4$. As we increase the number of transmitted degree-one variable nodes in the LT part, the code rate is reduced gradually.

3.2.2 Decoding and Encoding of PBRL Codes

Consider the decoding of a traditional Raptor code that collects the precoded symbols and encodes them with an LT code. In the case of an LDPC precode used with an LT code, decoding proceeds as follows: the decoder first performs BP decoding on the LT code and then performs BP decoding on the precode. The two-stage decoding implies the use of two different BP decoders, each exchanging their extrinsic information after the iterative decoding.

In [Sol06], the authors commented that the complexity of Raptor codes is higher than rate-compatible LDPC codes. In view of reducing system complexity, it is natural to consider a joint decoding of the Raptor code. The PBRL code family has deterministic connections in the LT code part and always transmits the output symbols of the precode, allowing joint decoding of the LT code part and the LDPC precode. This property also guarantees that the BP algorithm will always work for the initial transmission as well as the lower-rate codewords comprised of the original transmission and additional incremental redundancy. For traditional Raptor codes that use randomized encoding, the initial transmission may not contain enough information for BP decoding to succeed even in a noiseless setting.

For high-rate codes, the decoder can deactivate those check nodes in the LT part for which the neighboring degree-one variable node is not used. At the highest rate, when only the precode is transmitted, none of the check nodes in the LT part need to be activated, offering possible complexity reduction.

The encoding of the PBRL codes is as efficient as Raptor codes: after encoding the precode, the encoding of the LT code part only involves exclusive-or operations. A discussion on the encoding of the protograph precode can be found in [DDJ09].

3.3 Optimization of Protograph-Based Raptor-Like LDPC Codes

This section proposes a design technique for finding good PBRL codes. Belief propagation (BP) decoding is assumed and we begin by designing the protograph. Given a fixed initial code rate, the design begins by finding a good protograph code to serve as the precode and then optimize the protograph of the LT code part. Optimization of the precode protograph is the same as finding a good protograph code. When lifting a protograph with random permutations, the asymptotic meaning of a good protograph is to have its minimum distance growing linearly with its blocklength. We refer readers to a seminal work by Divsalar et al. [DDJ09], which contains an extensive study of finding good protograph codes.

3.3.1 Density Evolution with Reciprocal Channel Approximation

For the BI-AWGN channel, the asymptotic *iterative decoding threshold* [RU01] characterizes the performance of the ensemble of LDPC codes based on a specified protograph. This threshold indicates the minimum SNR required to transmit reliably with the underlying ensemble of codes as the blocklength grows to infinity.

Computing the exact iterative decoding threshold for BI-AWGN requires a large amount of computation. The reciprocal channel approximation (RCA) [Chu00, DDJ09] provides a fast and accurate approximation to the density evolution originally proposed by Richardson et al. [RU01, RSU01]. Experimental results [DDJ09, Chu00] show that the deviation from the exact density evolution is less than 0.01 dB. The following subsection describes an optimization process that uses the approximated threshold.

The reciprocal channel estimation for BI-AWGN channel uses a single real-valued parameter s, the SNR, to approximate the density evolution. Define the reciprocal SNR as $r \in \mathbb{R}$ such that $C(s) + C(r) = 1$ where $C(x)$ is the capacity of the BI-AWGN channel with SNR x:

$$C(x) = 1 - \int_{-\infty}^{\infty} \log_2 \left(1 + \exp\{4\sqrt{x}(u - \sqrt{x})\}\right) \frac{\exp\{-u^2\}}{\sqrt{\pi}} du \,. \tag{3.4}$$

The self-inverting reciprocal energy function $R(x) = C^{-1}(1 - C(x))$ [Chu00] transforms parameters s and r to each other. In other words, $r = R(s)$ and $s = R(r)$.

Let s_e be the message passed along an edge e from a variable node to a check node and r_e be the message passed along an edge e from a check node to a variable node. Let E_c be the set of edges that connect to the check node c and E_v be the set of edges that connect to the variable node v. The application of the RCA technique to the density evolution for a fixed channel SNR s_{chl} is summarized as follows:

0) *(Initial step)* For edges e connected to punctured variable nodes, set $s_e = 0$. For all other edges set $s_e = s_{\text{chl}}$.

1) The check node message passed along an edge e is generated by $r_e = \sum\limits_{i \in E_c \setminus e} R(s_i)$.

2) The variable node message passed along edge e is generated by $s_e = s_{\text{chl}} + \sum\limits_{i \in E_v \setminus e} R(r_i)$.

3) Repeat the iterative process from step 1) unless the stopping condition is met.

Using the above computation, the values s_e is additive at the variable nodes and the values r_e is additive at the check nodes for all edges. For a fixed precision, we apply the above procedure iteratively and tack the values of all messages s_e. The smallest value of s_{chl} that achieves unbounded growth of all messages s_e is the approximated density evolution threshold and the search can be obtained by a bisection search due to the monotonicity of the threshold [RU08]. This method of reciprocal channel approximation of the threshold will be used in the optimization process described in the following subsection.

3.3.2 Optimizing the LT code Part for PBRL Codes

Given a precode protograph constructed based on the techniques described in [DDJ09], we proceed to construct the LT code part. To construct the protograph of the LT code part, first add a new check node and a new degree-one variable node to the protograph. Connect the new check node and the new degree-one variable node with an edge. Additional edges are added between the new check node and the precoded variable nodes. Using the new degree distribution we can use RCA to compute the threshold. We find the connection that gives the lowest threshold among all possible connections, or a well-chosen subset of all possible connections. This process continues until the underlying protograph reaches the lowest rate desired.

For a given precode protograph, the optimization procedure of the LT code part is summarized as follows:

1. Add a new check and a new variable node that are connected to each other into the current protograph.

2. Using RCA to approximate the threshold, find the optimal connection between the new check node and the precoded symbols that gives the lowest threshold.

3. Update the protograph using the optimized one obtained in step 2). Start over with step 1) if the lowest rate desired is not yet reached. Go to step 4) if the lowest rate is reached.

4. Lift the resulting protograph with the circulant permutations to match the desired initial blocklength.

5. Select circulant permutations so that small cycles are avoided when the protograph is lifted. The selection is based on the circulant progressive edge growth (PEG) algorithm [HEA05, ADD04].

Table 3.1: Thresholds of the PBRL Codes (E_b/N_0 in decibels).

Rate	Threshold	Capacity	Gap
$6/8$	2.196	1.626	0.570
$6/9$	1.804	1.059	0.745
$6/10$	1.600	0.679	0.921
$6/11$	1.464	0.401	1.063
$6/12$	1.358	0.187	1.171
$6/13$	1.250	0.018	1.232
$6/14$	1.136	-0.122	1.258
$6/15$	1.016	-0.238	1.254
$6/16$	0.922	-0.337	1.259
$6/17$	0.816	-0.422	1.238
$6/18$	0.720	-0.495	1.215

We observed that in the optimization process, parallel edges in the *LT code* part of the protograph should be kept to a minimum (at most one pair of parallel edges in our examples). This prevents short-cycles in the lifting process. Fig. 3.3 is an example of an optimized PBRL code. This code *does not* have any parallel edges in the LT code part. Experimental results indicate that for PBRL codes with short blocklengths, direct lifting of the protograph with parallel edges yields better codes than a two-stage lifting such as the one described in [DDJ09].

The initial code rate, or the precode code rate, is $3/4$. The threshold of the precode is 2.196 dB (E_b/N_0). Suppose that the code is lifted 32 times, the initial blocklength is then 256. With a step size of 32, subsequent code rates $6/9, 6/10, \ldots, 6/18$ are obtained by transmitting the output symbols of the LT code from each successive group of variable nodes starting from the top.

The corresponding thresholds of each code rate are summarized in Table 3.1. We observe an increase in the gap between the threshold and the capacity as the code rate decreases. This is due

Table 3.2: Thresholds of the PN-PBRL LDPC Codes (E_b/N_0 in decibels).

Rate	Threshold	Capacity	Gap
$6/8$	2.020	1.626	0.394
$6/9$	1.638	1.059	0.579
$6/10$	1.468	0.679	0.789
$6/11$	1.352	0.401	0.951
$6/12$	1.248	0.187	1.061
$6/13$	1.186	0.018	1.168
$6/14$	1.018	-0.122	1.140
$6/15$	0.930	-0.238	1.168
$6/16$	0.848	-0.337	1.185
$6/17$	0.692	-0.422	1.114
$6/18$	0.602	-0.495	1.097

to the structural restrictions imposed on the protograph of the LT code part. Each subsequent protograph inherits the connections of the next-higher-rate protograph; the new protograph can only optimize over the connections emanating from the one additional check node. Also, the new check node must connect with a new degree-one variable node. This diminishing return of lowering the code rate is mitigated by introducing punctured variable nodes as we will discussed in the following section.

3.4 Punctured-Node PBRL Codes with Short Blocklengths

This section introduces Punctured-Node PBRL codes (PN-PBRL codes) that have structure similar to PBRL codes, but the protograph of the precode has at least one punctured (untransmitted) node.

