

Channel Coding for Anytime Transmission

Md. Noor-A-Rahim

B.Sc. (Honours)

Thesis submitted for the degree of
Doctor of Philosophy
in
Telecommunications



November 2015

Contents

List of Acronyms	vi
List of Symbols	ix
List of Figures	x
List of Tables	xiii
Abstract	xvi
Publications	xviii
Declaration	xix
Acknowledgements	xxi
1 Introduction	1
2 Background	7
2.1 Channel Models	7
2.1.1 Point-to-Point Communication Channel	7
2.1.2 Relay Channel	9
2.1.3 Multiple Access Channel	10
2.2 Low-Density Parity-Check Codes	12
2.2.1 LDPC Convolutional Codes	15
2.2.2 Spatially Coupled Codes	16
2.2.3 Existing Anytime Channel Code	18
2.2.4 Bilayer Expurgated LDPC codes	19
2.3 Decoding Algorithms	20
2.3.1 Erasure Decoding	20
2.3.2 Belief Propagation Decoding	20
2.4 Density Evolution	22
2.4.1 Over BEC	22
2.4.2 Over BIAWGN Channel	23
3 Tools for Anytime Code	27

3.1	Performance Estimation of Finite-Length LDPC Codes	27
3.1.1	Proposed Algorithm	29
3.1.2	Over BEC	30
3.1.3	Over BIAWGN Channel	31
3.1.4	Discussions	32
3.2	Density Evolution of SC-LDPC Codes over BIAWGN Channel . . .	32
3.2.1	Multi Edge Type Density Evolution	35
3.2.2	MET Density Evolution with Approximation	36
3.2.3	Proposed Density Evolution Analysis	38
3.2.4	Results and Discussion	39
3.3	SC-LDPC Code Design for Gaussian MAC	41
3.3.1	Code Design	42
3.3.2	Decoding Algorithm	42
3.3.3	Density Evolution	43
3.3.4	Optimization	45
3.3.5	Numerical Results	46
3.4	Conclusion	49
4	Anytime Codes for Point-to-Point Communication	51
4.1	System Model	51
4.2	Spatially Coupled Codes as Anytime Codes	53
4.3	Code performance over BEC	54
4.3.1	Asymptotic Analysis	54
4.3.2	Finite-Length Estimation	57
4.3.3	Adaptive Window Decoding	58
4.3.4	Numerical Results	61
4.4	Extension to the BIAWGN Channel Scenario	65
4.4.1	Asymptotic Analysis	65
4.4.2	Performance of Finite-Length Codes	66
4.4.3	Adaptive Window Decoding	66
4.5	Conclusion	72
5	Anytime Codes for Relay Channels	73
5.1	System Model	73
5.2	Proposed Bilayer Anytime SC-LDPC Code	74
5.3	Coded Relaying Scheme	76
5.3.1	Density Evolution	76
5.3.2	Delay-Exponent Analysis	78
5.4	Uncoded Relaying Scheme	81
5.4.1	Density Evolution	81
5.4.2	Delay-Exponent Analysis	83
5.4.3	Modified design	84
5.5	Conclusion	89

6	Anytime Codes for Multiple Access Channels	91
6.1	System Model	91
6.2	Proposed Anytime Code Over MAC	92
6.3	Over Erasure Multiple Access Channel	92
6.3.1	Density Evolution	93
6.3.2	Delay-Exponent Analysis	95
6.4	Over Gaussian Multiple Access Channel	97
6.5	Finite-Length Performance	98
6.6	Conclusion	100
7	Conclusions	101
7.1	Future Research	102
	Appendices	105
A	Proof of Theorems	107
A.1	Proof of Theorem 4.1	107
A.2	Proof of Theorem 5.1	109
A.3	Proof of Theorem 5.2	112
A.4	Proof of Theorem 6.1	114

List of Acronyms

AWD	adaptive window decoding.
BE-LDPC	bilayer expurgated low-density parity-check.
BEC	binary erasure channel.
BIAWGN	binary input additive white Gaussian noise.
BMS	binary-input memoryless symmetric-output.
BP	belief propagation.
BPSK	binary phase shift keying.
CF	compress-and-forward.
CN	check node.
DF	decode-and-forward.
EWD	expanding window decoding.
GA	Gaussian approximation.
GMAC	Gaussian multiple access channel.
LDPC	low-density parity-check.
LDPCC	LDPC convolutional.
LLRs	log likelihood ratios.
MAC	multiple access channel.

MAP	maximum-a-posteriori.
P2P	point-to-point.
pdfs	probability density functions.
RCA	reciprocal-channel approximation.
SC-LDPC	spatially coupled low-density parity-check.
UEP	unequal error protection.
VN	variable node.

List of Symbols

α	delay-exponent, page 52
$C_B(g)$	capacity of the BIAWGN channel with SNR g , page 24
$C_{\text{BEC}}(\epsilon)$	capacity of BEC with erasure probability ϵ , page 8
$C_{\text{BIAWGN}}(\sigma_n)$	capacity of the BIAWGN channel with variance σ_n^2 , page 9
$C_{\text{T}}(\sigma_n)$	capacity of the ternary input AWGN channel with variance σ_n^2 , page 44
ℓ	iteration number, page 21
ϵ	channel erasure probability, page 8
ϵ^*	decoding threshold over BEC, page 23
ν_i	fraction of edges that are connected to degree i variable nodes, page 14
γ	coupling length, page 17
Γ_i	fraction of check nodes of degree i , page 14
Λ_i	fraction of variable nodes of degree i , page 14
$\mathbb{E}_c(i)$	set of edge types that share the check node with an edge of type i , page 37
$\mathbb{E}_v(i)$	set of edge types that share the variable node with an edge of type i , page 37
$\mathbb{N}$	set of natural numbers, page 17
$\mathbb{S}_c(i)$	set of check nodes that are connected with i^{th} variable node, page 21

$\mathbb{S}_v(i)$	set of variable nodes that are connected with i^{th} check node, page 21
$\mathbb{U}$	input alphabet, page 8
$\mathbb{V}$	output alphabet, page 8
$\mathcal{C}$	an error correction code, page 12
$\mathcal{D}_t(\cdot)$	anytime decoder, page 52
$\mathcal{E}_t(\cdot)$	anytime encoder, page 52
$\mathcal{W}_{av}$	average decoding window length of AWD, page 60
m_s	syndrome former memory, page 16
J	number of check node, page 13
ρ_i	fraction of edges that are connected to degree i check nodes, page 14
σ_n^*	decoding threshold over BIAWGN channel, page 24
σ_n^2	noise variance
$\mathbf{p}$	vector of parity components, page 59
$\mathbf{s}$	vector of syndrome components, page 68
$\mathbf{B}$	base matrix, page 16
$\mathbf{c}$	codeword, page 14
$\mathbf{H}$	parity check matrix, page 14
$\mathbf{m}_t$	message at time t , page 52
d	delay, page 52
d_c	degree of check node, page 14
d_v	degree of variable node, page 14
$E[\cdot]$	expectation operator, page 2
E_b	energy per bit, page 9
$I(\cdot; \cdot)$	mutual information, page 10

L	chain length, page 16
N	block length, page 12
N_0	noise power spectral density, page 9
P_c	error probability realized from BIAWGN channel, page 30
P_e	error probability, page 30
$P_e(i, t)$	error probability of message i at time t , page 2
$P_{\text{ch}}(\cdot)$	transition probability, page 8
R	code rate, page 14
r	LLR obtained from the channel, page 21
t	decoding time instant, page 52
u	channel input, page 8
v	channel output, page 8
$x^{(\ell)}$	outgoing message from variable node at iteration ℓ , page 23
$y^{(\ell)}$	outgoing message from check node at iteration ℓ , page 23
z	noise realization at receiver, page 9

List of Figures

2.1	Point-to-point communication model	7
2.2	Binary erasure channel	8
2.3	Binary input additive white Gaussian noise channel	9
2.4	Three-terminal relay channel	9
2.5	2-user multiple access channel	10
2.6	An adder channel followed by an erasure channel	11
2.7	Gaussian multiple access channel	12
2.8	Tanner graph of an LDPC code	13
2.9	Spatially coupled LDPC code	17
2.10	Tanner graph of bilayer expurgated LDPC code	19
3.1	Finite-length performance estimation over BEC	31
3.2	Finite-length performance estimation of regular LDPC codes over BIAWGN channel	33
3.3	Finite-length performance estimation of irregular LDPC codes over BIAWGN channel	34
3.4	MET representation of SC-LDPC code	35
3.5	Decoding threshold prediction with RCA technique	40
3.6	Decoding threshold prediction with GA technique	40
3.7	Factor graph of SC-LDPC code structure for 2-user MAC	43
3.8	Bit error rate performance of the SC-LDPC GMAC code	48
4.1	Block diagram illustrating anytime communication	52
4.2	Asymptotic performance of proposed anytime code over BEC	55
4.3	Comparison between the theoretical exponent and the numerical exponent	56
4.4	Comparison between simulation and estimation results over BEC	58
4.5	Illustration of unterminated check matrix and active check matrix	59
4.6	Comparison between EWD and AWD over BEC	61
4.7	Performance comparison between our proposed anytime code and systematic LDPC convolutional code	62
4.8	Effect of exponential rate parameter on anytime SC-LDPC code	63
4.9	Effect of coupling length on anytime SC-LDPC code	64
4.10	Asymptotic performance of proposed anytime code over BIAWGN channel	66

4.11	Bit error rate comparison between our proposed anytime code and systematic LDPC convolutional code	67
4.12	Performance comparison between EWD and AWD over BIAWGN Channel	70
4.13	Complexity comparison between EWD and AWD over BIAWGN Channel	70
4.14	Probability density function of decoding window length using AWD-I	71
5.1	Bilayer anytime SC-LDPC code	75
5.2	Anytime performance of coded relaying scheme	77
5.3	Effect of relay's delay d_r on the decaying behaviour of erasure probability for coded relaying scheme.	79
5.4	Performances of coded relaying scheme at fixed overall delay	79
5.5	Comparison between theoretical and numerical delay-exponents for coded relaying scheme	80
5.6	Asymptotic performance of uncoded relaying scheme.	82
5.7	Comparison between theoretical and numerical delay-exponents of uncoded relaying scheme.	84
5.8	Tanner graph of modified uncoded relaying scheme	85
5.9	Asymptotic performance comparison between the proposed coded and uncoded relaying schemes	88
5.10	Finite-length performance comparison between the proposed coded and uncoded relaying schemes	89
6.1	Factor graph of proposed anytime MAC code	93
6.2	Asymptotic performance of the proposed code over the adder followed by an erasure channel	95
6.3	Delay-exponent comparison between theoretical and numerical delay-exponents for MAC	96
6.4	Asymptotic performance of the proposed anytime MAC code over Gaussian multiple access channel	99
6.5	Finite-length performance of the proposed anytime MAC code. . . .	99

List of Tables

3.1	Asymptotic performance of SC-LDPC GMAC code	47
3.2	Threshold comparison between the optimized irregular LDPC GMAC code and SC-LDPC GMAC code	47

Abstract

Reliable streaming of information in real time is known to be a relevant problem in communication engineering. Such transmission is particularly desired for wireless automation, where real time system state information from remote observers is required to determine appropriate control actions. Traditional communications are inappropriate for control applications, since they attain reliability at the price of large delays which may result in instability in closed loop controls. Anytime transmission, which separates the design of transmission from that of controlling, is the viable solution for control application.

Feasibility of anytime transmission was first demonstrated through random tree codes. However, these anytime codes were not practically feasible. Although practical linear anytime codes were proposed recently for point-to-point communication, existing linear anytime codes provide poor anytime performance along with high decoding complexity. Moreover, practical anytime code design for relay and multi-user communication remains an open problem. This thesis focuses on the design and analysis of practical channel codes for point-to-point, relay, and multi-user anytime transmissions.

Firstly, we propose an efficient channel coding scheme for point-to-point anytime transmission. We design this coding scheme based on spatially coupled codes and demonstrate the anytime property of the proposed code through density evolution analysis. We derive analytical expression of the delay-exponent and present an algorithm to predict the finite-length performance of the proposed anytime code. We also propose adaptive window decoding techniques to ensure low complexity anytime receivers. Through numerical results, we show that our proposed code exhibits better anytime performance than the existing anytime codes.

Secondly, we study the design and analysis of anytime codes for a three-terminal relay network. We propose a bilayer anytime coding scheme for the relay network.

We analyse the delay-exponent of the code and show that it matches the numerical exponent obtained from the density evolution analysis. Through simulation results, we also show the finite-length performance of the proposed bilayer anytime code.

Thirdly, we propose anytime coding scheme for multiple access channel. The anytime characteristics of the proposed coding scheme are investigated for two different multiple access channels. We also derive the analytical delay-exponent for the proposed coding scheme.

Apart from the above contributions, we also develop analytical tools to analyse the proposed anytime codes. These tools are also applicable to analyse and design classical block codes. For classical codes, we show that these tools give better accuracy and/or less complexity than the existing tools.

Publications

Journal Papers

M. Noor-A-Rahim, K. D. Nguyen, and G. Lechner, "*Delay-Universal Codes for Multiple Access Channels*," (to appear) Transactions on Emerging Telecommunications Technologies.

M. Noor-A-Rahim, K. D. Nguyen, and G. Lechner, "*Anytime Reliability of Spatially Coupled Codes*," IEEE Transactions on Communications, vol. 63, no. 4, pp. 1069–1080, April 2015.

Conference Papers

N. Zhang, M. Noor-A-Rahim, B. N. Vellambi and K. D. Nguyen, "*Anytime Properties of Protograph-Based Repeat-Accumulate Codes*," in Proc. Information Theory Workshop (ITW), Oct. 2015, South Korea.

M. Noor-A-Rahim, K. D. Nguyen, and G. Lechner, "*Anytime Reliable Bilayer Codes with Uncoded Relaying*," in Proc. IEEE International Conference on Communications (ICC), Jun. 2015, London, UK.

N. Zhang, M. Noor-A-Rahim, B. N. Vellambi and K. D. Nguyen, "*Protograph-Based Anytime Reliable Channel Coding Design*," in Proc. IEEE International Conference on Communications (ICC), Jun. 2015, London, UK.

M. Noor-A-Rahim, K. D. Nguyen, and G. Lechner, "*Delay-Exponent of Bilayer Anytime Code*," in Proc. IEEE Information Theory Workshop (ITW), Nov. 2014, Hobart, Australia.