Fig. 3.4 shows an example of an optimized PN-PBRL code. Note that the first variable node

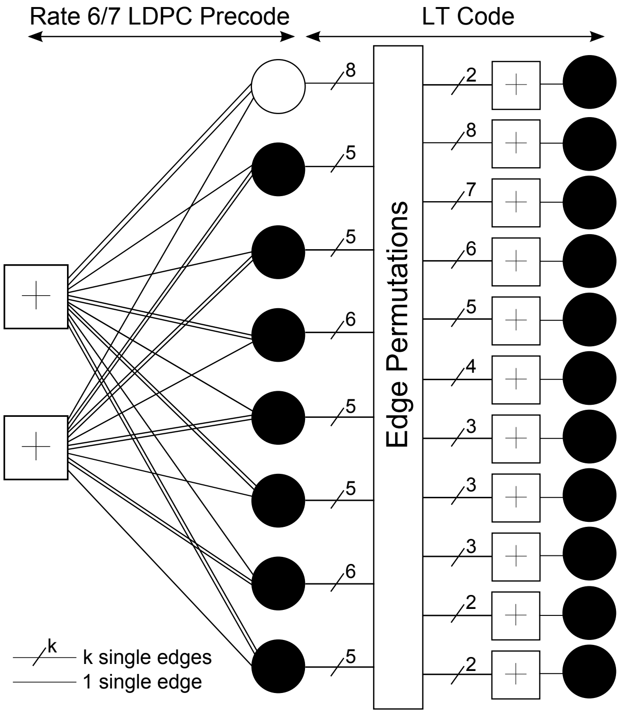

Figure 3.4: Protograph of a PN-PBRL code with a rate-$6/7$ precode. The first node in the precode is always punctured, denoted as a white circle. Lower-rate codes are obtained by transmitting the variable nodes in the LT code protograph starting from the top node. The matrices shown in (3.8) and (3.9) in Sec. 3.5 give the details of the edge connections in the LT code part.

of the precode protograph is punctured, giving a rate-$6/7$ precode. To obtain an initial code rate of $3/4$, the first variable node of the LT code protograph is transmitted. The optimization procedure is the same as in Sec. 3.3 but with a slight modification at step 2. In the optimization step 2 for PBRL codes we simply search for the best connection since the precoded symbols are all of the same type. In the case of PN-PBRL codes, whenever there is a tie in the decoding threshold for different connections, preference is given to the one with no edge connected to the punctured node. This heuristic helps prevent performance degradation when lifting the protograph to a shorter code (for blocklength less than 1000).

The subsequent code rates of $6/9, 6/10, \ldots, 6/18$ are obtained by transmitting the variable nodes of the LT code protograph from top to bottom. Regardless of the operating rate, the first variable node of the precode protograph is always punctured. The PN-PBRL codes yield better

Table 3.3: Thresholds of the PN-PBRL LDPC Codes with Parallel Edges(E_b/N_0 in decibels).

Rate	Threshold	Capacity	Gap
$6/8$	1.965	1.626	0.339
$6/9$	1.314	1.059	0.255
$6/10$	0.948	0.679	0.269
$6/11$	0.678	0.401	0.277
$6/12$	0.422	0.187	0.235
$6/13$	0.270	0.0179	0.252
$6/14$	0.118	-0.122	0.240
$6/15$	0.005	-0.238	0.243
$6/16$	-0.102	-0.337	0.235
$6/17$	-0.172	-0.422	0.250
$6/18$	-0.266	-0.495	0.229

thresholds as shown in Table 3.2.

We observed experimentally that adding more parallel edges connected between the punctured variable node in the precode protograph and the check nodes in the LT code protograph reduces the threshold significantly, as shown in Table 3.3. The gap between the threshold and the capacity are all less than 0.34 dB. The lifted codes with blocklength 256, however, do not manifest the gain obtained in threshold. This is because the large number of extra parallel edges is likely to cause undesirable trapping sets or absorbing sets in the bipartite graph, especially when a short blocklength such as 256 is used.

3.5 Examples and Simulations for Short-Blocklength PBRL Codes

This section provides examples of short-blocklength PBRL and PN-PBRL codes and presents the frame error rate (FER) and bit error rate (BER) simulations of the codes. Lifting of the protograph is accomplished by circulant permutation of each edge, which allows efficient implementation of the decoder. The design of the circulant permutation uses a greedy algorithm, called circulant progressive edge growth algorithm, to avoid all length-4 cycles and minimizes the number of length-6 cycles.

The PBRL and PN-PBRL codes that are considered in this section can be described as follows: let H_p be the parity-check matrix of the precode and H_LT be the parity-check matrix of the LT code excluding the degree-one variable nodes. Let σ be a 32×32 identity matrix shifted to the left by 1, I be the identity matrix and O be the all-zero matrix with proper dimensions. Entries with multiple terms of σ indicate parallel edges in the protograph. The full parity-check matrix for both examples can be expressed as

$$H = \begin{bmatrix} H_p & O \\ H_{LT} & I \end{bmatrix}. \tag{3.5}$$

For the PBRL code example, H_p is given as

$$H_p = \begin{bmatrix} \sigma^0+\sigma^1+\sigma^3+\sigma^7 & \sigma^{24} & \sigma^{14} & \sigma^{17}+\sigma^0 & \sigma^7 & \sigma^1+\sigma^6 & \sigma^{21} & \sigma^{21}+\sigma^0 \\ \sigma^4 & \sigma^4+\sigma^9 & \sigma^0+\sigma^1 & \sigma^0 & \sigma^0+\sigma^2 & \sigma^0 & \sigma^0+\sigma^3 & \sigma^2 \end{bmatrix}, \tag{3.6}$$

and H_{LT} is given as

$$H_{LT} = \begin{bmatrix} \sigma^{29} & \sigma^0 & \sigma^0 & \sigma^1 & \sigma^5 & \sigma^6 & \sigma^{10} & \sigma^4 \\ \sigma^{12} & \sigma^0 & \sigma^1 & \sigma^3 & \sigma^4 & \sigma^{16} & \sigma^{13} & 0 \\ \sigma^{16} & \sigma^0 & \sigma^2 & \sigma^6 & \sigma^0 & 0 & 0 & \sigma^1 \\ \sigma^{26} & 0 & \sigma^0 & 0 & 0 & \sigma^1 & \sigma^6 & \sigma^9 \\ 0 & \sigma^1 & 0 & 0 & \sigma^0 & 0 & \sigma^2 & \sigma^3 \\ \sigma^1 & 0 & 0 & \sigma^2 & 0 & \sigma^9 & 0 & 0 \\ 0 & 0 & \sigma^{16} & 0 & \sigma^0 & 0 & \sigma^4 & 0 \\ 0 & \sigma^{21} & 0 & \sigma^0 & 0 & \sigma^2 & 0 & 0 \\ \sigma^0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^1 \\ 0 & 0 & 0 & \sigma^{12} & 0 & 0 & \sigma^0 & 0 \end{bmatrix}. \tag{3.7}$$

For the PN-PBRL code example the precode parity-check matrix H_p is given as

$$H_p = \begin{bmatrix} \sigma^0+\sigma^1 & \sigma^{24} & \sigma^{14} & \sigma^{17}+\sigma^5 & \sigma^7 & \sigma^1+\sigma^3 & \sigma^{21} & \sigma^{21}+\sigma^0 \\ \sigma^4 & \sigma^0+\sigma^2 & \sigma^0+\sigma^3 & \sigma^{31} & \sigma^6+\sigma^0 & \sigma^1 & \sigma^0+\sigma^1 & \sigma^2 \end{bmatrix}, \tag{3.8}$$

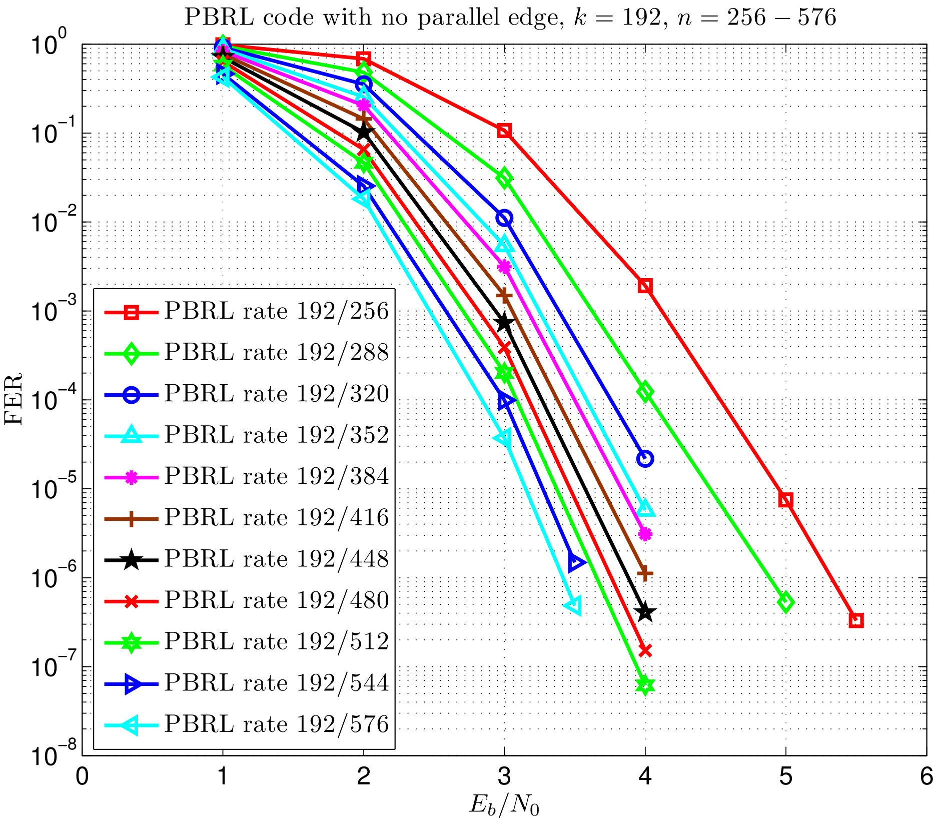

Figure 3.5: Frame error rate of the rate-compatible PBRL code family. Layered belief propagation is used for the decoder simulations.

and the LT code parity-check matrix H_{LT} is given as

$$H_{LT} = \begin{bmatrix} \sigma^2+\sigma^0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sigma^{29} & \sigma^0 & \sigma^2 & \sigma^0 & \sigma^9 & \sigma^6 & \sigma^7 & \sigma^6 \\ \sigma^{12} & \sigma^0 & \sigma^4 & \sigma^1 & \sigma^5 & \sigma^4 & \sigma^{10} & 0 \\ \sigma^{16} & \sigma^0 & \sigma^5 & \sigma^6 & \sigma^1 & 0 & 0 & \sigma^{11} \\ \sigma^{26} & 0 & \sigma^0 & 0 & 0 & \sigma^2 & \sigma^9 & \sigma^0 \\ 0 & \sigma^1 & 0 & 0 & \sigma^0 & 0 & \sigma^3 & \sigma^0 \\ \sigma^1 & 0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & 0 \\ 0 & 0 & \sigma^{16} & 0 & \sigma^0 & 0 & \sigma^2 & \sigma^0 \\ 0 & \sigma^{21} & 0 & \sigma^0 & 0 & \sigma^1 & 0 & 0 \\ \sigma^0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma^1 \\ 0 & 0 & 0 & \sigma^{12} & 0 & 0 & \sigma^0 & 0 \end{bmatrix}. \tag{3.9}$$

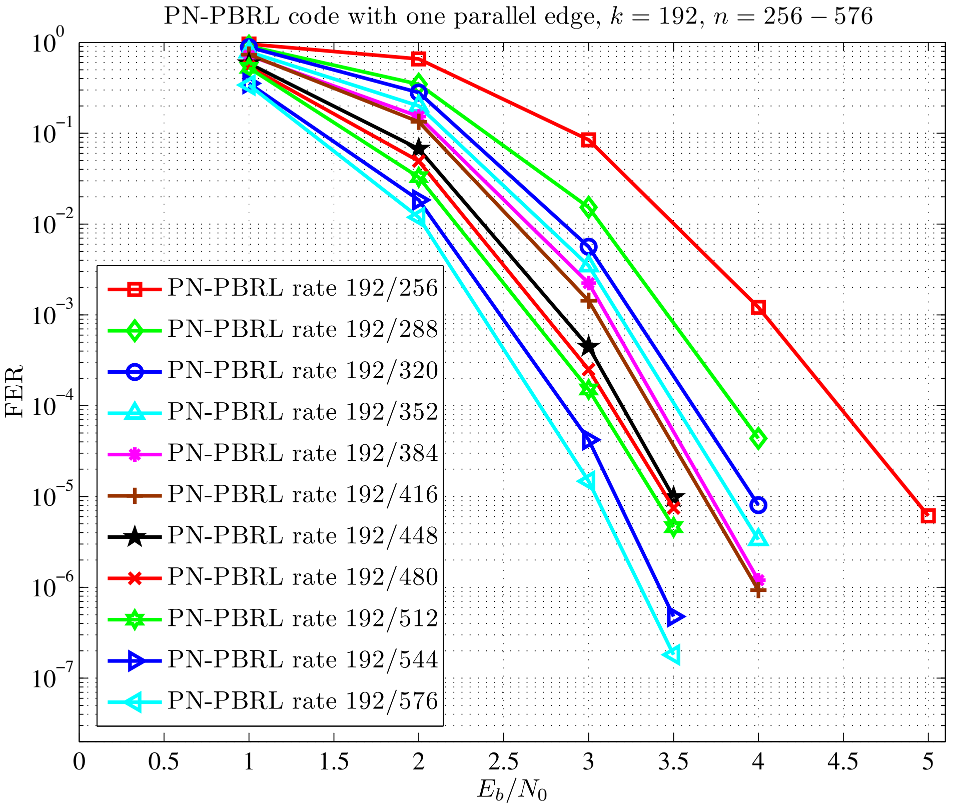

Figure 3.6: Frame error rate of the rate-compatible PN-PBRL code family. Layered belief propagation is used for the decoder simulations.

Figs. 3.5 and 3.6 show the simulations of the PBRL and PN-PBRL code family with rates $6/8, 6/9, \ldots, 6/18$. Layered belief propagation is used for the decoder simulations shown in Figs. 3.5 and 3.6. As shown in both figures, we observe a saturation of the performance. This coincide with the decoding threshold results as as shown in Table 3.1 and Table 3.2. Also consistent with the decoding threshold results, we see that the PN-PBRL code family outperforms the PBRL code family with a slight increase of encoding complexity.

Considering Raptor codes with the same precode as the PBRL code and the output distributions drawn from [ES06] and [Sol06], the simulation results show that these Raptor codes with information block of 192 bits have frame error rates much higher than both PBRL and PN-PBRL codes. This result is not surprising because a relatively short block of information is considered: since the degrees of each output node are drawn at random according to the optimal degree

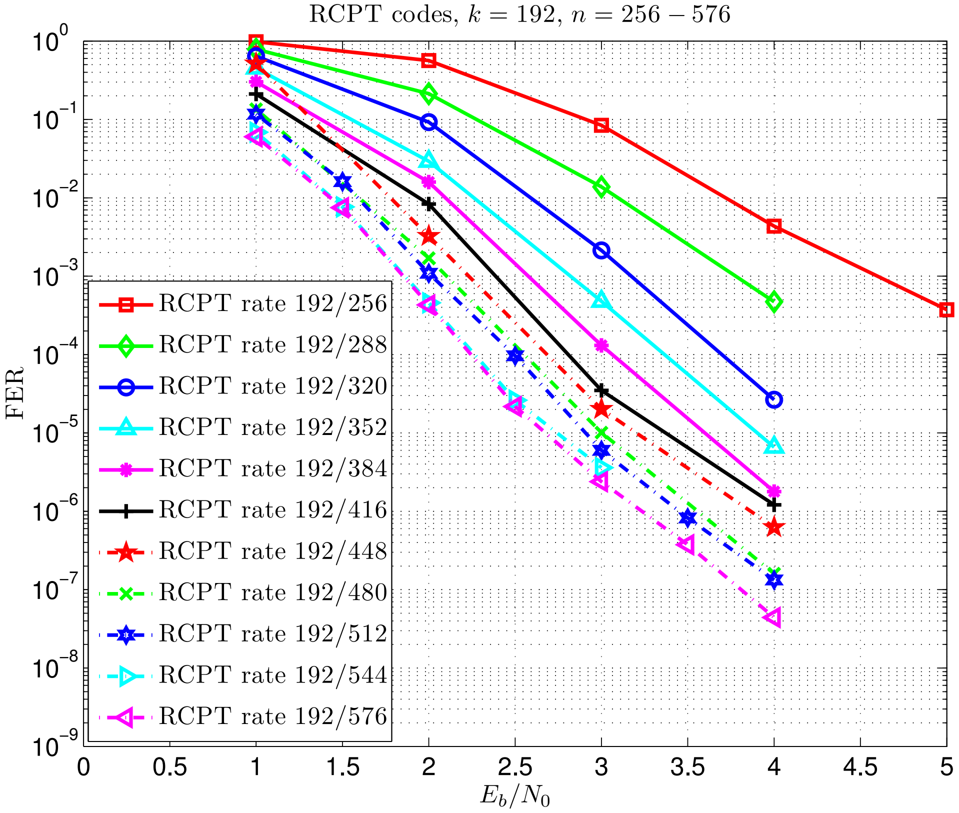

Figure 3.7: Frame error rate of the rate-compatible RCPT code family. Iterative BCJR algorithm is used for decoding with maximum 12 iterations.

distribution, a few hundreds of samples might not be enough to exhibit the optimal degree distribution. Fig. 3.7 shows the simulations of RCPT codes in 3GPP-LTE standard with the same range of code rates and blocklengths. Different code rates of the RCPT codes are obtained by pseudo-random puncturing described in [Gen08]. Although the RCPT code family performs better at low SNR regime, it also suffers from an error floor as soon as the FER reaches 10^{-3} for rate $3/4$ and 10^{-5} for rate $1/3$, respectively.

For ease of comparison, Fig. 3.8 and 3.9 separately plot the FER and BER with rates $3/4$ and $1/3$ of the PBRL codes, PN-PBRL codes and RCPT codes. Note that in Figs. 3.8 and 3.9, flooding is used for decoder simulations, which only gives slightly worse performance than the layered belief propagation decoding used in Figs. 3.5 and 3.6. At rate $3/4$, the PN-PBRL code performs similarly to RCPT code and outperforms RCPT code when SNR is higher than 3 dB in

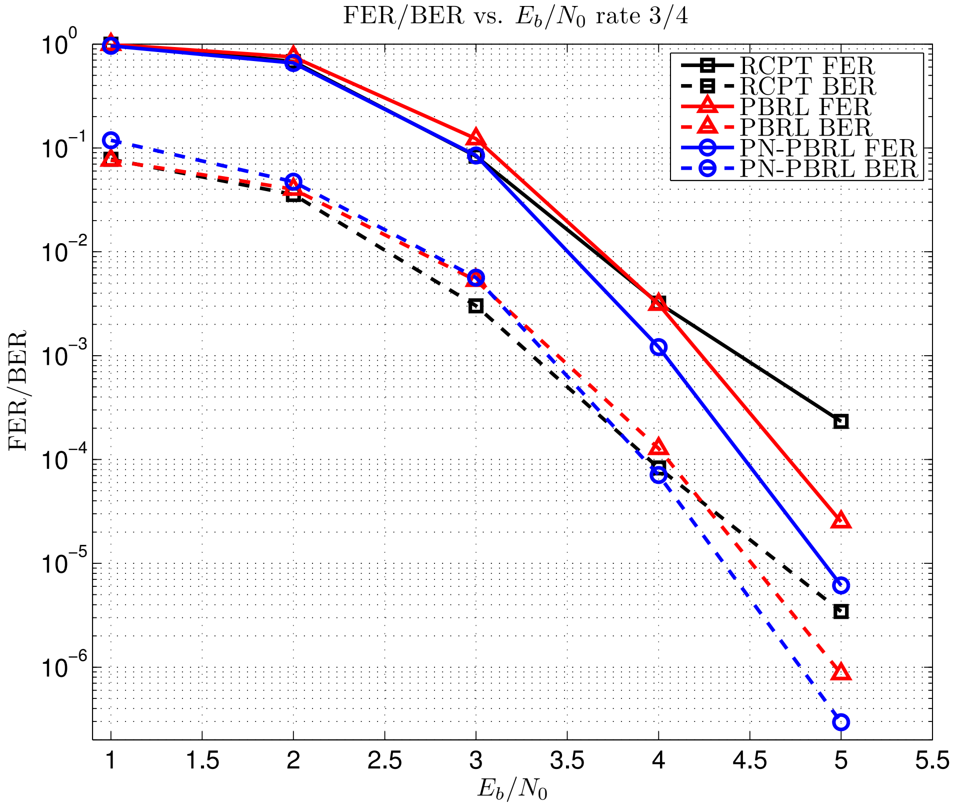

Figure 3.8: Frame error rate and bit error rate of the PBRL code, PN-PBRL code and RCPT code at code rate $3/4$. Flooding is used for the decoder simulations. Both PBRL and PN-PBRL codes outperform the RCPT codes at high E_b/N_0 regime but perform slightly worse than the RCPT code in the low E_b/N_0 regime.

terms of FER. At rate $1/3$, the PN-PBRL code starts to gain an advantage at SNR higher than 3.5 dB in terms of FER.