M. Noor-A-Rahim, K. D. Nguyen, and G. Lechner, "*Finite Length Analysis of LDPC Codes*," in Proc. IEEE Wireless Communication and Networking Conference (WCNC), Apr. 2014, Istanbul, Turkey.

M. Noor-A-Rahim, K. D. Nguyen, and G. Lechner, "*Anytime Spatially Coupled Codes for Relay Channel*," in Proc. Australian Communication Theory Workshop (AusCTW), Feb. 2014, Sydney, Australia.

M. Noor-A-Rahim, K. D. Nguyen, and G. Lechner, "*Anytime Characteristics of Spatially Coupled Code*," in Proc. 51st Annual Allerton Conference on Communication, Control, and Computing, Oct. 2013, Illinois, USA.

Submitted Manuscripts

M. Noor-A-Rahim, K. D. Nguyen, and G. Lechner, "*Design and Analysis of Anytime Codes for Relay Channels*," submitted to IEEE Transactions on Communications.

M. Noor-A-Rahim, K. D. Nguyen, and G. Lechner, "*SC-LDPC Code Design for Gaussian Multiple Access Channel*," submitted to IET Communications.

M. Noor-A-Rahim, G. Lechner, and K. D. Nguyen "Density Evolution Analysis of Spatially Coupled LDPC Codes Over BIAWGN Channel," submitted to Australian Communication Theory Workshop (AusCTW).

Declaration

I declare that this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge it does not contain any materials previously published or written by another person except where due reference is made in the text.



Md. Noor-A-Rahim

Acknowledgements

All praise, thanks, and glory to the Almighty for helping me to complete my PhD work and this thesis.

I owe my deepest gratitude to my supervisors Dr. Khoa Nguyen and Dr. Gottfried Lechner for their exemplary guidance, constant support and unceasing enthusiasm. I really appreciate my principal supervisor Dr. Khoa Nguyen for his sharp insight into the research problems and guiding me in the right directions. Khoa was always helpful and available for discussion. I would like to thank my co-supervisor Dr. Gottfried Lechner for the numerous stimulating discussions and clever suggestions throughout my PhD candidature. I acknowledge both of your significant support in producing this dissertation.

I thank the supporting staff and my Lab-mates at the Institute for Telecommunications Research (ITR) for creating friendship and learning environment. I would like to thank the University of South Australia for the financial support during my PhD study.

Last but not the least, I would like to thank my parents for their encouragement during my study. I would like to thank my wife Mahmuda for her support and understanding during my entire candidature. Thanks to my son Omar, who was a great entertainment for the last two years.

Without you all, this thesis would not have been accomplished.

Chapter 1

Introduction

After Claude Shannon published "The Mathematical Theory of Communication" [1], information theory has travelled a long way. Thanks to the enormous development of both source and channel coding, we are now able to develop high-speed communication systems. However, classical coding techniques are not suitable for specific applications. Consider a control application where the controller needs to track system state information over a noisy channel. For timely control actions, the controller needs reliable system state information in real time. In other words, both transmission reliability and delay are important in control applications. Classical codes, including block and convolutional codes, are inappropriate for such applications because block codes attain reliability only at infinite delays; while convolutional codes with finite memory-length result in non-zero error probability. Thus, classical codes are not suitable for control applications. The problem is better illustrated by the following example [2, Chapter 3].

Example: Consider an unstable Markov process described by:

$$X_{i+1} = AX_i + W_i,$$

where X_i is the Markov source with parameter $A \geq 1$ and $W_i \sim \text{Bernoulli}(0.5)$ is a binary noise process. The challenge is to design a transmission system to track the state X_i over a noisy channel. With a known initial state, tracking can be achieved by estimating the random noise W_i . Let $\hat{X}_{i+1}(t)$ be the estimation of X_{i+1} by the receiver at any time instance $t \geq i$. For $X_0 = 0$,

$$\hat{X}_{t+1}(t) = A\hat{X}_t(t) + \hat{W}_t(t)$$

$$= \sum_{i=0}^t A^{t-i} \hat{W}_i(t),$$

where $\hat{W}_i(t)$ is the estimation of W_i at time t . Then the estimated mean squared error is given by

$$\begin{aligned} E \left[\left| X_{t+1} - \hat{X}_{t+1}(t) \right|^2 \right] &= E \left[\sum_{i=0}^t A^{2(t-i)} \left| (W_i - \hat{W}_i(t)) \right|^2 \right] \\ &= \sum_{i=0}^t A^{2(t-i)} E \left[\left| (W_i - \hat{W}_i(t)) \right|^2 \right] \\ &= \sum_{i=0}^t A^{2(t-i)} P_e(i, t), \end{aligned} \tag{1.1}$$

where $P_e(i, t)$ is the error probability of symbol i at time t . It follows from (1.1) that the impact on the estimation error grows exponentially with $t - i$ due to erroneous decoding of W_i at time t . In other words, the mean squared error will grow unbounded unless the error probability in estimating W_i decays sufficiently fast with delay. Due to the finite delay requirement, classical codes exhibit a non-vanishing error probability which does not decay with delay. Thus, classical block and convolutional codes are not sufficient to track unstable Markov sources [2].

For controlling/tracking applications, tree codes were introduced in [3]. Later, the construction of error correcting code for automatic control of unstable process was proposed in [4]. However, the codes shown in [3, 4] are non-linear codes, leading to impractically complex code design and implementation. In addition, dynamic programming, that jointly optimizes control and transmission functions, were introduced for control applications in [5][6]. However, dynamic programming suffers from high computational complexity and is intractable in practical applications. Delay-universal/anytime transmission [2], on the other hand, allows for separating the transmission problem from controller design and therefore, is a promising approach.

Anytime transmission was first introduced by Anant Sahai in [2]. Sahai stated the following communication requirements for the anytime transmission.

- Causal encoding for real-time streaming.
- Decoding at any time instant for timely recovery of information bits.

- Exponential decay of error probability with the decoding delay. *Delay-exponent* is an important performance measure, which specifies how fast the error probability decays with the delay.

The tracking capability of anytime transmission can be illustrated using the aforementioned example. Assume that the noise process W_i is streamed in an anytime manner. Further assume that the error probability $P_e(i, t)$ can be bounded as $P_e(i, t) \leq \beta e^{-\alpha(t-i)}$ for $\alpha > 0$ and for all i and t . Then it follows from (1.1) that the estimation error can be written as:

$$\begin{aligned} E \left[\left| X_{t+1} - \hat{X}_{t+1}(t) \right|^2 \right] &\leq \beta \sum_{i=0}^t A^{2(t-i)} e^{-\alpha(t-i)} \\ &\leq \beta \sum_{i=0}^t (A^2 e^{-\alpha})^{t-i}. \end{aligned} \quad (1.2)$$

It is straightforward that the estimation error is bounded if and only if $\alpha > 2 \ln A$. The example illustrates that a bounded estimation error can be achieved if the error probability decays sufficiently fast with decoding delays. In (1.2), α is known as the delay-exponent. A more general analysis in [7] shows that anytime transmission is indeed necessary and sufficient for controlling/tracking of unstable processes over noisy channels.

The feasibility of point-to-point anytime transmission was first proven in [2] using sequential random codes, but the codes are not suitable for practical implementation [8]. The existence of anytime linear codes was shown in [8], where the anytime behavior of time-invariant codes under maximum likelihood decoding was investigated. A causal anytime encoding scheme using unequal error protection (UEP) rateless codes was studied in [9]. A first result on practical anytime codes was presented in [10], where anytime reliability of low-density parity-check (LDPC) convolutional codes was shown over binary erasure channel (BEC). Later, analysis of this anytime code was extended over binary input additive white Gaussian noise (BIAWGN) channel in [11]. However, these codes suffer from error floors at large decoding delay and have limited practicality due to their high decoding complexity¹. Apart from the unavailability of suitable anytime codes, limited analytical tools are available to analyse practical anytime codes.

On the other hand, the extension of anytime transmission to the multi-user networks is particularly important for the wireless sensor/actuator networks, where

¹In the existing anytime codes, the decoding complexity grows linearly with increasing delay.

several sensors must transmit their own observation to a central control unit, and can interfere with each other or relay each other's observation. Anytime transmission for relay and multi-user networks were studied in [12–16]. With a feedback link between source and relay terminals in the two hop relay network, a characterization of the delay-exponent for regenerative relaying was discussed in [12]. Using sequential random tree codes, an achievable delay-exponent of decode-and-forward (DF) relaying strategy was studied in [13] and [14]. The achievable delay-exponent of a simple compress-and-forward (CF) relaying was presented in [15]. With sequential random codes, a theoretical analysis of the delay-exponent for multiple access channels was studied in [16]. Note that although theoretical analysis of the delay-exponent were studied in [12–16], practical anytime code design for relay and multi-user networks remains an open problem.

This thesis aims at the design and analysis of practical anytime codes for the point-to-point, relay, and multi-user communications. We organise the remainder of this thesis into six chapters. A brief overview of the major contributions of each chapter is presented as follows:

- In chapter 2, we provide brief reviews of the materials and tools relating to this thesis. We first describe the channel models including point-to-point, relay, and multiple access channels. Next, we discuss the LDPC codes and spatially coupled codes along with the iterative decoding algorithms. Finally, we briefly discuss density evolution technique as analytical tool.
- As mentioned earlier, there exist limited tools to analyse practical anytime codes. In this thesis, we develop analytical tools for anytime code, which are also applicable to analyse and design of classical block codes. These tools are first presented in Chapter 3 for block codes and then used in later chapters to analyse and design anytime codes. The following contributions are presented in this chapter.

Finite-Length error performance: We develop a novel algorithm to estimate the finite-length performance of LDPC codes. We find that our estimation technique provides better approximations than existing estimation methods.

Density evolution of spatially coupled LDPC codes: The exact density evolution of spatially coupled LDPC codes over BIAWGN channel has substantially high computational requirements. We propose a

low complexity density evolution technique for spatially coupled LDPC codes over BIAWGN channel.

SC-LDPC codes for GMAC: We present the design and analysis of spatially coupled LDPC codes for Gaussian multiple access channel (MAC). For 2-user Gaussian MAC, we present a low complexity density evolution analysis of spatially coupled LDPC codes and propose a simple optimization procedure.

- In chapter 4, we propose anytime codes for point-to-point communication based on spatially coupled codes. Through density evolution analysis, we show that our proposed anytime codes exhibit the desired anytime transmission properties. Over BEC, we find an analytical bound for the corresponding delay-exponent using mathematical induction technique and provide an algorithm to estimate the finite-length performance of the proposed anytime code. We also develop low complexity decoders for both BEC and BIAWGN channels.
- In chapter 5, we present the design and the analysis of anytime coding scheme for a three-terminal relay network. We propose a bilayer anytime code for decode-and-forward relaying over the network. The anytime characteristics of the proposed code are shown numerically for both asymptotic and finite-length case. The delay-exponents of the proposed bilayer anytime codes are analytically found using mathematical induction technique.
- In chapter 6, we propose anytime code for multiple access channel. The anytime characteristics of the proposed coding scheme are investigated asymptotically over two different multiple access channels. We derive the analytical delay-exponent for the erasure multiple access channel model.
- In chapter 7, we provide concluding remarks and give directions for future research.

Chapter 2

Background

In this chapter, we provide a brief background about channel models, LDPC codes, decoding algorithms, and density evolution analysis. In Section 2.1, we describe the channel models including point-to-point, relay, and multiple access channels. A brief review of LDPC codes and decoding algorithms is provided in Section 2.2 and Section 2.3, respectively. Finally to analyse the decoding algorithms, density evolution technique is described in Section 2.4.

2.1 Channel Models

The channel models considered throughout this thesis are given in the following.

2.1.1 Point-to-Point Communication Channel

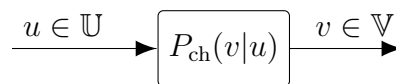


FIGURE 2.1: A point-to-point (P2P) communication model.

In a point-to-point (P2P) communication system, as shown in Fig. 2.1, one sender transmits a signal to one receiver through a communication channel. The channel can be modelled by the transition probability $P_{\text{ch}}(v|u)$, where u is the channel input from alphabet $\mathbb{U}$ and v is the channel output from alphabet $\mathbb{V}$. For the P2P communication scenario, we only consider binary-input memoryless symmetric-output (BMS) channels. Let $\mathbf{u} = [u_1, u_2, \dots, u_n] \in \{0, 1\}^n$ be the sequence of

channel inputs¹ and $\mathbf{v} = [v_1, v_2, \dots, v_n] \in \mathbb{V}^n$ be the sequence of channel outputs. For a BMS channel, following conditions must be satisfied:

- The channel input u is taken from the alphabet $\mathbb{U} = \{0, 1\}$.
- The input-output transition probabilities satisfy

$$P_{\text{ch}}(\mathbf{v}|\mathbf{u}) = \prod_{i=1}^n P_{\text{ch}}(v_i|u_i), \quad (2.1)$$

- Satisfy the symmetric condition $P_{\text{ch}}(v|u) = P_{\text{ch}}(-v|-u)$ for any $y \in \mathbb{V}$.

Among the different BMS channels, BEC and BIAWGN channel are considered in this thesis.

Binary Erasure Channel

A binary erasure channel (BEC) with erasure probability ϵ implies $\mathbb{U} = \{0, 1\}$, $\mathbb{V} = \{0, 1, ?\}$, $P_{\text{ch}}(v_i = ?|u_i = 0) = P_{\text{ch}}(v_i = ?|u_i = 1) = \epsilon$ and $P_{\text{ch}}(v_i = 0|u_i = 0) = P_{\text{ch}}(v_i = 1|u_i = 1) = 1 - \epsilon$, where $?$ indicates an erasure. A binary erasure channel is shown in Fig. 2.2. The capacity (in bits/channel use) of the BEC with erasure probability ϵ is given by

$$C_{\text{BEC}}(\epsilon) = 1 - \epsilon. \quad (2.2)$$

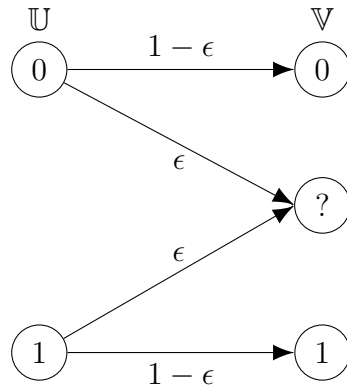


FIGURE 2.2: A binary erasure channel (BEC).

¹For binary phase shift keying (BPSK) signal, $\mathbf{u} = [u_1, u_2, \dots, u_n] \in \{+1, -1\}^n$.