3.6 PN-PBRL Code Construction for Long Blocklengths

Motivated by the low thresholds observed in Table. 3.3, this section investigates the construction and optimization of PN-PBRL codes with long blocklengths. Similar to its short-blocklength counterpart, the design begins by finding a good protograph LDPC code to serve as a precode, i.e. a protograph with linear growth of its minimum distance as a function of blocklength. The

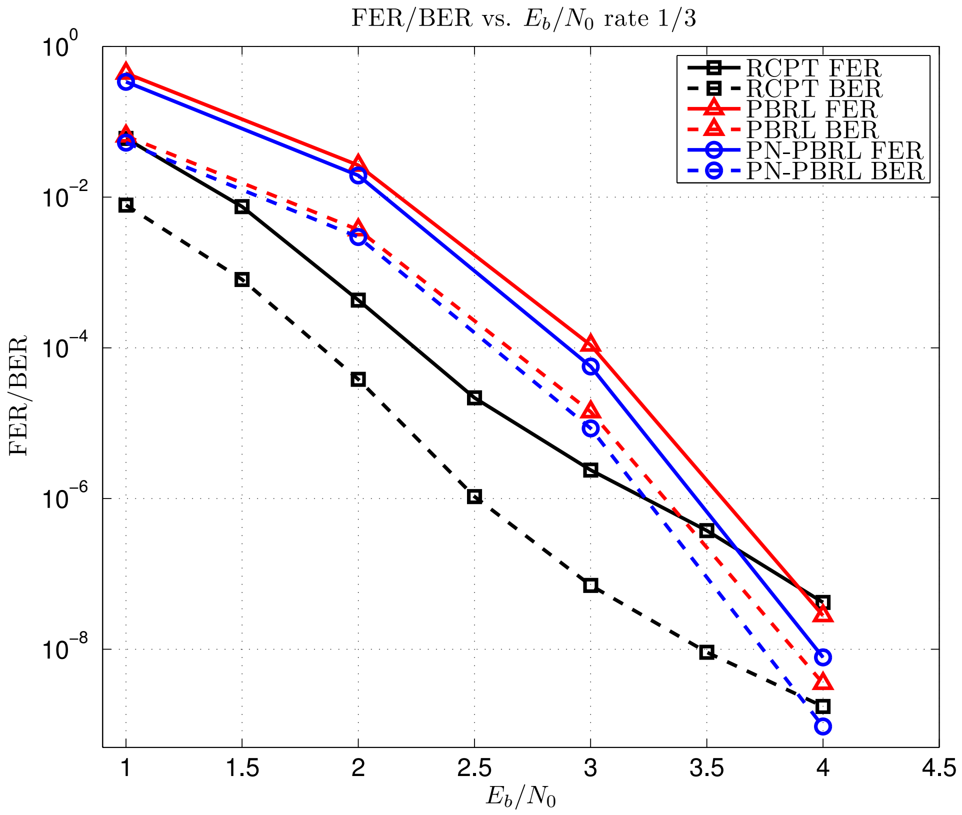

Figure 3.9: Frame error rate and bit error rate of the PBRL code, PN-PBRL code and RCPT code at code rate $1/3$. Flooding is used for the decoder simulations. Here the RCPT code outperforms the PN-PBRL and PBRL code at low SNR range, but the PN-PBRL code starts to outperform the RCPT code at around SNR 3.5 dB. There is no sign of an error floor for the both PBRL and PN-PBRL code.

construction of the LT code part, however, requires different design criteria as for the short-blocklength case in order to obtain good thresholds at each subsequent code rate.

First of all, we observed that the punctured node of the precode must connect to all check nodes in the LT code part with at least a single edge. These edges induce high-degree punctured variable nodes at low rates. We observe experimentally that the removal of these edges would result in a notable degradation of the thresholds at low rates. This approach was originally avoided in the design of short-blocklength PN-PBRL codes as described in Sec 3.4. For codes with short blocklengths, the high-degree punctured node creates undesirable structures under BP decoding. Thus for short-blocklength codes, the resulting performance of the lifted codes does not correspond to the threshold results in the cases we studied, making the optimization process ad hoc rather than systematic. However, for longer blocklengths we see good agreement between the thresholds and the simulated performance.

Second, additional parallel edges between the punctured variable node in the precode protograph and the check nodes in the LT code protograph can further reduce the thresholds. Table 3.4 summarizes the thresholds of the code that has parallel edges between the punctured node and every check node in the LT code part. The gaps between the thresholds and the capacities are all less than 0.34 dB except the highest rate (the precode). The threshold of the rate-$1/3$ code is -0.266 dB and the gap to BI-AWGN capacity is only 0.229 dB.

The lifted codes for low rates, however, do not manifest the gain obtained in terms of thresholds. This is because that the connections of parallel edges hinder the flow of information from the (lifted) degree-one variable nodes to the precoded variable nodes, preventing the BP decoder from converging. The thresholds will slightly increase by limiting the number of parallel edges between the punctured variable node and the check nodes in the LT code part, but the lifted code graph will allow BP decoder to converge with a reasonable number of iterations.

For a given protograph precode with the property of linear growth in minimum distance, the optimization procedure of the LT code part is summarized as follows:

1. Add a new check and a new variable node that are connected to each other into the current protograph. Add an edge between the new check node and the punctured node in the precode protograph.

2. Using RCA to approximate the threshold, find the optimal connections between the new check node and the precoded symbols that gives the lowest threshold. The connection between the new check node and the punctured node in the precode is retained.

3. Update the protograph using the optimized one obtained in step 2). Start over with step 1) if the lowest rate desired is not yet reached. Go to step 4) if the lowest rate is reached.

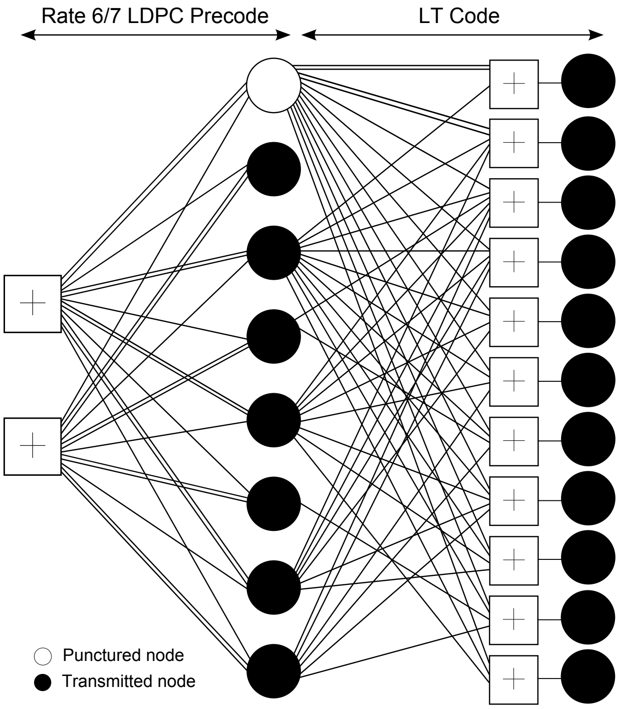

Figure 3.10: PN-PBRL code with a rate-$6/7$ precode producing thresholds shown in Table 3.5. The first node in the precode is always punctured. Lower-rate codes are obtained by transmitting the variable nodes in the LT code protograph starting from the top node.

4. Lift the resulting protograph with the circulant permutations to match the desired initial blocklength.

5. Select circulant permutations so that small cycles are avoided when the protograph is lifted. The selection is based on the circulant progressive edge growth (PEG) algorithm [HEA05, ADD04].

Note that parallel edges are allowed in step 2 but should be kept to a small number in the optimization process to avoid convergence problem in the lifted code graph.

Fig. 3.10 shows an example of the optimized protograph with at most two parallel edges between the punctured variable node in the precode part and the check nodes in the LT code part. The resulting thresholds obtained from the protograph in Fig. 3.10 are shown in Table 3.5. We comment that the decoding thresholds in Table 3.5 are all lower than the thresholds in the

Table 3.4: Thresholds of the PN-PBRL Codes with additional parallel edges to punctured node. (E_b/N_0 in decibels).

Rate	Threshold	Capacity	Gap
$6/7$	3.077	2.625	0.452
$6/8$	1.956	1.626	0.330
$6/9$	1.314	1.059	0.255
$6/10$	0.948	0.679	0.269
$6/11$	0.678	0.401	0.277
$6/12$	0.422	0.187	0.235
$6/13$	0.270	0.0179	0.252
$6/14$	0.118	-0.122	0.240
$6/15$	0.005	-0.238	0.243
$6/16$	-0.102	-0.337	0.235
$6/17$	-0.172	-0.422	0.250
$6/18$	-0.266	-0.495	0.229

example in Sec. 3.4 (c.f. Table 3.2). This is mainly because we allow an extra parallel edge and a high-degree punctured node in the protograph. The actual performance of a finite-length code remains to be identified. The following section gives an example of constructing long-blocklength PN-PBRL codes and presents the simulation results.

3.7 Examples and Simulations for Long-Blocklength PN-PBRL Codes

This section provides an example of a long-blocklength PN-PBRL code and presents the simulation results. The protograph used for our example is that of Fig. 3.10 shown in Sec. 3.6.

As shown in [DDJ09], a two-stage lifting process is necessary to obtain good protograph codes with long blocklengths. The first stage, also known as pre-lifting, uses a relatively small

Table 3.5: Thresholds of the PN-PBRL Codes shown in Fig. 1. (E_b/N_0 in decibels).

Rate	Threshold	Capacity	Gap
$6/7$	3.077	2.625	0.452
$6/8$	1.956	1.626	0.330
$6/9$	1.392	1.059	0.333
$6/10$	1.078	0.679	0.399
$6/11$	0.798	0.401	0.397
$6/12$	0.484	0.187	0.297
$6/13$	0.338	0.018	0.320
$6/14$	0.144	-0.122	0.266
$6/15$	0.072	-0.238	0.310
$6/16$	0.030	-0.337	0.367
$6/17$	-0.024	-0.422	0.398
$6/18$	-0.150	-0.495	0.345

lifting number (i.e. the number of replicas) and aims to remove the parallel edges in the protograph. The second stage then lifts the protograph resulting from the previous stage to the desired blocklength. Suppose that the code is pre-lifted 4 times and then further lifted 682 times, the resulting information blocklength is then $6 \times 4 \times 682 = 16368$.

With a step size of $4 \times 682 = 2728$, subsequent code rates $6/8, 6/9, \ldots, 6/18$ are obtained by transmitting the output symbols of the LT code from each successive group of variable nodes starting from the top. The corresponding thresholds of each code rate are summarized in Table 3.5. All possible rates from $1/3$ to $6/7$ with a resolution of 1 bit are indeed feasible by adding one variable node at a time.

The lifting process of the protograph is accomplished by circulant permutation of each edge, which allows efficient hardware implementation of the decoder. The design of the circulant permutation is based on circulant PEG algorithm. The minimum loop size of the precode part is

10 while the minimum loop size of the LT code part is 8.

The pre-lifted protograph of the PN-PBRL code is described as follows: Let H_p be the parity check matrix of the precode and H_{LT} be the parity matrix of the LT code part excluding the degree-one variable nodes. Let σ be a 4×4 identity matrix shifted to the left by 1 and let 0 represent the 4×4 all-zero matrix. Let I is the identity matrix and 0 is the all-zero matrix with proper dimensions. The parity-check matrix of the precode is given by

$$H = \begin{bmatrix} H_p & 0 \\ H_{LT} & I \end{bmatrix} \tag{3.10}$$

where H_p and H_{LT} are given in equations (3.11) and (3.12). Entries with multiple terms of σ indicate parallel edges in the protograph.

$$H_p = \begin{bmatrix} \sigma^0+\sigma^1 & \sigma^3 & \sigma^1+\sigma^0 & \sigma^2 & \sigma^0+\sigma^1 & \sigma^0 & \sigma^2+\sigma^0 & \sigma^2 \\ \sigma^0 & \sigma^0+\sigma^1 & \sigma^0 & \sigma^0+\sigma^1 & \sigma^0 & \sigma^0+\sigma^1 & \sigma^0 & \sigma^0+\sigma^1 \end{bmatrix}. \tag{3.11}$$

$$H_{LT} = \begin{bmatrix} \sigma^3+\sigma^0 & 0 & \sigma^0 & 0 & 0 & 0 & 0 & 0 \\ \sigma^1+\sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 \\ \sigma^2 & 0 & \sigma^0 & \sigma^0 & \sigma^0 & 0 & \sigma^0 & \sigma^0 \\ \sigma^3 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & \sigma^0 \\ \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 \\ \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & 0 & \sigma^0 \\ \sigma^2 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & \sigma^2 \\ \sigma^0 & 0 & \sigma^0 & \sigma^0 & 0 & 0 & 0 & \sigma^0 \\ \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 \\ \sigma^1 & 0 & \sigma^0 & 0 & 0 & \sigma^0 & 0 & \sigma^0 \\ \sigma^0 & 0 & \sigma^0 & 0 & \sigma^0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.12}$$

The pre-lifted protograph contains no parallel edges and is further lifted by 682 in a similar fashion but using a larger matrix Σ: a 682×682 identity matrix shifted to the left by 1. The powers of the matrix Σ, i.e. the assignments of the circulant for all edges in the pre-lifted protograph are provided along with this dissertation as a supplemental file.

All simulations use BP decoding with flooding in this section. The iterative procedure will terminate if the decoding is successful or if it reaches the maximum number of iterations. The

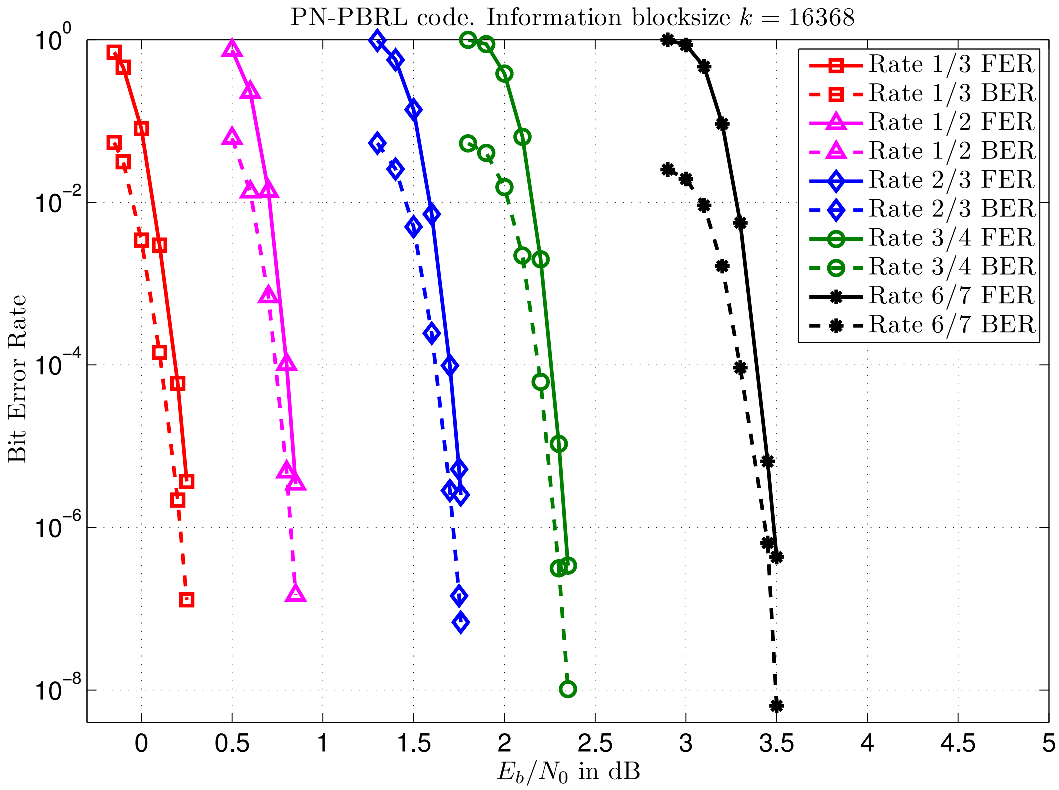

Figure 3.11: Frame error rates and bit error rates for codes in the PN-PBRL example.

maximum iteration is 100 for all simulations if not otherwise stated. Simulation results with 50 iterations are also presented in the following for reader's convenience to compare with the DVB-S2 standard [DVB09].

Fig. 3.11 shows the simulations of the example PN-PBRL code family. The information block size is $k = 16368$ and the simulated codes have rates $6/7, 3/4, 2/3, 1/2$ and $1/3$, respectively. The blocklengths are 19096, 21824, 24552, 32736 and 49104, respectively. At a fixed frame error rate of 10^{-5}, the estimated gaps of these codes to the BI-AWGN capacity are summarized in Table 3.6. The gaps range from 0.643 to 0.765 bits. Consistent with the threshold results in Table 3.5, these PN-PBRL codes have capacity-approaching performance while providing extensive rate-compatibility.

Fig. 3.12 presents the frame error rates for DVB-S2 LDPC code, DVB-S2 LDPC code concatenated with an outer BCH code [DVB09], AR4JA codes (in CCSDS standard [CCS07]) and

Table 3.6: SNRs Required to Achieve FER 10^{-5} for Code Example 1.

Rate	Required SNR	Capacity	Gap to capacity
$6/7$	3.39	2.625	0.765
$6/8$	2.30	1.626	0.674
$6/9$	1.74	1.059	0.681
$6/12$	0.83	0.187	0.643
$6/18$	0.23	-0.495	0.725

PN-PBRL codes at rate $1/2$ and rate $2/3$. Note that the maximum iteration for LDPC+BCH is 50. The blocklengths of the DVB-S2 codes are fixed to 64800 bits where the PN-PBRL and AR4JA codes have a fixed information length of 16368 bits and blocklengths 32736 bits and 24552 bits for rate $1/2$ and rate $2/3$, respectively. The overall rates of DVB-S2 codes after concatenation are 0.497 bits and 0.664 bits.