Binary Input Additive White Gaussian Noise Channel

The channel output of a binary input additive white Gaussian noise (BIAWGN) channel can be described by $v_i = u_i + z_i$, where $u_i \in \{-1, +1\}$ and z_i is the Gaussian random noise with zero mean and variance σ_n^2 . The transition probability of a BIAWGN channel is given by

$$P_{\text{ch}}(v_i|u_i) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{(v_i - u_i)^2}{2\sigma_n^2}}.$$

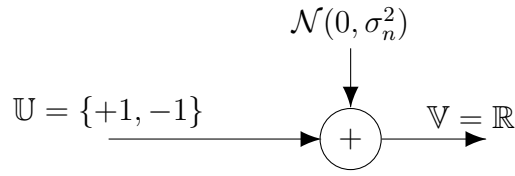


FIGURE 2.3: A binary input additive white Gaussian noise (BIAWGN) channel.

A BIAWGN channel is shown in Fig. 2.3. The capacity (in bits/channel use) of the BIAWGN channel with variance σ_n^2 is given by [17, Chapter 1]

$$C_{\text{BIWGN}}(\sigma_n) = - \int_{-\infty}^{\infty} \psi_{\sigma_n}(g) \log_2 \psi_{\sigma_n}(g) dg - \frac{1}{2} \log_2 2\pi e \sigma_n^2, \quad (2.3)$$

where

$$\psi_{\sigma_n}(g) = \frac{1}{\sqrt{8\pi\sigma_n^2}} \left(e^{-\frac{(g+1)^2}{2\sigma_n^2}} + e^{-\frac{(g-1)^2}{2\sigma_n^2}} \right).$$

2.1.2 Relay Channel

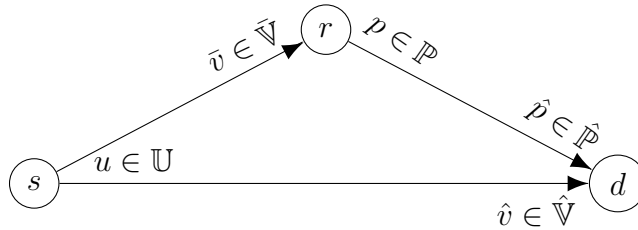


FIGURE 2.4: A three-terminal relay channel.

A relay channel consists of one source terminal s and one destination terminal d with a number of intermediate terminals known as relays to help the communication between the source and the destination terminals [18, Chapter 14]. In this

thesis, we consider three-terminal relay channel which has only one relay terminal r with communication links/channels: source-relay sr , relay-destination rd and source-destination sd . A three-terminal relay channel, as shown in Fig. 2.4, can be described by the transition probability $P_{\text{ch}}(\hat{v}, \hat{p}, \bar{v}|u, p)$, where $u \in \mathbb{U}$ and $p \in \mathbb{P}$ are the channel inputs at the source and the relay terminals, respectively; $\bar{v} \in \bar{\mathbb{V}}$, $\hat{v} \in \hat{\mathbb{V}}$ and $\hat{p} \in \hat{\mathbb{P}}$ are correspondingly the outputs of sr , sd and rd links. For simplicity, we assume that all three links are orthogonal and each of them is a binary erasure channel.

Different cooperative relay protocols were studied in [19, 20]. The decode-and-forward (DF) strategy, which is one of the prominent protocols, is considered in this thesis. The DF strategy requires the relay to decode the source message and to forward side information to the destination. The destination then attempts to recover the source message based on the information obtained from both source and relay. The achievable rate of DF relaying is known as [20]

$$R_{DF} \leq \max_{\Pr(u,p)} \min \left(I(\mathbb{U}; \bar{\mathbb{V}}|\mathbb{P}), I(\mathbb{V}, \mathbb{P}; \hat{\mathbb{P}}, \hat{\mathbb{V}}) \right)$$

where $I(\cdot)$ is the mutual information and $\Pr(u, p)$ is the joint probability distribution of the choice of symbols selected by the source and the relay.

2.1.3 Multiple Access Channel

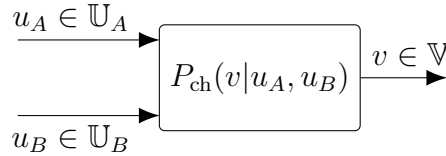


FIGURE 2.5: A 2-user multiple access channel.

Multiple access channels have many applications in the area of communication including distributed wireless communication system. Multiple access channel (MAC) is a channel in which two (or more) senders/users simultaneously transmit their information to a common receiver. In this thesis, we consider 2-user discrete memoryless MAC as shown in Fig 2.5. This MAC can be described by three alphabets, $\mathbb{U}_A$, $\mathbb{U}_B$ and $\mathbb{V}$ with transition probability $P_{\text{ch}}(v|u_A, u_B)$, where $u_A \in \mathbb{U}_A$ denotes the channel input of user-A, $u_B \in \mathbb{U}_B$ denotes the channel input of user-B, and $v \in \mathbb{V}$ denotes the channel output. The capacity region of the 2-user MAC is defined in [18, Chapter 14] by the set of rate pair (R_A, R_B) which satisfy the following constraints:

$$\begin{aligned}
R_A &\leq I(\mathbb{U}_A; \mathbb{V} | \mathbb{U}_B) \\
R_B &\leq I(\mathbb{U}_B; \mathbb{V} | \mathbb{U}_A) \\
R_A + R_B &\leq I(\mathbb{U}_A, \mathbb{U}_B; \mathbb{V})
\end{aligned}$$

where R_k is the rate of user- k , $k \in \{A, B\}$. In this thesis, the following 2-user multiple access channels are considered with perfect synchronization between user- A and user- B .

Adder Followed by an Erasure Channel

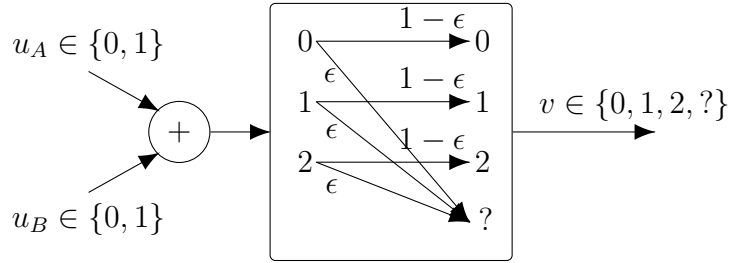


FIGURE 2.6: Multiple access channel modelled as adder channel followed by an erasure channel.

Fig. 2.6 shows a 2-user MAC which is modelled as an adder channel followed by an erasure channel. Let $u_A \in \{0, 1\}$ and $u_B \in \{0, 1\}$ denote the channel inputs of user- A and user- B , respectively. The output $v \in \{0, 1, 2, ?\}$ is given by,

$$v = \begin{cases} u_A + u_B & \text{with probability } 1 - \epsilon \\ ? & \text{with probability } \epsilon \end{cases}$$

The above channel model is the simplest 2-user MAC and hence it allows tractable analysis. Later we refer to the above described adder channel followed by an erasure channel as erasure multiple access channel.

Gaussian Multiple Access Channel

The 2-user Gaussian multiple access channel (GMAC) model considered in this thesis is shown in Fig. 2.7, where user- A and user- B transmit their BPSK signals

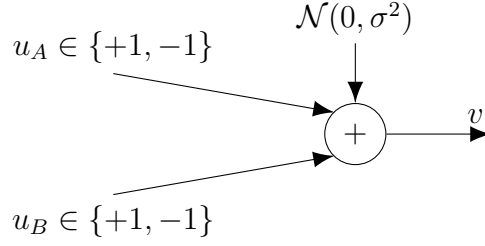


FIGURE 2.7: 2-user Gaussian multiple access channel model

$u_A \in \{+1, -1\}$ and $u_B \in \{+1, -1\}$, respectively to the receiver. Assuming unit transmission power for both user, the received signal is given by:

$$v = u_A + u_B + z,$$

where z is the noise realization at the receiver, which is assumed to be Gaussian with zero mean and variance σ_n^2 .

2.2 Low-Density Parity-Check Codes

Channel coding is the most effective technique to achieve reliable transmission of information bits over noisy communication channel. The key idea of channel coding is to transmit enough redundant bits to allow receiver to recover the information bits. In [1], Shannon stated the limit of reliable transmission via channel code. It was shown that there exist a channel code that allows the error probability at the receiver to be made arbitrarily small if the code rate is less than the capacity of the channel.

In this thesis, we focus on the linear code, an important subclass of channel codes, for which any linear combination of codewords is also a codeword.

Definition 2.1. (Linear Codes [21, Chapter 3])

A code $\mathcal{C}$ over a field $\mathbb{F}_q$ of q symbols of length N and q^K codewords is a q -ary linear code if and only if its q^K codewords form a K -dimensional vector subspace of the vector space of all the N -tuples $\mathbb{F}_q^N$. The number N is said to be the length of the code and the number K is the dimension of the code. The rate of the code is $R = \frac{K}{N}$.

Over binary field (i.e. for $\mathbb{F}_2$), a linear code $\mathcal{C}$ is described by

$$\mathcal{C} = \{\mathbf{c} \in \{0, 1\}^N : \mathbf{H} \cdot \mathbf{c}^T = \mathbf{0} \pmod{2}\},$$

where $\mathbf{c}$ is a codeword of length N and $\mathbf{H}$ is the binary parity check matrix. Among the existing linear codes, much attention has been paid to low-density parity-check (LDPC) codes, since LDPC codes allow reliable transmission at rates close to the Shannon limit, while offering a decoding complexity that scales linearly with the block length.

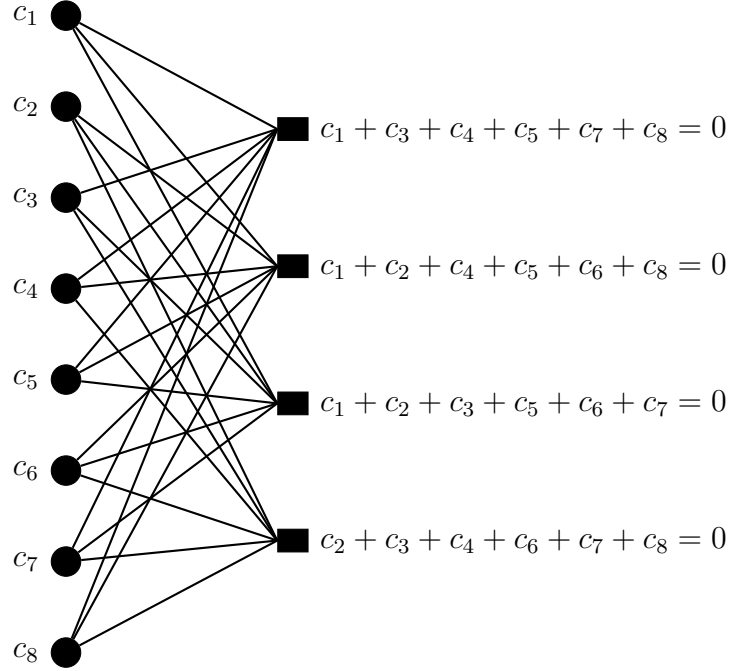


FIGURE 2.8: Tanner graph of an LDPC code.

LDPC codes were first introduced by Robert Gallager in his PhD dissertation [22] and later revisited by MacKay and Neal in [23]. In general, an LDPC code is represented by a Tanner graph [24] consisting of N variable nodes and J check nodes, where N is the block length and J is the number of check equations. Each variable node represents a coded bit and each check node represents a check equation which satisfies that the modulo-2 sum of the all connecting variable nodes is zero. An example of LDPC code with 8 variable nodes (on the left) and 4 check nodes (on the right) is illustrated in Fig. 2.8. An LDPC code can also be represented by a parity check matrix $\mathbf{H}$ of size $J \times N$. A "1" at entry (i, j) of $\mathbf{H}$ represents a connection or edge between i^{th} check node and j^{th} variable node in the Tanner graph. The parity check matrix $\mathbf{H}$ for the above graph is:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$$

An LDPC code is called (d_v, d_c) -regular, if every variable node has exactly d_v edges and every check node is connected via exactly d_c edges. In other words, the corresponding parity check matrix has d_v ones in every column and d_c ones in every row. d_v and d_c are known as the degrees of the variable and check nodes, respectively. On the other hand, irregular LDPC codes are those that have non-uniform row degrees and non-uniform column degrees in the parity check matrix. In general, the degree distributions of LDPC codes are characterized by the following functions [17, Chapter 3]:

- $\Lambda(s) = \sum_{i=2}^{d_{v\max}} \Lambda_i s^i$, where Λ_i is the fraction of variable nodes of degree i ,
- $\Gamma(s) = \sum_{i=2}^{d_{c\max}} \Gamma_i s^i$, where Γ_i is the fraction of check nodes of degree i ,
- $\nu(s) = \sum_{i=2}^{d_{v\max}} \nu_i s^{i-1}$, where ν_i is the fraction of edges that are connected to degree i variable nodes,
- $\rho(s) = \sum_{i=2}^{d_{c\max}} \rho_i s^{i-1}$, where ρ_i is the fraction of edges that are connected to degree i check nodes,

where $d_{v\max}$ and $d_{c\max}$ correspond to the maximum degree of variable nodes and check nodes, respectively. For example, the degree distribution functions for the code of Fig. 2.8 can be written as $\Lambda(s) = s^3$, $\Gamma(s) = s^6$, $\nu(s) = s^2$, and $\rho(s) = s^5$. With the above mentioned degree distributions, the code rate becomes [17, Chapter 3]

$$R = 1 - \frac{\sum_i \Lambda_i i}{\sum_i \Gamma_i i} = 1 - \frac{\sum_i \rho_i / i}{\sum_i \nu_i / i}.$$

The promising features of LDPC codes inspired the development of different class of LDPC ensembles [25–33]. Among the classes of LDPC codes, LDPC convolutional codes are candidate codes for the design of anytime code due to their suitability in streaming applications.