Fig. 3.12 shows that in the waterfall region, the PN-PBRL codes outperform the best known AR4JA codes [CCS07] up to the highest SNR we simulated. Fig. 3.12 also shows the performance of the DVB-S2 LDPC codes concatenated with BCH codes (with maximum 50 iterations). With shorter blocklengths and higher rates, the PN-PBRL codes still outperform the concatenated codes in the waterfall region. The error floor performance of PN-PBRL codes needs to be simulated by FPGA and is left as potential future work.

3.8 Concluding Remarks

This chapter proposes a class of rate-compatible LDPC codes and provides a systematic procedure of constructing practical codes. Motivated by the excellent performance of the repeated IR-NTC scheme in the previous chapter, the first part of this chapter focuses on the construction of PBRL codes with short blocklengths.

Optimization of PBRL codes is based on asymptotic results of LDPC codes, i.e., density

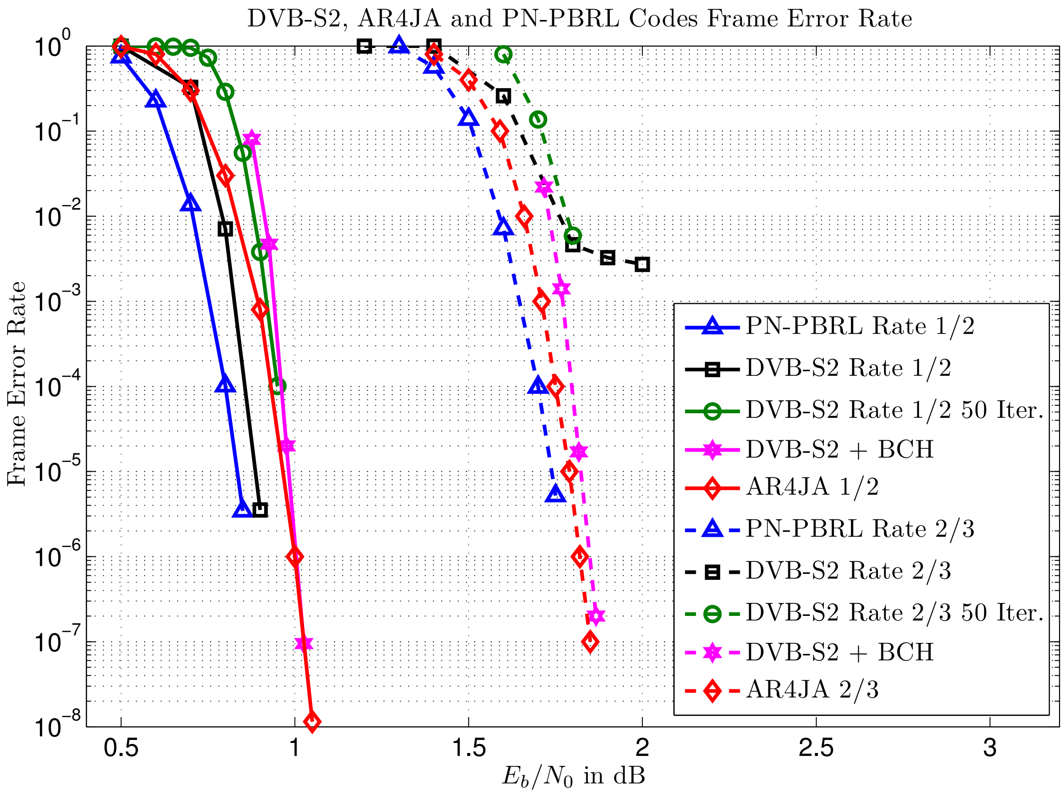

Figure 3.12: Frame error rates comparison for PN-PBRL codes, LDPC codes in the DVB-S2 standard, and AR4JA LDPC codes in the CCSDS standard.

evolution. Instead of the original density evolution, the reciprocal channel approximation is used to obtain a fast and accurate approximation for the thresholds of PBRL codes. The assignment of the circulants is based on circulant PEG algorithm. The simulation results show that although we are operating in a short blocklength regime, optimization using density evolution still provides useful guidance to improve the performance of the PBRL code.

To further improve the performance, this chapter introduces a modification of PBRL codes by puncturing variable nodes in the precode protograph, referred to as PN-PBRL codes. PN-PBRL codes have better performance but a slightly more complicated encoder for the initial transmission. These short-blocklength PN-PBRL codes outperform the RCPT codes in the 3GPP-LTE standard [Gen08] at high SNRs and do not have error floors up to the highest SNRs studied. The short-blocklength PN-PBRL codes perform worse than the RCPT codes at low SNRs.

A threshold saturation is observed as the rate decreases in the optimization of PBRL codes. Adding more parallel edges in the LT code part of the PN-PBRL protograph alleviates the saturation issue. The lifted codes with short blocklengths, however, do not perform better than the codes considered in Sec. 3.5. These low thresholds motivated us to extend the PN-PBRL code design for long blocklengths.

The final part of this chapter discussed the optimization of PN-PBRL codes with long blocklengths. The excellent threshold results of PN-PBRL codes manifest in the long-blocklength regime as shown in Sec. 3.7. The long-blocklength codes shown in Sec. 3.7 demonstrate excellent capacity-approaching performance and do not have error floors down to FER as low as 10^{-7}. The gaps are all within 0.8 dB to the BI-AWGN capacity across a variety of rates.

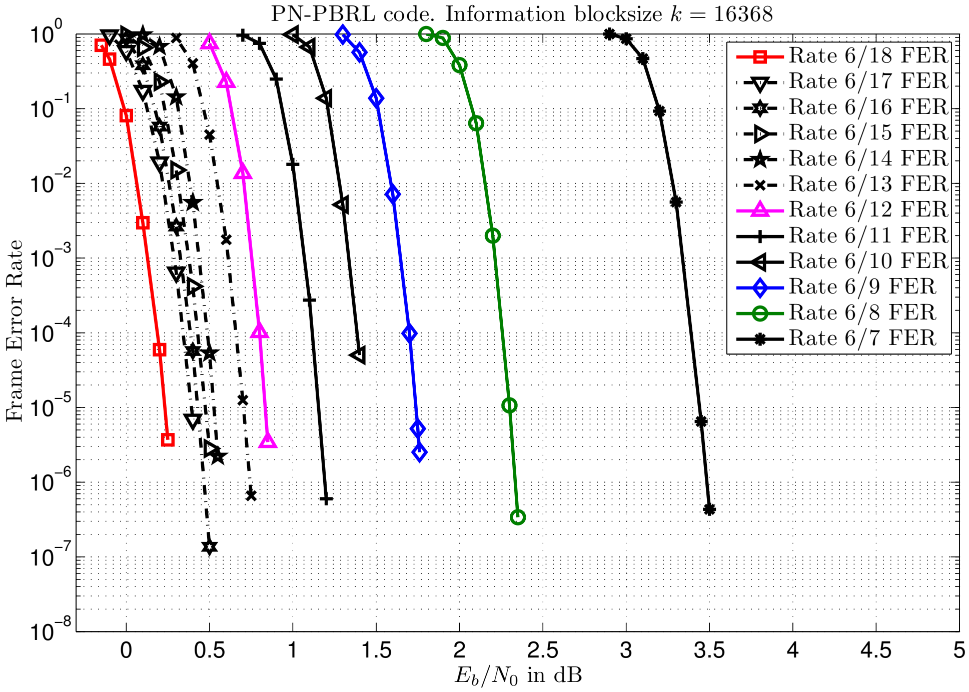

Figure 3.13: Frame error rates for PN-PBRL codes with rates $6/7, 6/8, \ldots, 6/18$.

3.A Additional Simulation Results for Long-Blocklenth PBRL Codes

This appendix presents additional simulation results for long-blocklength PN-PBRL codes. The same codes as in Sec. 3.7 are used. Fig. 3.13 shows simulations for all the possible code rates associated with the constructed protograph : $6/7, /6/8, 6/9, \ldots, 6/18$. As shown in Fig. 3.13 the waterfall performance for each rate is outstanding and as the rate decreases the frame error rate improves. Error floor for each rate has not yet to be observed up to the highest SNR simulated. A diminishing return is observed in the lower rate regime. Perhaps using an even larger protograph could solve this issue.

Fig. 3.14 compares the PN-PBRL codes to the LDPC codes in the DVB-S2 standard. Note that the blocklengths of the DVB-S2 codes are fixed to 64800 whereas the PN-PBRL codes have blocklength 21824, 24552, 32736 and 49104 for rates $3/4, 2/3, 1/2$ and $1/3$, respectively. Although having a set of blocklengths that are shorter than the DVB-S2 codes, PN-PBRL codes still have comparable (often better) performance. Take the rate $1/2$ PN-PBRL code for example:

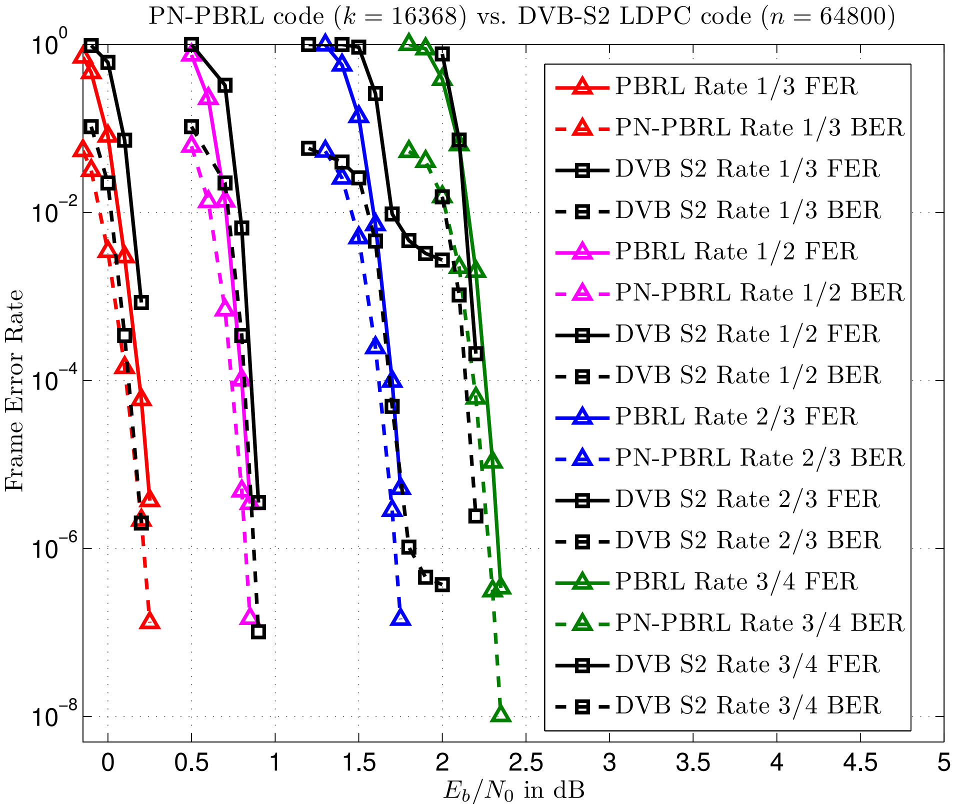

Figure 3.14: Frame error rates and bit error rates for both DVB-S2 (LDPC codes only) and PN-PBRL codes.

with almost half of the blocklength, the PN-PBRL still out-performs the DVB-S2 code.

CHAPTER 4

Conclusion

4.1 Summary of the Dissertation

The first part of this dissertation studied the benefit of noiseless feedback from both the perspective of information theory and that of practical system design. We first categorized the possible assumptions for feedback systems and investigated a particular class of incremental redundancy systems that assumes noiseless transmitter termination, the IR-NTC system. We analyzed the IR-NTC system with the constraint of finite-length codewords and limited decoding times using random coding. As a function of expected latency ℓ, we proved the necessary growth rates of the finite blocklength N and the decoding interval I to obtain asymptotic optimality. These theoretical results provide practical guidance to system design.

This dissertation also proposed a novel approximation method, rate-compatible sphere-packing approximation, providing useful guidance to practical system design. For repeated IR-NTC with limited number of transmissions m, RCSP shows excellent agreement with our simulation results for a family of tail-biting rate-compatible convolutional codes. For expected latencies larger than 200, the marginal RCSP approximation with BD decoding becomes an accurate estimate of the performance for IR-NTC, allowing efficient optimization of IR-NTC. This dissertation also demonstrated simulations of IR-NTC exceeding the corresponding random coding lower bounds.

The second part of this dissertation proposed a new class of rate-compatible LDPC codes called PBRL codes. We provided the construction and optimization methods in both the short-blocklength regime and the long-blocklength regime. We showed several examples of constructing PBRL codes and provided simulation results, demonstrating the excellent performance of these codes.

4.2 Future Direction

This dissertation studied single-user communication with noiseless feedback over memoryless channels, emphasizing on the non-asymptotic analysis. This dissertation then proposed a new class of rate-compatible LDPC codes that has excellent performance at a wide range of blocklengths and SNRs. We briefly discuss the some potential research directions to conclude this dissertation.

The non-asymptotic analysis for noiseless feedback in this dissertation focuses on single-user communication. It would be very interesting to study whether the same conclusion for the single-user case would hold for more than one users. Many problems in network information theory remain unsolved even without the complication of an additional feedback channel. However, it would be interesting to study the non-asymptotic behavior for the networks with known capacity region, e.g. the degraded broadcast channels.

In the context of single-user communication, it is known that the optimal error exponent can be achieved by two extremely different approaches: Burnashev scheme and Yamamoto-Itoh scheme. Burnashev used an extremely active approach that utilizes feedback to adjust the transmissions at every time instance, while Yamamoto and Itoh only used feedback to perform error detection and confirmation. One would suspect that a highly active encoder would imply a higher expected throughput. An interesting question to ask is whether it is possible to characterize the performance of a system such that the resulting metric is monotonically increasing as the activeness of a system increases.

For the design of PBRL codes, we comment that in the short-blocklength regime, the asymptotic analysis of the PBRL codes might not be as accurate as for the long-blocklength codes. Finding a better criterion for designing a good short blocklength code is an interesting direction for future research. It is also observed that the low-SNR performance of these short-blocklength codes are not performing well in the waterfall region compared to turbo codes. Perhaps extending the PBRL codes to non-binary PBRL codes or using a novel decoding algorithm would help mitigate the issue.

REFERENCES

[ADD04] K. Andrews, S. Dolinar, D. Divsalar, and J. Thorpe. "Design of low-density parity-check (LDPC) codes for deep-space applications." *JPL IPN Progress Report*, **42-159**, Nov. 2004.

[Ana01] A. Anastasopoulos. "A comparison between the sum-product and the min-sum iterative detection algorithms based on density evolution." In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, volume 2, pp. 1021–1025, 2001.

[BF64] R. Benice and A.H. Frey. "An Analysis of Retransmission Systems." *Communication Technology, IEEE Transactions on*, **12**(4):135–145, 1964.

[BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima. "Near Shannon limit errorcorrecting coding and decoding: Turbo codes." In *Proc. IEEE Int. Conf. Commun.*, Geneva, Switzerland, May 1993.

[Bur76] M. V. Burnashev. "Data transmission over a discrete channel with feedback. Random transmission time." *Probl. Inf. Transm*, **12**(4):10–30, 1976.

[BV04] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[CCS07] "Low density parity check codes for use in near-earth and deep space. Orange Book." *CCSDS standard, Issue No.2*, Sep. 2007.

[CF98] V. Chande and N. Farvardin. "A dynamic programming approach to constrained feedback hybrid ARQ design." In *Proc. 1998 IEEE Int. Symp. Inf. Theory (ISIT)*, p. 286, Aug. 1998.

[Cha56] S. S. L. Chang. "Theory of information feedback systems." *IEEE Trans. Inf. Theory*, **PGIT-2**:29–40, Sep. 1956.

[Cha85] D. Chase. "Code Combining–A Maximum-Likelihood Decoding Approach for Combining an Arbitrary Number of Noisy Packets." *IEEE Trans. Commun.*, **33**(5):385 – 393, May. 1985.

[Chu00] S. Y. Chung. *On the construction of some capacity-approaching coding schemes*. PhD thesis, MIT, Cambridge, MA, 2000.

[CSS10] T.-Y. Chen, N. Seshadri, and B-Z. Shen. "Is feedback a performance equalizer of classic and modern codes?" In *Proc. 2010 Inf. Theory and Applications Workshop (ITA)*, San Diego, CA, USA, Feb. 2010.

[CSW11] T.-Y. Chen, N. Seshadri, and R. D. Wesel. "A sphere-packing analysis of incremental redundancy scheme with feedback." In *Proc. IEEE Intl. Conf. Comm.*, Kyoto, Japan, Jun. 2011.

[Dav72] George I. Davida. “An Analysis of Retransmission Systems.” *Information and Control, Elsevier*, **21**:117–133, 1972.

[DDJ09] D. Divsalar, S. Donlinar, C. R. Jones, and Kenneth Andrews. “Capacity-approaching protograph codes.” *IEEE J. Sel. Areas Commun.*, **27, No. 6**:876–888, Aug. 2009.

[Dol05] S. Dolinar. “A rate-compatible family of protograph-based LDPC codes built by expurgation and lengthening.” In *Proc. International Symposium on Information Theory*, pp. 1627–1631, 2005.

[DVB09] “Digital Video Broadcasting; Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications.” *ETSI EN 302 307 V1.2.1*, Aug. 2009.

[EHB09] M. El-Khamy, Jilei Hou, and N. Bhushan. “Design of rate-compatible structured LDPC codes for hybrid ARQ applications.” *Selected Areas in Communications, IEEE Journal on*, **27**(6):965–973, 2009.

[Eli56] P. Elias. “Channel Capacity without Coding.” In *MIT Res. Lab of Electronics, Quarterly Progress Report, no. 43, VIII (http://hdl.handle.net/1721.1/51989)*, Cambridge, MA, USA, Oct. 1956.

[ES06] O. Etesami and A. Shokrollahi. “Raptor codes on binary memoryless symmetric channels.” *Information Theory, IEEE Transactions on*, **52**(5):2033–2051, 2006.

[FH09] J.C. Fricke and P.A. Hoeher. “Reliability-based retransmission criteria for hybrid ARQ.” *IEEE Trans. Commun.*, **57**(8):2181–2184, Aug. 2009.

[For68] Jr. Forney, G.D. “Exponential error bounds for erasure, list, and decision feedback schemes.” *IEEE Trans. Inf. Theory*, **14**(2):206–220, 1968.

[FS04] J. Freudenberger and B. Stender. “An algorithm for detecting unreliable code sequence segments and its applications.” *IEEE Trans. Commun.*, **52**(11):1833–1839, Nov. 2004.

[Gal63] R. G. Gallager. *Low-Density Parity-Check Codes*. PhD thesis, MIT, Cambridge, MA, 1963.

[Gal10] R. G. Gallager. “Variations on a theme by Schalkwijk and Kailath.” *IEEE Trans. Inf. Theory*, **56**(1):6–17, January 2010.

[Gen08] 3rd Generation Partnership Project (http://www.3gpp.org). “3GPP TS 36.212 Multiplexing and channel coding.” **Release 8**, Mar. 2008.

[Hag88] J. Hagenauer. “Rate-compatible punctured convolutional codes (RCPC codes) and their applications.” *IEEE Trans. Commun.*, **36**(4):389–400, Apr. 1988.

[HEA05] X.-Y. Hu, E. Eleftheriou, and D.-M. Arnold. “Regular and irregular progressive edge-growth tanner graphs.” *IEEE Trans. Inf. Theory*, **51, No. 1**:386–398, Jan. 2005.