2.2.1 LDPC Convolutional Codes

LDPC convolutional (LDPCC) codes were first introduced by Felstrom and Zangirov [27]. The lower diagonal structure of parity check matrix of LDPC convolutional codes permits causal encoding of information bits and hence streaming applications. Moreover, LDPC convolutional codes have the fast encoding property.²

A rate $R = \frac{a}{b}$ binary LDPC convolutional code is the set of all sequences $\mathbf{c}_{[\infty]}$ satisfying the equation $\mathbf{H}_{[\infty]} \cdot \mathbf{c}_{[\infty]}^T = 0$, where $\mathbf{c}_{[\infty]}$ is the semi-infinite binary row vector $[c_1, c_2, \dots]$ and $\mathbf{H}_{[\infty]}$ is the parity-check matrix given by [35]

$$\mathbf{H}_{[\infty]} = \begin{pmatrix} \mathbf{H}_0(1) & & & & \\ \mathbf{H}_1(1) & \mathbf{H}_0(2) & & & \\ \vdots & \mathbf{H}_1(2) & \ddots & & \\ \mathbf{H}_{m_s}(1) & \vdots & \ddots & \mathbf{H}_0(L) & \\ & \mathbf{H}_{m_s}(2) & \ddots & \mathbf{H}_1(L) & \ddots \\ & & \ddots & \vdots & \ddots \\ & & & \mathbf{H}_{m_s}(L) & \ddots \\ & & & & \ddots \end{pmatrix}$$

where m_s is the syndrome former memory [34]. Each element $\mathbf{H}_i(L), i = 0, \dots, m_s$ is binary sub-matrix of size $(b - a) \times b$ that satisfies the following properties [34]:

- $\mathbf{H}_i(L) = 0$, for $i < 0$ and $i > m_s$, for all L

²In [34], it was shown that the encoding of LDPC convolutional codes can be performed by employing simple shift registers.

- There exists L such that $\mathbf{H}_{m_s}(L) \neq 0$
- $\mathbf{H}_0(L) \neq 0$ and has full rank, for all L

Code Construction: Different approaches are proposed in [27–30] to construct LDPC convolutional codes. Among them, protograph based LDPCC codes have demonstrated very good iterative decoding threshold [30]. The concept of constructing LDPC codes by connecting small graphs, was first introduced by Thorpe in [31], where the small graphs are known as *protographs*. A large graph can be obtained by copy and permutation procedure [35]. Assume that we have a protograph of size $J_p \times N_p$, i.e the protograph consists of J_p check nodes and N_p variable nodes. The protographs can be represented as submatrices $\mathbf{B}_i, i = 0, \dots, m_s$ of equal size $J_p \times N_p$. The value of each entry of $\mathbf{B}_i$ represents the number of edges that exist between a check node and a variable node. To obtain a large matrix of size $J \times N$, first we have to copy the protograph M times, where M is called lifting factor [36] such that $J = MJ_p$ and $N = MN_p$. Then connecting the edges by permutation between the M replicated protographs will result in the required LDPCC codes. The base matrix of a terminated LDPC convolutional codes is given by [35]

$$\mathbf{B}_{[1, m_s]} = \overbrace{\begin{pmatrix} \mathbf{B}_0 & & & & \\ & \mathbf{B}_1 & \mathbf{B}_0 & & \\ & \vdots & \mathbf{B}_1 & \ddots & \\ & \mathbf{B}_{m_s} & \vdots & \ddots & \mathbf{B}_0 \\ & & \mathbf{B}_{m_s} & \ddots & \mathbf{B}_1 \\ & & & \ddots & \vdots \\ & & & & \mathbf{B}_{m_s} \end{pmatrix}}^L$$

2.2.2 Spatially Coupled Codes

A class of protograph based LDPCC codes, so called spatially coupled low-density parity-check (SC-LDPC) codes, was first analytically investigated in [32]. SC-LDPC codes are constructed by coupling a chain of L standard (d_v, d_c) -regular

LDPC codes or protographs. We refer to L as the chain length and assume that the protographs are at positions³ $[1, L]$. Each of the d_v connections of a variable node at position i is uniformly connected to their neighbouring check nodes in the range i to $i + \gamma - 1$, where γ is the coupling length. A terminated SC-LDPC code can be achieved by adding extra check nodes at positions $L + 1$ to $L + \gamma - 1$. A terminated spatially coupled LDPC code is shown in Fig. 2.9. We assume that each protograph/position contains M variable nodes and $\frac{d_v}{d_c}M$ check nodes. We refer to the above terminated code as (d_v, d_c, γ, L) -SC-LDPC code with design rate

$$R_d = 1 - \frac{\frac{d_v}{d_c} \left(L + \gamma + 1 - 2 \sum_{i=0}^{\gamma} \left(\frac{i}{\gamma} \right)^{d_c} \right)}{L}.$$

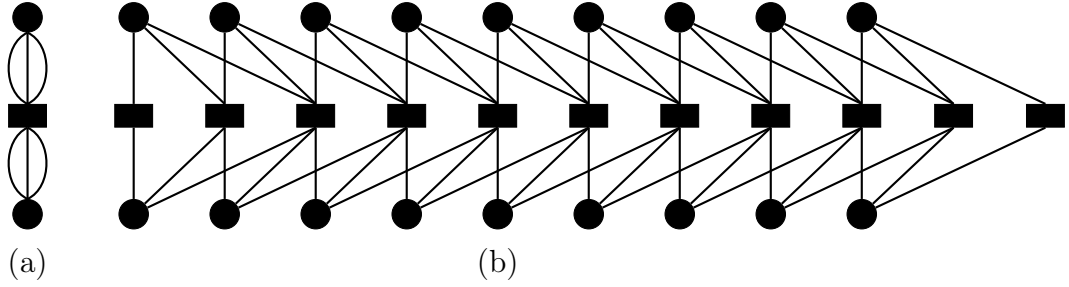


FIGURE 2.9: (a) (3, 6)-regular LDPC protograph (b) SC-LDPC code with $L = 9$ and $\gamma = 3$.

With traditional decoding algorithm, spatially coupled code with large chain length results a very complex decoder. A sliding window decoder was introduced in [35] to reduce the decoding complexity while maintaining the performance of the BP decoder. At each decoding step, the sliding window decoder only considers certain number of rows and columns within a window. The aim of each decoding step is to recover all of the first *targeted* symbols in the window. After successful recovery of targeted symbol or after a required number of iterations, the window slides down.

Spatially coupled codes exhibit capacity approaching⁴ performance over BMS channels. It has been shown that, for $L \rightarrow \infty$, spatially coupled codes exhibit a BP decoding threshold that approaches the maximum-a-posteriori (MAP) decoding threshold of the underlying protograph [32, 37–40]. For this reason, we propose a practical anytime channel code based on spatially coupled codes in Chapter 4.

³In [32], authors assumed that the protographs are at positions $[-L, L]$, $L \in \mathbb{N}$.

⁴We refer to a code as capacity approaching, when the rate of the code approaches the capacity of the corresponding channel while maintaining a reliable transmission over that channel.

2.2.3 Existing Anytime Channel Code

The existing practical anytime code, proposed in [10], was constructed based on LDPC convolutional codes. The general base matrix for this anytime code at time t was given by:

$$\mathbf{B}_{[1,t]} = \begin{bmatrix} \mathbf{B}_0 & & & & \\ \mathbf{B}_1 & \mathbf{B}_0 & & & \\ \vdots & \mathbf{B}_1 & \ddots & & \\ \vdots & \vdots & \ddots & & \\ \mathbf{B}_{t-1} & \mathbf{B}_{t-2} & \cdots & \mathbf{B}_1 & \mathbf{B}_0 \end{bmatrix} \quad (2.4)$$

Notice that, the memory m_s of the above LDPCC code increases as the number of time instances grows. This property ensures the improvement of the reliability with delay. In [10], the parameters of the above matrix was chosen such that $J_p = 1$, $N_p = 2$, $\mathbf{B}_0 = [1 \ 1]$ and $\mathbf{B}_i = [x \ 0]$ for $i > 0$, where $x \in \mathbb{N}$. Thus, the resultant base matrix becomes

$$\mathbf{B}_{[1,t]} = \begin{bmatrix} \overline{1 \ 1} & & & & & & & & \\ x \ 0 \ 1 \ 1 & & & & & & & & 0 \\ x \ 0 \ x \ 0 \ 1 \ 1 & & & & & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & \\ x \ 0 \ x \ 0 \ x \ 0 \ \cdots & 1 \ 1 & & & & & & \end{bmatrix}. \quad (2.5)$$

Although our proposed anytime code and the anytime code proposed in [10] are both LDPC convolutional codes, they differ in the following manner. For the code in [10], the degrees of variable and check nodes increase linearly with the decoding time instant. On the other hand, we propose a code structure with fixed variable and check nodes degrees. More details on our proposed anytime code will be given in Chapter 4.

2.2.4 Bilayer Expurgated LDPC codes

The aforementioned LDPC codes were designed for P2P communication. These codes are also potential candidates for relay communication. With DF relaying protocol, relay code design based on LDPC codes has been widely studied in recent literature including [41–45]. The relay codes shown in [41–45] are also known as bilayer LDPC codes. These works are based on the parity bit forwarding approach, where the relay helps the destination by transmitting additional parity bits provided that the relay is able to recover the source’s message correctly. In this thesis, we consider bilayer expurgated low-density parity-check (BE-LDPC) codes shown in [41]. In the following, we briefly describe the structure of BE-LDPC codes.

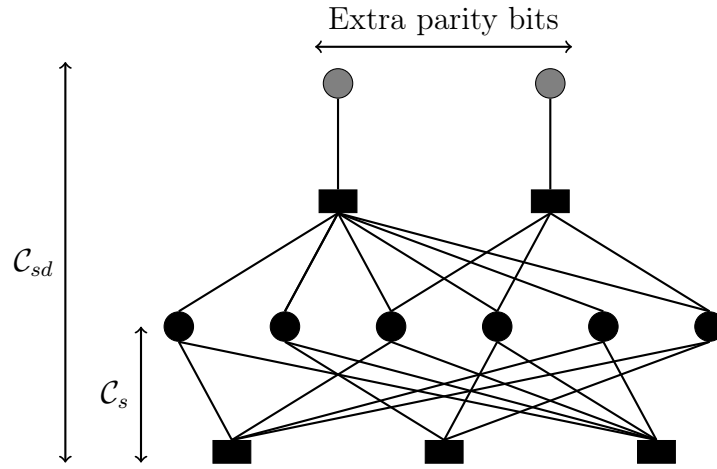


FIGURE 2.10: Tanner graph of BE-LDPC code [41]. The circles and squares represent the variable nodes and check nodes, respectively. The black circles correspond to the source transmitted bits and the gray circles correspond to the relay transmitted bits.

A BE-LDPC code for the three-terminal relay network is illustrated in Fig. 2.10. According to [41], the source encodes the message bits using a code $\mathcal{C}_s$ (lower layer graph of code $\mathcal{C}_{sd}$ in Fig. 2.10) and transmits the codeword to the relay and the destination. Code $\mathcal{C}_s$ is designed such that it satisfies capacity approaching property for sr link. Thus, we assume that relay can decode the codeword correctly. However, the destination may not be able to recover the codeword if the sd link is worse than the sr link (i.e., the rate of code $\mathcal{C}_s$ is greater than the capacity of sd link). To help the destination, the relay adds new check nodes and connects them to the structure of code $\mathcal{C}_s$, which results in the upper layer graph of Fig. 2.10. Using the upper layer, the relay generates extra parity or syndrome bits which

satisfy the corresponding new check equations. These extra parity bits are encoded by the relay using another capacity approaching code for the rd link and then transmitted to the destination. After decoding the relay transmitted bits, the destination obtains an overall code $\mathcal{C}_{sd}$ by combining extra parity bits with the code $\mathcal{C}_s$. The upper graph of Fig. 2.10 is designed such that the overall code $\mathcal{C}_{sd}$ becomes capacity approaching for the sd link. Following the above described bilayer codes, we propose anytime code for three-terminal network in Chapter 5.

2.3 Decoding Algorithms

The main strength of the LDPC codes is message passing decoding which allows decoding in time linear to the block length of the code. The following message passing decoding algorithms are considered in this thesis.

2.3.1 Erasure Decoding

For the BEC, the message passing algorithm is straight forward, since a transmitted bit is either received correctly or completely unknown. In the conventional decoder for BEC, message passing between variable node and check node occurs in the following manner. At each iteration, a variable node V whose value is known (either from the channel output or from previous iterations) sends its value to the connected check nodes, and a variable node whose value is not yet known/recovered sends an erasure symbol. On the other hand, a check node C sends to a connected variable node V an erasure if it receives at least one erasure from the connected variable nodes other than V ; otherwise C sends to V the sum (mod 2) of all incoming messages except the message from V .

2.3.2 Belief Propagation Decoding

For BIAWGN channels, belief propagation (BP) decoding algorithm is the most prominent category of message passing algorithms. In the belief propagation algorithm, the messages are expressed as log likelihood ratios (LLRs) to simplify the mathematical operations. Let u_i and v_i be the channel input and output corresponding to the variable node i . The LLR obtained from the channel for variable

node i is defined by

$$r_i = \log \frac{P_{\text{ch}}(v_i | u_i = 0)}{P_{\text{ch}}(v_i | u_i = 1)}.$$

At the beginning of the BP decoding process, each variable node sends the corresponding LLR obtained from the channel to the connected check nodes. Let $\mathbb{S}_v(i)$ represent the set of variable nodes that are connected with the i^{th} check node. Similarly, let $\mathbb{S}_c(i)$ represent the set of check nodes that are connected with the i^{th} variable node. Let $F_{j,i}^{(\ell)}$ be the message sent from variable node i to check node j at iteration ℓ . The message sent from check node j to variable node i at iteration ℓ is [17, Chapter 2]:

$$G_{j,i}^{(\ell)} = 2 \tanh^{-1} \left(\prod_{i' \in \mathbb{S}_v(j), i' \neq i} \tanh \left(\frac{F_{j,i'}^{(\ell)}}{2} \right) \right). \quad (2.6)$$

Based on the messages from check nodes, the message sent from a variable node i to check node j is

$$F_{j,i}^{(\ell+1)} = \sum_{j' \in \mathbb{S}_c(i), j' \neq j} G_{j',i}^{(\ell)} + r_i. \quad (2.7)$$

For a fixed number of iterations, the complexity of the above message passing decoders scales linearly with the number of edges. Since the number of edges grows linearly with the increasing of block length, the complexity of the above decoders scales linearly with the block length. As stated in Chapter 1, anytime decoding involves the estimation of current as well as past bits. Thus, with progression of time, the decoder has to deal with an increasingly large block and hence the window length of existing decoding techniques will grow unbounded. The existing window decoding techniques [35, 46] does not satisfy the anytime requirements because after each decoding instant the decoding window slides to the next window, leaving behind uncorrected errors/erasures. Thus, it is necessary to develop suitable decoding algorithms for anytime transmission where the complexity will not grow unbounded as time progresses. By modifying above described decoding algorithms, we present adaptive decoding algorithms for anytime transmission in Chapter 4.