[HKK06] Jeongseok Ha, Jaehong Kim, D. Klinc, and S.W. McLaughlin. “Rate-compatible punctured low-density parity-check codes with short block lengths.” *Information Theory, IEEE Transactions on*, **52**(2):728–738, 2006.

[HKM04] Jeongseok Ha, Jaehong Kim, and S.W. McLaughlin. “Rate-compatible puncturing of low-density parity-check codes.” *IEEE Trans. Inf. Theory*, **50**(11):2824–2836, 2004.

[JS07] N Jacobsen and R. Soni. “Deign of rate-compatible irregular LDPC codes based on edge growth and parity splitting.” In *Proc. IEEE Vehicular Technology Conference (VTC)*, Baltimore, MD, USA, Oct. 2007.

[KFL01] F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. “Factor graphs and the sum-product algorithm.” *Information Theory, IEEE Transactions on*, **47**(2):498–519, 2001.

[Kra69] A. Kramer. “Improving communication reliability by use of an intermittent feedback channel.” *IEEE Trans. Inf. Theory*, **IT-15**(1):52–60, Jan. 1969.

[KRM09] Jaehong Kim, A. Ramamoorthy, and S.W. McLaughlin. “The design of efficiently-encodable rate-compatible LDPC codes.” *IEEE Trans. Commun.*, **57**(2):365–375, 2009.

[LC04] Shu Lin and Daniel J. Costello. *Error Control Coding*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2004.

[LMS01] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman. “Improved low-density parity-check codes using irregular graphs.” *IEEE Trans. Inform. Theory*, **47**:585–598, Feb. 2001.

[LN02] J. Li and K.R. Narayanan. “Rate-compatible low density parity check codes for capacity-approaching ARQ schemes in packet data communications.” In *Proc. Int. Conf. on Comm., Internet, and Info. Tech. (CIIT)*, Nov. 2002.

[LSS03] Ruoheng Liu, P. Spasojevic, and E. Soijanin. “Punctured turbo code ensembles.” In *Information Theory Workshop, 2003. Proceedings. 2003 IEEE*, pp. 249–252, 2003.

[Lub02] M. Luby. “LT codes.” In *Proc. 43rd Annu. IEEE Symp. Foundations of Computer Science (FOCS)*, Vancouver, BC, Canada, November 2002.

[LY82] Shu Lin and P.S. Yu. “A Hybrid ARQ Scheme with Parity Retransmission for Error Control of Satellite Channels.” *Communications, IEEE Transactions on*, **30**(7):1701–1719, 1982.

[Mac99] D. J. C. MacKay. “Good error-correcting codes based on very sparse matrices.” *IEEE Trans. Inform. Theory*, **45**:399–431, Mar. 1999.

[Man74] D.M. Mandelbaum. “An adaptive-feedback coding scheme using incremental redundancy (Corresp.).” *IEEE Trans. Inf. Theory*, **20**(3):388–389, 1974.

[MMC98] R.J. McEliece, D. J C MacKay, and Jung-Fu Cheng. “Turbo decoding as an instance of Pearl’s belief propagation algorithm.” *IEEE J. Sel. Areas Commun.*, **16**(2):140–152, 1998.

[MW86] H. Ma and J. Wolf. “On Tail Biting Convolutional Codes.” *IEEE Trans. Commun.*, **34**(2):104 – 111, Feb. 1986.

[NG08] B. Nakiboğlu and R.G. Gallager. “Error Exponents for Variable-Length Block Codes With Feedback and Cost Constraints.” *IEEE Trans. Inf. Theory*, **54**(3):945 –963, Mar. 2008.

[NJ12] Mohammad Naghshvar and Tara Javidi. “Active Sequential Hypothesis Testing.” *arXiv:1203.4626v3 [cs.IT]*, Oct. 2012.

[NND12] T.V. Nguyen, A. Nosratinia, and D. Divsalar. “The Design of Rate-Compatible Protograph LDPC Codes.” *IEEE Trans. Commun.*, **60**(10):2841–2850, 2012.

[PPV10] Y. Polyanskiy, H.V. Poor, and S. Verdú. “Channel Coding Rate in the Finite Blocklength Regime.” *IEEE Trans. Inf. Theory*, **56**(5):2307 –2359, May. 2010.

[PPV11] Y. Polyanskiy, H. V. Poor, and S. Verdú. “Feedback in the Non-Asymptotic Regime.” *IEEE Trans. Inf. Theory*, **57**(8):4903–4925, Aug. 2011.

[PV08] D.P. Palomar and S. Verdú. “Lautum Information.” *IEEE Trans. Inf. Theory*, **54**(3):964 –975, Mar. 2008.

[RM00] D.N. Rowitch and L.B. Milstein. “On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo (RCPT) codes.” *Communications, IEEE Transactions on*, **48**(6):948–959, 2000.

[RSU01] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke. “Design of capacity-approaching irregular low-density parity-check codes.” *IEEE Trans. Inf. Theory*, **47**(2):618–637, Feb 2001.

[RU01] T. J. Richardson and R. L. Urbanke. “The capacity of low-density parity-check codes under message-passing decoding.” *IEEE Trans. Inf. Theory*, **47**(2):599–618, Feb. 2001.

[RU08] Tom Richardson and Ruediger Urbanke. *Modern Coding Theory*. Cambridge University Press, New York, NY, USA, Mar. 2008.

[Sch66] J. Schalkwijk. “A coding scheme for additive noise channel with feedback–II: Bandlimited signals.” *IEEE Trans. Inf. Theory*, **IT-12**(2):183–189, Apr. 1966.

[SCV04] S. Sesia, G. Caire, and G. Vivier. "Incremental redundancy hybrid ARQ schemes based on low-density parity-check codes." *IEEE Trans. Commun.*, **52**(8):1311–1321, 2004.

[SGB67] C. E. Shannon, R. G. Gallager, and E. R. Berlekamp. "Lower Bounds to Error Probability for Coding on Discrete Memoryless Channels I." *Inf. Contr.*, **10**:65–103, 1967.

[Sha56] C. E. Shannon. "The zero error capacity of a noisy channel." *IEEE Trans. Inf. Theory*, **PGIT-2**:8–19, Sep. 1956.

[Sha59] C. E. Shannon. "Probability of error for optimal codes in a Gaussian channel." *Bell Syst. Tech. J.*, **PGIT-2**:611–656, 1959.

[Sho06] A. Shokrollahi. "Raptor codes." *IEEE Trans. Inf. Theory*, **52**(6):2551–2567, 2006.

[SHS08] Seungmoon Song, Daesung Hwang, Sunglock Seo, and Jeongseok Ha. "Linear-Time Encodable Rate-Compatible Punctured LDPC Codes with Low Error Floors." In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pp. 749–753, 2008.

[SK66] J. Schalkwijk and T. Kailath. "A coding scheme for additive noise channel with feedback–I: No bandwidth constraint." *IEEE Trans. Inf. Theory*, **IT-12**(2):172–182, Apr. 1966.

[Sol06] E. Soljanin. "Punctured vs. rateless codes for hybrid ARQ." In *Proc. IEEE Inform. Theory Workshop '06*, Punta del Este, Uruguay, Mar 2006.

[Tan81] R.M. Tanner. "A recursive approach to low complexity codes." *Information Theory, IEEE Transactions on*, **27**(5):533–547, 1981.

[Tho03] J. Thorpe. "Low Density Parity Check (LDPC) codes constructed from protographs." *JPL IPN Progress Report*, **42–154**, August 2003.

[URG03] E. Uhlemann, L.K. Rasmussen, A. Grant, and P.-A. Wiberg. "Optimal Incremental-redundancy strategy for type-II hybrid ARQ." In *Proc. 2003 IEEE Int. Symp. Inf. Theory (ISIT)*, p. 448, July 2003.

[Van43] H. C. A. Van Duuren. "Printing Telegraph Systems.", Mar 1943.

[VF09] B.N. Vellambi and F. Fekri. "Finite-length rate-compatible LDPC codes: a novel puncturing scheme." *Communications, IEEE Transactions on*, **57**(2):297–301, 2009.

[VGW10] A.I. Vila Casado, M. Griot, and R.D. Wesel. "LDPC Decoders with Informed Dynamic Scheduling." *Communications, IEEE Transactions on*, **58**(12):3470–3479, 2010.

[VS10] S. Verdú and S. Shamai. "Variable-Rate Channel Capacity." *IEEE Trans. Inf. Theory*, **56**(6):2651 –2667, Jun. 2010.

[VST05] E. Visotsky, Yakun Sun, V. Tripathi, M.L. Honig, and R. Peterson. "Reliability-based incremental redundancy with convolutional codes." *IEEE Trans. Commun.*, **53**(6):987– 997, June 2005.

[VTH03] E. Visotsky, V. Tripathi, and M. Honig. "Optimum ARQ design: a dynamic programming approach." In *Proc. 2003 IEEE Int. Symp. Inf. Theory (ISIT)*, p. 451, July 2003.

[WCW12] A. R. Williamson, T.-Y. Chen, and R. D. Wesel. "A Rate-Compatible Sphere-Packing Analysis of Feedback Coding with Limited Retransmissions." In *Proc. 2012 IEEE Int. Symp. Inf. Theory (ISIT)*, Cambridge, MA, USA, Jul. 2012.

[YB04] M.R. Yazdani and A.H. Banihashemi. "On construction of rate-compatible low-density Parity-check codes." *Communications Letters, IEEE*, **8**(3):159–161, 2004.

[YI79] H. Yamamoto and K. Itoh. "Asymptotic performance of a modified Schalkwijk-Barron scheme for channels with noiseless feedback." *IEEE Trans. Inf. Theory*, **25**:729–733, Nov. 1979.

[Zig70] K. Sh. Zigangirov. "Upper bounds for the error probability for channels with feedback." *Probl. Pered. Inform.*, **6**(1):87–92, 1970.