2.4 Density Evolution

Density evolution is an analytical tool to analyse the performance of iterative decoding of a particular code ensemble. This technique tracks the evolution of probability density functions (pdfs) of the messages through the decoding process and provides the decoding behaviour for a given code ensemble. In general, density evolution is used to find *decoding threshold*, which is the worst channel parameter such that the decoder converges. Following properties are required for the derivation of the density evolution [17, Chapter 7].

Symmetry condition: If the channel is symmetric, the LLRs output by the iterative decoder are also symmetric. With this symmetry condition, the performance of iterative decoding becomes independent of the codeword transmitted. Therefore the all-zero codeword can be assumed for the analysis of iterative decoding performance.

Concentration: The iterative decoding threshold of a randomly chosen code from an ensemble is likely to be close to the ensemble average threshold.

Cycle-free graphs: As the block length of the code goes to infinity, the average iterative decoding performance of an ensemble converges to the performance of decoding on a cycle-free graph.

Density evolution analysis of regular and irregular LDPC codes can be found in [17, Chapter 7], which we have utilized in Chapter 3 to analyse the finite-length performance of LDPC codes. On the other hand, density evolution of SC-LDPC codes is shown in [32, 47]. However, existing density evolution of SC-LDPC code over BIAWGN channel suffers from high computational complexity and is not suitable for anytime coding scheme. Hence, we propose a low complexity density evolution technique in Chapter 3, which can also be applied in the anytime code. For the sake of simplicity and to familiarize with terminologies, we now illustrate the density evolution of (d_v, d_c) -regular LDPC code.

2.4.1 Over BEC

Let $x^{(\ell)}$ and $y^{(\ell)}$ denote the erasure probabilities of a message from variable and check nodes, respectively at iteration ℓ . The density evolution of (d_v, d_c) -regular LDPC code over BEC is explained in the following.

- Initialization: $x^{(0)} = \epsilon$.
- Check node update:

$$y^{(\ell)} = 1 - (1 - x^{(\ell)})^{d_c - 1}. \quad (2.8)$$

- Variable node update:

$$x^{(\ell)} = \epsilon (y^{(\ell-1)})^{d_v - 1}. \quad (2.9)$$

The recursions in (2.9) and (2.8) are termed density evolution equations since they describe how the erasure probability of the iterative decoder evolves as a function of the iteration number ℓ . The decoding threshold ϵ^* can be obtained by

$$\epsilon^* = \sup \left\{ \epsilon : \lim_{\ell \rightarrow \infty} x^{(\ell)} = 0 \right\}.$$

2.4.2 Over BIAWGN Channel

Density evolution over the BEC is straightforward, since it tracks the erasure probability, which is a scalar quantity. However, this is not the case for the BIAWGN channel. The tracking parameter for the BIAWGN channel is the pdf of the LLR message [48]. Thus, density evolution over BIAWGN channel becomes computationally intensive task. To reduce the complexity, approximation techniques were introduced in [49], which allow tracking of one dimensional parameters/metrics. These one dimensional parameters are deduced from the LLR message distribution. It was shown that the message distributions can be well approximated through these scalar parameters and hence, one can find the threshold without much sacrifice in accuracy.

Reciprocal-Channel Approximation

In reciprocal-channel approximation (RCA) density evolution, proposed in [49], the one dimensional tracking metrics are SNR (for variable node to check node metric) and the reciprocal of the SNR (for check node to variable node metric). For a given SNR s , there exist a reciprocal of s denoted by r such that $C_B(s) + C_B(r) = 1$, $C_B(g)$ denotes the capacity⁵ of the BIAWGN channel with SNR g . s and r are

⁵With SNR g , $C_B(g)$ is equivalent to $C_{\text{BIAWGN}}(\sqrt{\frac{1}{g}})$ of (2.3).

related with each other such that $r = C_B^{-1}(1 - C_B(s))$ and $s = C_B^{-1}(1 - C_B(r))$. With RCA in density evolution, s is additive at variable nodes and r is additive at check nodes [50]. Let $x^{(\ell)}$ and $y^{(\ell)}$ be the metric outgoing from variable node and check node, respectively at iteration ℓ . Using the RCA technique, we get the following density evolution equations for the (d_v, d_c) -regular LDPC codes over BIAWGN channel.

- Initialization: $x^{(0)} = \frac{1}{\sigma_n^2}$.
- Check node update:

$$y^{(\ell)} = (d_c - 1)C_B^{-1}(1 - C_B(x^{(\ell)})).$$

- Variable node update:

$$x^{(\ell)} = \frac{1}{\sigma_n^2} + (d_v - 1)C_B^{-1}(1 - C_B(y^{(\ell-1)})).$$

The decoding threshold σ_n^* can be found by

$$\sigma_n^* = \sup \left\{ \sigma_n : \lim_{\ell \rightarrow \infty} x^{(\ell)} = \infty \right\}.$$

Gaussian Approximation

In Gaussian approximation (GA) density evolution, proposed in [49], the one dimensional tracking parameter is the mean μ_L of the channel LLR message and the pdf of the LLR messages are approximated as a symmetric Gaussian $\mathcal{N}(\mu_L, \sigma_L^2 = 2\mu_L)$ distribution. Let $x^{(\ell)}$ and $y^{(\ell)}$ be the mean of LLR message outgoing from variable node and check node, respectively at iteration ℓ . Using the GA technique, we get the following density evolution equations for (d_v, d_c) -regular LDPC codes.

- Initialization: $x^{(0)} = \frac{2}{\sigma_n^2}$.
- Check node update:

$$y^{(\ell)} = \phi^{-1} \left(1 - [1 - \phi(x^{(\ell)})]^{d_c-1} \right).$$

- Variable node update:

$$x^{(\ell)} = \frac{2}{\sigma_n^2} + (d_v - 1)y^{(\ell-1)}.$$

following [48] we define

$$\phi(g) = \begin{cases} 1 - \frac{1}{\sqrt{4\pi g}} \int_{-\infty}^{\infty} \left(\tanh \frac{f}{2} \right) e^{-\frac{(f-g)^2}{4g}} df, & \text{if } g > 0 \\ 1, & \text{if } g = 0 \end{cases}$$

The decoding threshold σ_n^* can be found by

$$\sigma_n^* = \sup \left\{ \sigma_n : \lim_{\ell \rightarrow \infty} x^{(\ell)} = \infty \right\}.$$

Both reciprocal-channel and Gaussian approximations compute the decoding threshold with very similar accuracy and complexity. However, there is no natural way of generalizing the reciprocal-channel approximation to irregular LDPC codes [49]. On the other hand, Gaussian approximation can be generalised for both regular and irregular LDPC codes.

Chapter 3

Tools for Anytime Code

In this chapter, we propose analytical tools to analyse and design the block codes. We show that proposed tools exhibit better performance and/or less complexity than the existing tools. Additionally, proposed tools can be used to analyse the performance of practical anytime codes.

This chapter is organised in the following manner. An algorithm to estimate the bit erasure/error probability of finite-length LDPC codes is introduced in Section 3.1. In the next section, low complexity density evolution techniques are proposed for the SC-LDPC codes over BIAWGN channel. In Section 3.3, we show the design of GMAC codes using SC-LDPC codes and present low complexity density evolution technique for 2-user GMAC. Note that the GMAC code design in Section 3.3 does not represent a analytical tool, rather act as a stepping stone to design anytime code for GMAC.

3.1 Performance Estimation of Finite-Length LDPC Codes

Asymptotic behaviours of LDPC codes are well understood and the asymptotic performance can be measured through density evolution analysis. In contrast to the asymptotic analysis, the behaviour of finite-length LDPC codes is not straightforward. The finite-length performance of LDPC codes is divided into two regions, namely the error floor region and the waterfall region. Error floors occur due to small structures generated in the code, where the error performance of the code is dominated by the minimum distance of the code and does not decrease as rapidly

as it might be expected. On the other hand, in the waterfall region, the error performance of the code drops off quickly as a function of the channel parameter.

Finite-length behaviour of LDPC codes over binary erasure channel was first investigated in [51], which describes both the waterfall and error floor regions. This approach was based on stopping set analysis. Although stopping set analysis gives an accurate estimation, it becomes impractical for large block length due to very high computational complexity. A similar approach with slightly lower complexity was proposed in [52]. In [53], a scaling law method was introduced, which can predict the performance of finite-length LDPC codes almost accurately. The prediction accuracy of the scaling law method depends on how accurately one can find the *scaling parameters* [53] and the decoding threshold. Although the scaling law method provides a low complexity analysis, finding the scaling parameters is not an easy task for all ensembles and decoding algorithms. More recently, another approach was proposed in [54], which provides finite-length performance estimation in the waterfall region. Considering the decoding failure as an event when the realized channel quality is worse than the decoding threshold, the block error probability was obtained. Using the extrinsic information transfer (EXIT) chart analysis and the obtained block error probability, this method calculates the bit error probability. This threshold method offers an efficient and low complexity estimation in the waterfall region. However, the prediction becomes inaccurate for small block lengths. Moreover, both scaling and threshold methods do not work for the cases where it is not possible to find the decoding threshold. Suitable examples of such cases are the anytime codes.

In this Section, we present a new method to estimate the finite-length performance of LDPC codes. Utilizing the density evolution technique, our approach can predict the bit erasure/error probability of any given ensemble. This approach holds for various BMS channels. Through comparison with simulation results, we show that our approach gives a close approximation for the bit erasure/error performance of LDPC codes. Although our proposed method exhibits higher computational complexity, it provides more accurate estimations than that obtained from the method proposed in [54]. More importantly, our proposed method can be used to predict the finite-length performance of anytime code.

3.1.1 Proposed Algorithm

In the asymptotic density evolution, the realized channel parameter observed by each codeword is concentrated and fixed. However, for finite-length codes, the channel observed by a codeword is a random quantity. Similar to [54], our estimation technique takes into account this randomness of the channel to evaluate the performance of finite-length codes, while assuming that the codes are cycle-free. For example, for the erasure channel, we define a random variable E , the empirical bit erasure probability observed by a codeword. Then NE has binomial distribution $B(N, \epsilon)$. For large N , the distribution of E can be approximated by a Gaussian distribution $\mathcal{N}(\epsilon, \frac{\epsilon(1-\epsilon)}{N})$ with mean ϵ and variance $\frac{\epsilon(1-\epsilon)}{N}$. Similarly, for the BIAWGN channel, the distribution of the observed error probability can be approximated by $\mathcal{N}(P_c, \frac{P_c(1-P_c)}{N})$, where P_c is the probability of receiving a bit in error due to the BIAWGN channel. Conditioning on a channel realization (erasure/error probability), we first find the conditional decoded erasure/error probability from density evolution. Then by marginalizing over the channel realizations, we find the erasure/error probability.

Let k denote the realized number of erasures or the empirical mean of LLR. Conditioned on k , we consider δ_k as the erasure probability (for BEC) or the LLR mean (for BIAWGN channel) realized from the channel. We also consider $x_k^{(\ell)}$ and $y_k^{(\ell)}$ as the erasure probabilities obtained from the variable node decoder and the check node decoder, respectively at iteration ℓ . We denote e_k as the realized channel erasure/error probability¹ and $P_{e|k}$ as the bit erasure/error probability returned by the decoder given k erasures/LLR mean realized at the channel. Using the following steps we can estimate the performance of finite-length LDPC codes.

Quantization: Quantize the realized channel erasures (BEC) or empirical mean of channel LLR (BIAWGN channel).

$P_{e|k}$ **calculation:** For each of the quantized values of k , find the corresponding bit erasure/error probability $P_{e|k}$ returned by the decoder using following steps:

1. At the variable node decoder, compute $x_k^{(\ell)}$ for given δ_k and $y_k^{(\ell-1)}$ [Equation (3.2) and (3.5) for BEC and BIAWGN channel, respectively].
2. At the check node decoder, compute $y_k^{(\ell)}$ for given $x_k^{(\ell)}$ [Equation (3.3) and (3.6) for BEC and BIAWGN channel, respectively].

¹For BEC, $\delta_k = e_k$.

3. Repeat step 1 and 2 for a maximum number of iterations $\ell_{\max}$.
4. Find the conditional bit erasure/error probability $P_{e|k}$ using δ_k and $y_k^{(\ell_{\max})}$ [Equation (3.4) and (3.7) for BEC and BIAWGN channel, respectively].

Marginalization: Marginalize $P_{e|k}$ over k to obtain the bit erasure/error probability P_e .

$$P_e = \sum_k P_r(e_k) P_{e|k}, \quad (3.1)$$

where $P_r(e_k)$ denotes the probability of the erasure/error probability e_k , which can be obtained from the distribution of the realized channel erasure/error probability. The rules of finding $x_k^{(\ell)}$, $y_k^{(\ell)}$, and $P_{e|k}$ depend on the variation of channel type and decoding algorithm.

3.1.2 Over BEC

Following [17, Chapter 7] the variable node update, check node update, and conditional erasure probabilities are as follows:

Variable node update:

$$x_k^{(\ell)} = \delta_k \nu \left(y_k^{(\ell-1)} \right). \quad (3.2)$$

Check node update:

$$y_k^{(\ell)} = 1 - \rho \left(1 - x_k^{(\ell)} \right). \quad (3.3)$$

Conditional erasure probability:

$$P_{e|k} = \delta_k \Lambda \left(y_k^{(\ell_{\max})} \right). \quad (3.4)$$

For k erasures from the channel, we calculate the observed erasure probability by $e_k = \frac{k}{N}$. We find $P_r(e_k)$ from the pdf of the realized channel erasure probability and marginalize $P_{e|k}$ over k using (3.1) to calculate the erasure probability P_e of a given ensemble. Fig. 3.1 shows a comparison between simulation result and our prediction method for (3, 6)-regular LDPC codes.

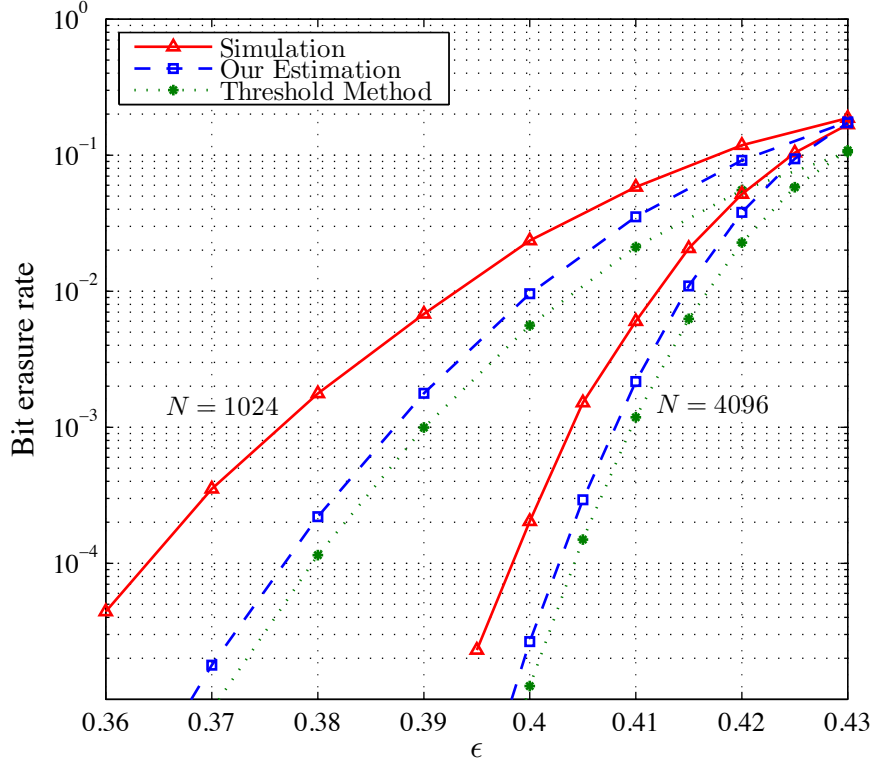


FIGURE 3.1: Bit erasure probability comparison for $(3,6)$ -regular LDPC codes between simulation results, our estimations and threshold method results proposed in [54].

3.1.3 Over BIAWGN Channel

For the BIAWGN channel, we consider LDPC codes under BP decoding algorithm. To implement density evolution based on BP decoding algorithm, we have utilized the Gaussian approximation technique shown in [49], which provides close results compared to the actual implementation. Following [48], the update equations for the variable node, the check node, and the conditional erasure probability are given by:

Variable node update:

$$x_k^{(\ell)} = \sum_{i=2}^{d_{v\max}} \nu_i \left(\delta_k + (i-1)y_k^{(\ell-1)} \right). \quad (3.5)$$

Check node update:

$$y_k^{(\ell)} = \sum_{j=2}^{d_{c\max}} \rho_j \phi^{-1} \left(1 - \left[1 - \sum_{i=2}^{d_{v\max}} \nu_i \phi \left(\delta_k + (i-1)y_k^{(\ell-1)} \right) \right]^{j-1} \right). \quad (3.6)$$

Conditional error probability:

$$P_{e|k} = \sum_{i=2}^{d_{v\max}} \Lambda_i \left(\sqrt{\frac{\delta_k + iy_k^{(\ell_{\max})}}{2}} \right). \quad (3.7)$$

Each quantized LLR value k corresponds to the error probability given by $e_k = Q(\sqrt{\frac{k}{2}})$. For finite-length N , the distribution of e_k can be approximated by a Gaussian distribution with $\mathcal{N}(P_c, \frac{P_c(1-P_c)}{N})$ and $P_r(e_k)$ can be calculated from this distribution. With noise variance σ_n^2 and assuming symmetric Gaussian distribution, the average error probability P_c can be calculated by $P_c = Q(\frac{1}{\sigma_n})$ [54]. By marginalizing $P_{e|k}$ over k using (3.1), we can obtain the error probability P_e of a given ensemble.² In Fig. 3.2, our estimations for (3,6)-regular LDPC codes of different block lengths are compared to the simulation and reference [54] results. Then we compare our estimation with the simulation result for irregular case. For a rate $\frac{1}{2}$ irregular code, Fig. 3.3 shows the comparison between our estimation, simulation and reference [54] results.

3.1.4 Discussions

From the above comparisons, we find that our estimation technique provides better results than the method in [54]. Similar to [54], we have observed a reduction in the gap between estimation and simulation results for large block length. We also observe that our proposed algorithm provides very good estimation results when the channel parameters are close to the decoding threshold of the code. Moreover, our proposed method is useful in the case, where it is difficult to find the decoding threshold accurately.

3.2 Density Evolution of SC-LDPC Codes over BIAWGN Channel

As discussed in Section 2.4 of Chapter 2, the tracking parameter for the BIAWGN channel is the pdf of the LLR message. Thus, the actual density evolution of SC-LDPC codes over BIAWGN channel requires tracking multiple pdfs of messages

²Note that due to the infinite range of LLR value, we choose to marginalize with respect to the realized bit error probability instead of marginalizing with respect to LLR mean.

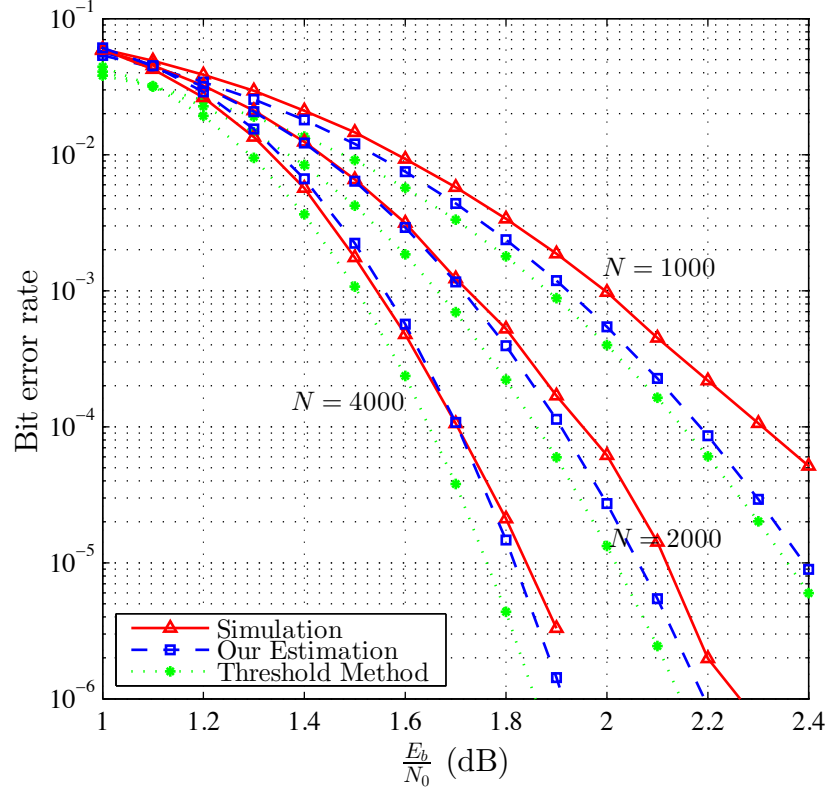


FIGURE 3.2: Bit error rate comparison between the simulation results, our estimation and threshold method [54] for (3,6)-regular LDPC codes over the BIAWGN channel.

travelling over multiple edge types. This results in substantially high computational requirements. To reduce complexity, the reciprocal channel approximation (RCA) was used within the multi-edge type (MET) framework in [50], where the tracking parameter of the density evolution analysis is a scalar quantity.

Due to the MET framework, the density evolution analysis of SCLDPC codes in [50] tracks the parameter of each edge type separately. Moreover, deterministic connection between variable node and check node was assumed in [50]. On the other hand, for SC-LDPC codes with probabilistic coupling between protographs, applying the MET framework results in prohibitively complex analysis. Examples of such cases are anytime spatially coupled codes proposed in [55], where MET framework can not be applied.

In this section, we propose a low complexity density evolution approximation for SC-LDPC codes over BIAWGN channel. Together with the RCA and GA, we introduce averaging techniques in density evolution. Instead of tracking the parameters of RCA or GA on different edge-types, we track these parameters for different node-types. Since the number of node-types is generally much smaller

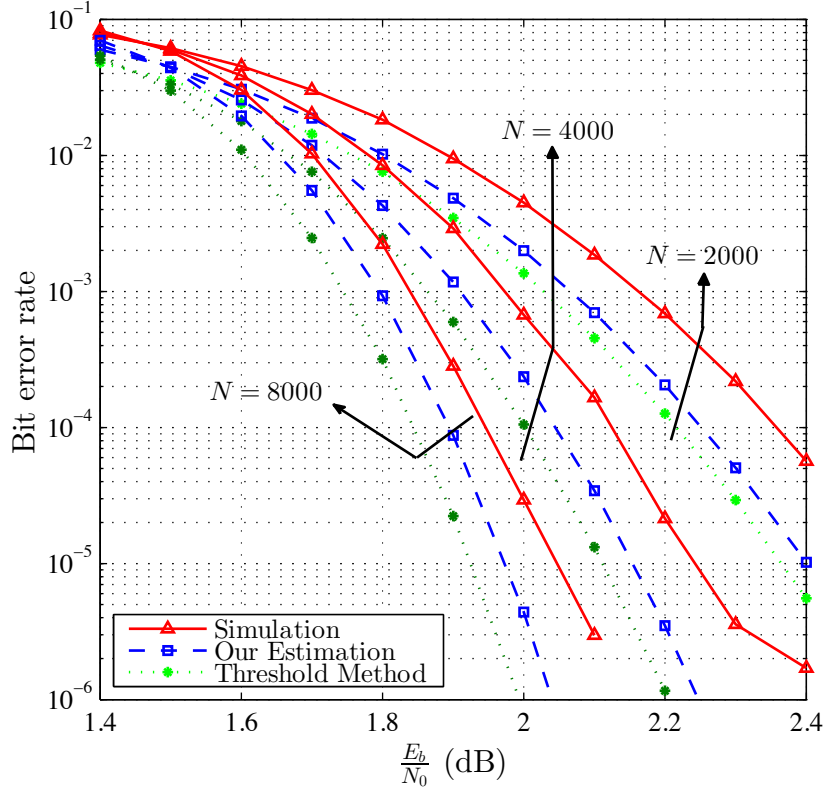


FIGURE 3.3: Bit error rate comparison between the simulation, our estimation and threshold method [54] for irregular LDPC codes. We specified the degree distribution pair of the code as $\nu(s) = 0.4s^2 + 0.4s^5 + 0.2s^8$ and $\rho(s) = s^8$.

than the number of edge-types, the proposed techniques have significantly lower complexity than density evolution with RCA or GA in the MET framework. The decoding thresholds obtained from the proposed techniques are compared with the exact decoding thresholds obtained from the actual density evolution over the MET framework. Although the proposed techniques slightly reduce the accuracy, their computational complexity is close to that of density evolution over the BEC. Moreover, the proposed density evolution techniques are also applicable for spatially coupled codes with random coupling.

The remainder of this section is organized as follows. For sake of completeness, we present the actual MET density evolution in Subsection 3.2.1 and MET with approximation techniques in Subsection 3.2.2. In Subsection 3.2.3, we introduce proposed density evolution analysis and present the numerical results in Subsection 3.2.4.

3.2.1 Multi Edge Type Density Evolution

We first describe the MET representation of SC-LDPC codes and then show the exact density evolution of SC-LDPC codes over the BIAWGN channel. A MET code can be specified by two node perspective multinomials defined as $\Lambda(\mathbf{s}) = \sum \Lambda_{\mathbf{q}} \mathbf{s}^{\mathbf{q}}$ and $\Gamma(\mathbf{s}) = \sum \Gamma_{\mathbf{q}} \mathbf{s}^{\mathbf{q}}$, where $\mathbf{s}, \mathbf{q}$ are vectors which are explained as follows:

- $\mathbf{s} = [s_1, \dots, s_{n_e}]$ denotes a vector of variables, each corresponds to an edge-type. n_e is the number of edge-types used to represent the graph ensemble;
- $\mathbf{q} = [q_1, \dots, q_{n_e}]$ denotes degrees of edge-types, where q_i is the number of edges of type i that connect to the same check/variable node.

The non-negative coefficient $\Lambda_{\mathbf{q}}$ ($\Gamma_{\mathbf{q}}$) represents the fraction of variable (check) nodes of type $\mathbf{q}$. SC-LDPC codes can be included in the MET framework. A simple example of a MET SC-LDPC code is shown in Fig. 3.4, where we assume deterministic connections between variable nodes and check nodes with $\gamma = d_v$. For this example, the multinomials are given by

$$\begin{aligned}\Lambda(\mathbf{s}) &= 0.5s_1s_2s_3 + 0.5s_4s_5s_6, \\ \Gamma(\mathbf{s}) &= 0.25s_1^2 + 0.25s_2^2s_4^2 + 0.25s_3^2s_5^2 + 0.25s_6^2.\end{aligned}$$

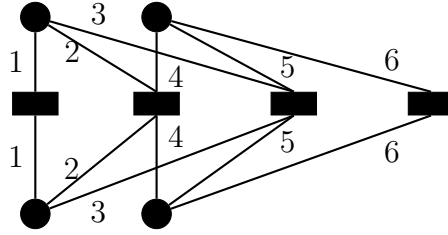


FIGURE 3.4: MET representation of $(d_v, d_c, \gamma, L) = (3, 6, 3, 2)$ -SC-LDPC code with $n_e = 6$. The edge types are denoted along side of each edges.

Additionally, for density evolution, we define two following edge perspective multinomials $\nu(\mathbf{s})$ and $\rho(\mathbf{s})$:

$$\begin{aligned}\nu(\mathbf{s}) &= \left(\frac{\Lambda_{s_1}(\mathbf{s})}{\Lambda_{s_1}(\mathbf{1})}, \frac{\Lambda_{s_2}(\mathbf{s})}{\Lambda_{s_2}(\mathbf{1})}, \dots, \frac{\Lambda_{s_{n_e}}(\mathbf{s})}{\Lambda_{s_{n_e}}(\mathbf{1})} \right), \\ \rho(\mathbf{s}) &= \left(\frac{\Gamma_{s_1}(\mathbf{s})}{\Gamma_{s_1}(\mathbf{1})}, \frac{\Gamma_{s_2}(\mathbf{s})}{\Gamma_{s_2}(\mathbf{1})}, \dots, \frac{\Gamma_{s_{n_e}}(\mathbf{s})}{\Gamma_{s_{n_e}}(\mathbf{1})} \right),\end{aligned}$$

where,

$$\Lambda_{s_i}(\mathbf{s}) = \frac{d\Lambda(\mathbf{s})}{ds_i} \text{ and } \Gamma_{s_i}(\mathbf{x}) = \frac{d\Gamma(\mathbf{s})}{ds_i}.$$

We also define a vector of n_e pdfs $\mathbf{A}^{(\ell)} = [A_1^{(\ell)}, A_2^{(\ell)}, \dots, A_{n_e}^{(\ell)}]$, where, $A_i^{(\ell)}$ is the pdf of a message on an edge of type i at iteration ℓ . According to [36, Chapter 5], we get the following density evolution recursion at iteration ℓ :

$$\mathbf{A}^{(\ell)} = A_{\text{BIAWGN}(\sigma_n)} * \boldsymbol{\nu} \left(\boldsymbol{\rho} \left(\mathbf{A}^{(\ell-1)} \right) \right), \quad (3.8)$$

where $A_{\text{BIAWGN}(\sigma_n)}$ is the pdf of the LLR of received message corresponding to a bit transmitted over the BIAWGN channel with noise variance σ_n^2 . In (3.8), the multiplication operation inside $\boldsymbol{\nu}(\mathbf{X})$ ($\boldsymbol{\rho}(\mathbf{X})$) is convolution of densities in the variable (check) node domain. Let Δ_k be the Dirac delta function of unit mass concentrated at k . By $\boldsymbol{\Delta}_0$ and $\boldsymbol{\Delta}_\infty$, we define vectors of densities where each density is Δ_0 and Δ_∞ , respectively. We assume $\mathbf{A}^{(0)} = A_{\text{BIAWGN}(\sigma_n)} * \boldsymbol{\nu}(\boldsymbol{\Delta}_0)$, i.e., initially each variable node sends its received value. According to [36], we find the threshold by

$$\sigma_n^* = \sup \left\{ \sigma_n : \lim_{\ell \rightarrow \infty} A_{\text{BIAWGN}(\sigma_n)} * \Lambda \left(\boldsymbol{\rho} \left(\mathbf{A}^{(\ell)} \right) \right) = \boldsymbol{\Delta}_\infty \right\}.$$

For more details about MET density evolution, we refer the readers to [36]. This density evolution provides an accurate prediction of decoding thresholds. However, MET density evolution suffers from high computational complexity, since it tracks the whole distribution of each edge-type message.

3.2.2 MET Density Evolution with Approximation

We now summarise density evolution of SC-LDPC codes using approximations (i.e., RCA and GA) in the MET framework as depicted in Fig. 3.4. These approximations reduce the complexity of the actual MET density evolution analysis by tracking a scalar quantity instead of the whole message distribution along each edge-type. In [49], it is shown that the message distributions can be well approximated through these scalar quantities and hence, one can find the threshold without much sacrifice in accuracy.

Let $\mathbb{E}_v(i)$ be the set of edge types that share the variable node with an edge of type i . Similarly, let $\mathbb{E}_c(i)$ be the set of edge types that share the check node with

an edge of type i . Let $x_e^{(\ell)}(i)$ and $y_e^{(\ell)}(i)$ $i \in \{1, 2, \dots, n_e\}$ be the metrics, which are deduced from the LLR message distribution through RCA or GA, passed from variable and check nodes, respectively, along the edge type i , at iteration ℓ . For RCA and GA, the nature of $x_e^{(\ell)}(i)$ and $y_e^{(\ell)}(i)$ are mentioned in Section 2.4 of Chapter 2. The RCA and GA in the MET framework are more rigorously described in the following.

The RCA Technique

- Initialization: $x_e^{(0)}(i) = \frac{1}{\sigma_n^2}$.
- $y_e^{(\ell)}(i)$ update:

$$y_e^{(\ell)}(i) = (q_i - 1)C_B^{-1}(1 - C_B(x_e^{(\ell)}(i))) + \sum_{j \in \mathbb{E}_c(i)} q_j C_B^{-1}(1 - C_B(x_e^{(\ell)}(j))).$$

- $x_e^{(\ell)}(i)$ update:

$$x_e^{(\ell)}(i) = \frac{1}{\sigma_n^2} + (q_i - 1)C_B^{-1}(1 - C_B(y_e^{(\ell-1)}(i))) + \sum_{j \in \mathbb{E}_v(i)} q_j C_B^{-1}(1 - C_B(y_e^{(\ell-1)}(j))).$$

The GA Technique

- Initialization: $x_e^{(0)}(i) = \frac{2}{\sigma_n^2}$.
- $y_e^{(\ell)}(i)$ update:

$$y_e^{(\ell)}(i) = \phi^{-1} \left[1 - (1 - \phi(x_e^{(\ell)}(i)))^{q_i-1} \prod_{j \in \mathbb{E}_c(i)} (1 - \phi(x_e^{(\ell)}(j)))^{q_j} \right].$$

- $x_e^{(\ell)}(i)$ update:

$$x_e^{(\ell)}(i) = \frac{2}{\sigma_n^2} + (q_i - 1)y_e^{(\ell-1)}(i) + \sum_{j \in \mathbb{E}_v(i)} q_j y_e^{(\ell-1)}(j).$$

For both RCA and GA techniques, we find the decoding threshold σ_n^* by

$$\sigma_n^* = \sup \left\{ \sigma_n : \lim_{\ell \rightarrow \infty} x_e^{(\ell)}(i) = \infty, \quad \forall i \right\}.$$

3.2.3 Proposed Density Evolution Analysis

In BEC, an average of the incoming erasure probabilities at the variable or check nodes can be used to calculate the outgoing erasure probabilities [32]. This property results in a low complexity density evolution analysis of SC-LDPC codes over the BEC. Motivated by the density evolution over BEC, we propose averaging techniques in the density evolution analysis over BIAWGN channels, while utilizing RCA (or GA). In our proposed density evolution, we track the messages from different nodes instead of tracking the message for different edge-types. We compute the outgoing messages from a node based on an averaging function of the incoming messages. Using RCA and GA, we propose averaging the mutual information between the incoming message and the associated code bit. For each variable or check node, we convert each of the incoming metric (SNR or mean of LLR message) to the corresponding mutual information and average these mutual information. Then, the average mutual information is converted back to the corresponding metric, which is used to calculate the outgoing metric. For the proposed density evolution analysis, we consider $x^{(\ell)}(i)$ and $y^{(\ell)}(i)$ to be the metrics³, outgoing from variable and check nodes, respectively. The indices i and ℓ denote the node position and iteration number, respectively. In the following density evolution, we set $x^{(\ell)}(i) = \infty$, $\forall \ell$ when $i \notin [1, L]$. Using RCA and GA, in the following we present the proposed update steps of $x^{(\ell)}(i)$ and $y^{(\ell)}(i)$ for $i \in [1, L]$.

The RCA Technique

- Initialization: $x^{(0)}(i) = \frac{1}{\sigma_n^2}$.
- $y^{(\ell)}(i)$ update:

$$y^{(\ell)}(i) = (d_c - 1)C_B^{-1} \left(1 - \frac{1}{\gamma} \sum_{k=0}^{\gamma-1} C_B(x^{(\ell)}(i - k)) \right).$$

- $x^{(\ell)}(i)$ update:

$$x^{(\ell)}(i) = \frac{1}{\sigma_n^2} + (d_v - 1)C_B^{-1} \left(1 - \frac{1}{\gamma} \sum_{k=0}^{\gamma-1} C_B(y^{(\ell-1)}(i + k)) \right).$$

³The nature of $x^{(\ell)}(i)$ and $y^{(\ell)}(i)$ are same as $x_e^{(\ell)}(i)$ and $y_e^{(\ell)}(i)$, respectively.

The GA Technique

- Initialization: $x^{(0)}(i) = \frac{2}{\sigma_n^2}$.
- $y^{(\ell)}(i)$ update:

$$y^{(\ell)}(i) = \phi^{-1} \left[1 - \left\{ 1 - \phi \left(\mathcal{J}^{-1} \left(\frac{1}{\gamma} \sum_{k=0}^{\gamma-1} \mathcal{J} (x^{(\ell)}(i-k)) \right) \right) \right\}^{d_c-1} \right].$$

- $x^{(\ell)}(i)$ update:

$$x^{(\ell)}(i) = \frac{2}{\sigma_n^2} + (d_v - 1) \mathcal{J}^{-1} \left(\frac{1}{\gamma} \sum_{k=0}^{\gamma-1} \mathcal{J} (y^{(\ell-1)}(i+k)) \right),$$

where

$$\mathcal{J}(g) = 1 - \frac{1}{\sqrt{4\pi g}} \int_{-\infty}^{\infty} e^{-\frac{(f-g)^2}{4g}} \log_2(1 + e^{-f}) df.$$

For both RCA and GA techniques, the decoding threshold is

$$\sigma_n^* = \sup \left\{ \sigma_n : \lim_{\ell \rightarrow \infty} x^{(\ell)}(i) = \infty, \quad \forall i \right\}.$$

Complexity Comparison: Our proposed density evolution analysis offers a reduction in complexity compared to the earlier MET techniques by tracking node-type instead of edge-type messages. For SC-LDPC codes, the number of messages to be tracked in our proposed approach is at most $\frac{1}{d_v}$ of that in the MET techniques. For example, with $(d_v, d_c, \gamma = d_v, L)$ and $(d_v, d_c, \gamma = 2d_v, L)$ -SC-LDPC codes in the MET framework, there exist d_v and $2d_v$ types of edges, respectively at each position. Thus, MET techniques track $d_v L$ and $2d_v L$ messages (variable node to check node), respectively. On the other hand, for both SC-LDPC codes, there exists single type of node at each position. Thus, our proposed density evolution tracks only L messages.

3.2.4 Results and Discussion

In this section, we present numerical results obtained from the aforementioned density evolution techniques. The thresholds for different SC-LDPC codes obtained from different density evolution techniques are plotted in Fig. 3.5 and Fig. 3.6.

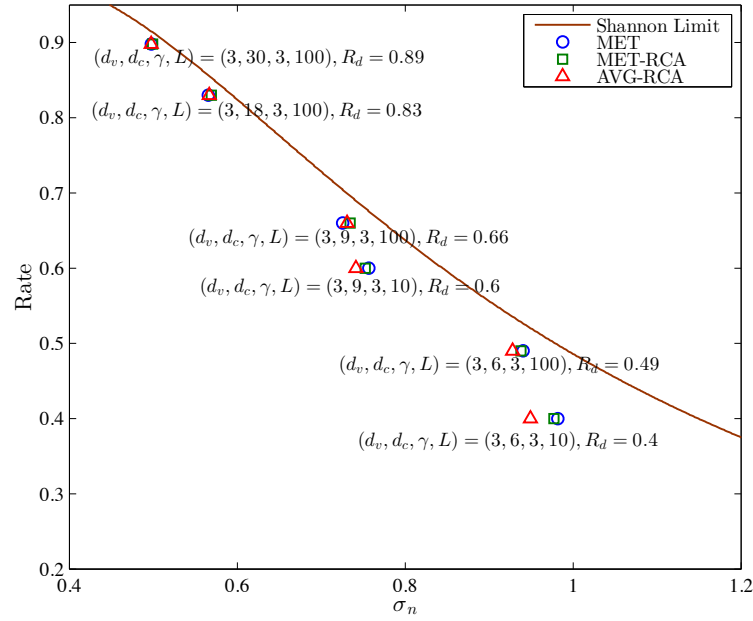


FIGURE 3.5: Decoding thresholds of different SC-LDPC codes with reciprocal channel approximation (RCA) technique

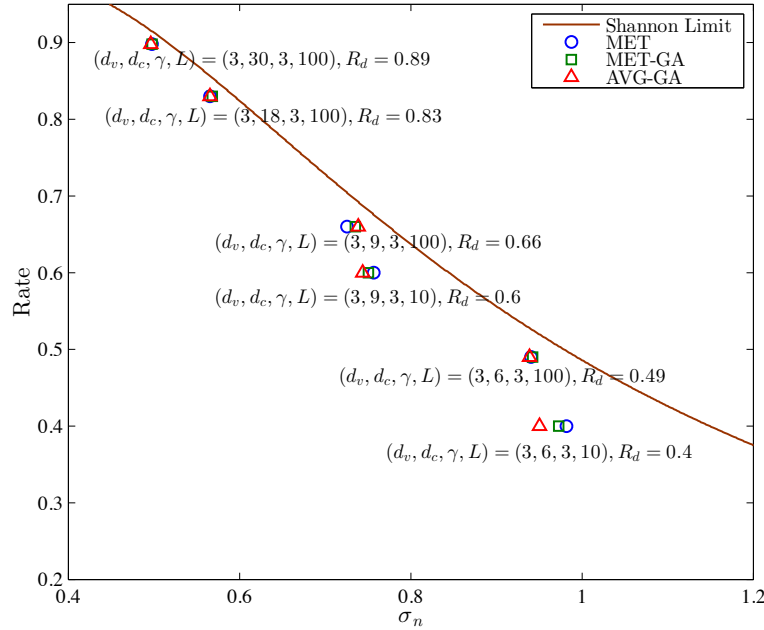


FIGURE 3.6: Decoding thresholds of different SC-LDPC codes with Gaussian approximation (GA) technique

We observe that the threshold obtained from RCA-MET and GA-MET density evolutions is a close approximation to the exact threshold obtained from the actual MET density evolution as described in Subsection 3.2.1. Although MET density evolution techniques lead to a good prediction of decoding thresholds, they suffer from high computational complexity and become very complicated when there exist probabilistic connections between variable and check nodes.

On the other hand, the decoding thresholds obtained from the proposed RCA-averaging and GA-averaging techniques provide slightly inaccurate approximations, especially at low rates. However, they offer the least computational complexity. Moreover, the proposed techniques provide tractable density evolution analysis for any coupling distribution as well as for probabilistic connections between variable and check nodes. It is worth mentioning that the decoding thresholds given by the proposed density evolution are no longer accurate for small L and/or very low rate ($R_d < 0.15$) SC-LDPC codes.

3.3 SC-LDPC Code Design for Gaussian MAC

We now present channel code design for Gaussian multiple access channel. Although the capacity region of general MAC is not known, the capacity region of Gaussian MAC (GMAC) has been known for many years. The *corner points* of GMAC's capacity region can be achieved through single user decoding with successive cancellation at the decoder [18]. To achieve any point of the capacity region, time sharing or rate splitting approach can be combined with single user decoding approach [56]. On the other hand, random coding with joint decoding is an alternative technique to obtain any point of the capacity region. LDPC codes are good candidates for this technique, since they provide capacity approaching performance with low complexity decoding. Design of practical GMAC code based on irregular LDPC codes were studied in [57, 58]. Due to the irregular LDPC codes, an extensive degree optimization is required to achieve a capacity approaching GMAC code.

Similar to irregular LDPC codes, SC-LDPC codes are also potential candidates for GMAC code due to their *threshold saturation* property over binary memoryless channel. Moreover, simple optimization procedure can be formulated for different channel conditions, since SC-LDPC codes are constructed from regular LDPC protographs. The threshold saturation of SC-LDPC codes for MAC channel was first shown in [59], where the MAC was modelled as a binary adder channel followed

by an erasure channel. Later, the threshold saturation of SC-LDPC codes was shown for GMAC in [60]. However, the density evolution analysis presented in [60] suffers from high computational complexity. In addition, for a given GMAC, [60] does not provide any explicit code design procedure.

In this section, we present a low complexity density evolution analysis for SC-LDPC GMAC code with the channel model presented in Subsection 2.1.3. A simple optimization procedure is presented to design a SC-LDPC GMAC code for a given GMAC. With numerical examples, we show that SC-LDPC GMAC codes are capacity approaching for wide range of design rates. We also show that SC-LDPC GMAC codes give better asymptotic performance than the optimized irregular LDPC GMAC codes reported in [57, 58].

3.3.1 Code Design

Following the code design proposed in [59], we design the SC-LDPC code for the 2-user GMAC. We consider that user- A takes a $(d_{v_A}, d_{c_A}, \gamma, L)$ -SC-LDPC code, denoted by code $\mathcal{C}_A$ of rate R_A , to encode its information bits. Similarly, user- B takes a $(d_{v_B}, d_{c_B}, \gamma, L)$ -SC-LDPC code, denoted by code $\mathcal{C}_B$ of rate R_B , to encode its information bits. Then both users simultaneously transmit their encoded bits through the 2-user GMAC. The factor graph used in the decoding process is shown in Fig. 3.7, where code $\mathcal{C}_A$ and code $\mathcal{C}_B$ are connected through state nodes (SNs) which contain the effect of MAC. A state node at position- i connects variable nodes at position- i of user- A and user- B . We refer to the combination of code $\mathcal{C}_A$ and code $\mathcal{C}_B$ as $(d_{v_A}, d_{v_B}, d_{c_A}, d_{c_B}, \gamma, L)$ -SC-LDPC GMAC code. Note that for equal rate scenario (i.e. for $R_A = R_B = R$), we consider $d_{v_A} = d_{v_B}$ and $d_{c_A} = d_{c_B}$.

3.3.2 Decoding Algorithm

At the MAC receiver, we consider the belief propagation (BP) decoding algorithm which is extended for GMAC scenario in [58]. We also consider a parallel scheduled decoding, where the state nodes first send its information⁴ to the variable nodes of code $\mathcal{C}_A$ and code $\mathcal{C}_B$. Then a single round of message passing between variable nodes and check nodes occurs in both users' code. After that an extrinsic information is sent back to the state nodes which updates the information contained in the state nodes.

⁴At the beginning, state nodes send the GMAC observation.

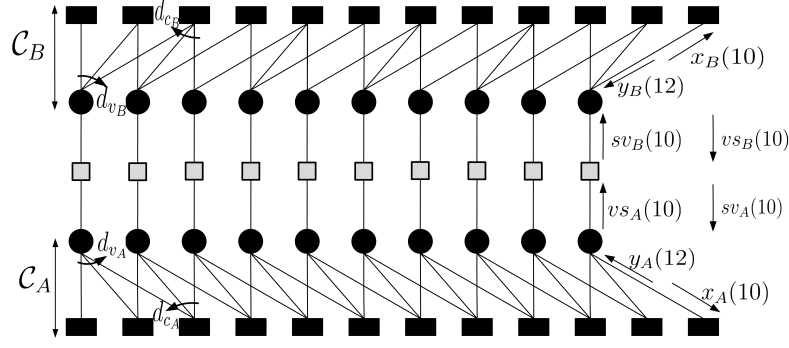


FIGURE 3.7: Factor graph of SC-LDPC code structure for 2-user MAC with $d_{v_A} = d_{v_B} = 3, d_{c_A} = d_{c_B} = 6, \gamma = 3$ and $L = 10$. The variable nodes of code $\mathcal{C}_A$ and the code $\mathcal{C}_B$ are connected through state nodes, which are represented by light gray square nodes. The degrees of the nodes are depicted in the left side, while the update messages used in the density evolution analysis are depicted in the right side of the chain.

Three out of four decoding update messages namely (i) variable node to check node (ii) check node to variable node and (iii) variable node to state node can be obtained from the standard BP decoding rules [more details can be found in [61]]. However, the decoding update rule for state node to variable node is rather new. Let $SV_{ji}^{(\ell)}$ denotes the message from state node- i to variable node- i of user $j \in \{A, B\}$ at iteration ℓ . Following [58], we get the following update rules:

$$SV_{Ai}^{(\ell)} = \log \frac{e^{\frac{2(v_i-1)}{\sigma_n^2} + VS_{Bi}^{(\ell)}} + 1}{e^{VS_{Bi}^{(\ell)}} + e^{-\frac{2(v_i+1)}{\sigma_n^2}}},$$

$$SV_{Bi}^{(\ell)} = \log \frac{e^{\frac{2(v_i-1)}{\sigma_n^2} + VS_{Ai}^{(\ell)}} + 1}{e^{VS_{Ai}^{(\ell)}} + e^{-\frac{2(v_i+1)}{\sigma_n^2}}},$$

where $VS_{ji}^{(\ell)}$ is the message from variable node- i to state node- i of user $j \in \{A, B\}$ at iteration ℓ and v_i is the channel observation corresponds to state node- i .

3.3.3 Density Evolution

We now consider the density evolution analysis of the SC-LDPC GMAC code. Unlike the symmetric channel scenario, transmission of all +1 BPSK codeword for both users cannot be assumed in the MAC density evolution, since the channel seen by each user is not symmetric. In [58], it has been shown that, to predict the asymptotic performance of 2-user GMAC code, it is sufficient to assume one user transmits all +1 BPSK codeword and the other user transmits a BPSK codeword with half of symbols are +1 and half of symbols are -1. Note that, this assumption

only affects the update equation for the message from state node to variable node. For more details, we refer the readers to [58]. In the density evolution, we consider the following notations at iteration ℓ .

- $x_j^{(\ell)}(i)$ denotes the mean of LLR message outgoing from the variable node at position i of user $j \in \{A, B\}$.
- $y_j^{(\ell)}(i)$ denotes the mean of LLR message outgoing from the check node at position i of user $j \in \{A, B\}$.
- $sv_j^{(\ell)}(i)$ denotes the mean of LLR message outgoing from the state node at position i to variable node at position i of user $j \in \{A, B\}$.
- $vs_j^{(\ell)}(i)$ denotes the mean of LLR message outgoing from the variable node at position i to state node at position i of user $j \in \{A, B\}$.

The above notations are also depicted in Fig. 3.7. Following [62] and [58], density evolution update equations for the SC-LDPC GMAC code are given by:

- **Check node to variable node:** For user $j \in \{A, B\}$

$$y_j^{(\ell)}(i) = \phi^{-1} \left[1 - \left\{ 1 - \phi \left(\mathcal{J}^{-1} \left(\frac{1}{\gamma} \sum_{k=0}^{\gamma-1} \mathcal{J} \left(x_j^{(\ell)}(i-k) \right) \right) \right) \right\}^{d_{c_j}-1} \right],$$

where $x_j^{(\ell)}(i) = \infty$ if $i \leq 0$ or $i > L$. We initialize above update equations with the following messages:

$$x_j^{(0)}(i) = \mathcal{J}^{-1} \left(\frac{1}{3} C_T(\sigma_n) \right) \quad \text{if } 0 < i \leq L,$$

where $C_T(\sigma_n)$ is the capacity of ternary input AWGN channel with input distribution $\frac{1}{4}\Delta_{-2} + \frac{1}{2}\Delta_0 + \frac{1}{4}\Delta_{+2}$ and noise variance σ_n^2 [58]. $C_T(\sigma_n)$ is given by

$$C_T(\sigma_n) = - \int_{-\infty}^{\infty} \Psi_{\sigma}(g) \log_2 \Psi_{\sigma}(g) dg - \frac{1}{2} \log_2 2\pi e \sigma^2,$$

where

$$\Psi_{\sigma_n}(g) = \frac{1}{\sqrt{32\pi\sigma^2}} \left(e^{-\frac{(g+2)^2}{2\sigma_n^2}} + e^{-\frac{(g-2)^2}{2\sigma_n^2}} + 2e^{-\frac{g^2}{2\sigma_n^2}} \right).$$

- **Variable node to state node:** For user $j \in \{A, B\}$

$$vs_j^{(\ell)}(i) = d_{v_j} \mathcal{J}^{-1} \left(\frac{1}{\gamma} \sum_{k=0}^{\gamma-1} \mathcal{J} \left(y_j^{(\ell)}(i+k) \right) \right).$$

- **State node to variable node:** Following [58],

$$\begin{aligned} sv_A^{(\ell)}(i) &= \mathcal{J}^{-1} \left(\frac{1}{3} \left(1 + \mathcal{J} \left(vs_B^{(\ell)}(i) \right) \right) C_T(\sigma_n^2) \right), \\ sv_B^{(\ell)}(i) &= \mathcal{J}^{-1} \left(\frac{1}{3} \left(1 + \mathcal{J} \left(vs_A^{(\ell)}(i) \right) \right) C_T(\sigma_n^2) \right). \end{aligned}$$

- **Variable node to check node:** For user $j \in \{A, B\}$

$$x_j^{(\ell+1)}(i) = sv_j^{(\ell)}(i) + (d_{v_j} - 1) \mathcal{J}^{-1} \left(\frac{1}{\gamma} \sum_{k=0}^{\gamma-1} \mathcal{J} \left(y_j^{(\ell)}(i+k) \right) \right).$$

We calculate the noise threshold σ_n^* of the $(d_{v_A}, d_{v_B}, d_{c_A}, d_{c_B}, \gamma, L)$ -SC-LDPC GMAC code by:

$$\sigma_n^* = \max \left\{ \sigma_n : \lim_{\ell \rightarrow \infty} x_A^{(\ell)}(i) = \infty \text{ and } x_B^{(\ell)}(i) = \infty, \forall i \right\}.$$

Using the above density evolution analysis, we present the optimization procedure in the next subsection.

3.3.4 Optimization

For a given 2-user GMAC (i.e. for a given σ_n^2), we now present the optimization procedure for the SC-LDPC GMAC code. For sake of simplicity, we first set $\gamma = d_{v_A}$. For equal power/equal rate scenario, the rate R can be maximized by minimizing $\frac{d_{v_A}}{d_{c_A}}$. Thus, we get the following optimization problem.

$$\text{Minimize } \frac{d_{v_A}}{d_{c_A}}$$

such that

$$\begin{aligned} x_A^{(\ell)}(i) &> x_A^{(\ell-1)}(i), & \forall i \text{ and } \ell = 1, 2, 3, \dots \\ x_B^{(\ell)}(i) &> x_B^{(\ell-1)}(i), & \forall i \text{ and } \ell = 1, 2, 3, \dots \end{aligned}$$

$$x_A^{(\ell_{\max})}(i) \rightarrow \infty, \quad \forall i$$

We assume that the final constraints of the above optimization problem are satisfied when $x_A^{(\ell_{\max})}(i)$ and $x_B^{(\ell_{\max})}(i) \forall i$ reach to a high value (for example 100), where $\ell_{\max}$ is a predefined value of maximum number of iterations. Note that the above optimization procedure is essentially determining thresholds of SC-LDPC GMAC codes with various d_{v_A} and d_{c_A} values. Moreover, fixing all the parameters except d_{v_A} and d_{c_A} limits the rate compatibility property of the SC-LDPC GMAC code. The optimization procedure can be made more robust by varying other parameters as well. However, inclusion of other parameters along with d_{v_A} and d_{c_A} will increase the complexity of the optimization procedure.

3.3.5 Numerical Results

Asymptotic Performance

From the density evolution analysis described in Subsection 3.3.3, we now present the asymptotic performance of SC-LDPC GMAC code. We fix $d_{v_A} = d_{v_B} = \gamma = 4$, $L = 200$ and vary d_{c_A} (or d_{c_B}) to obtain SC-LDPC GMAC codes of different rates. The rates, thresholds, and Shannon limits⁵ are summarized in Table 3.1. The gap between Shannon limit and threshold are also shown in Table 3.1. Note that we present the thresholds and the Shannon limits in terms of $\frac{E_b}{N_0}$ (dB)⁶.

From the table, we observe that there exists a narrow gap between the threshold of SC-LDPC GMAC code and their corresponding Shannon limit. We also observe that this gap decreases as the rate increases.

Comparison with Existing GMAC code

We now compare the performance of the SC-LDPC GMAC codes with the performance of optimized LDPC GMAC codes proposed in [58]. For the SC-LDPC GMAC code, first we consider $(d_{v_A}, d_{v_B}, d_{c_A}, d_{c_B}, \gamma, L) = (3, 3, 6, 6, 3, 200)$, which gives the code rate for each user $R = 0.4967$. From the density evolution analysis as shown in Subsection 3.3.3, we get the threshold of this SC-LDPC GMAC code as 2.15dB while the Shannon limit is 1.97dB. We notice that there exist a

⁵For a given rate R , the Shannon limit of noise standard deviation σ_n^{Sl} is calculated from $\sigma_n^{Sl} = C_T^{-1}(2R)$

⁶For a given σ_n and rate R , we calculate $\frac{E_b}{N_0}$ by $\frac{E_b}{N_0} = 10 \log_{10} \frac{1}{2R\sigma_n^2}$ dB.

TABLE 3.1: Threshold of SC-LDPC GMAC code for different rates. We consider $d_{v_A} = d_{v_B} = \gamma = 4, L = 200$ and vary d_{c_A} to obtain SC-LDPC GMAC codes of different rates. SL and TH stand for the Shannon limit and threshold, respectively.

d_{c_A}	Rate	TH (dB)	SL (dB)	Gap (dB)
6	0.3259	0.87	0.57	0.30
8	0.4944	2.14	1.94	0.20
10	0.5955	3.18	3.02	0.16
12	0.6629	4.18	4.06	0.12

0.18dB gap between the code threshold and the Shannon limit. For comparison, with rate 0.5, the threshold of the optimized irregular code proposed in [58] is 2.18dB, and the gap to Shannon limit is 0.19dB. With rate $R = 0.5955$, then we consider $(d_{v_A}, d_{v_B}, d_{c_A}, d_{c_B}, \gamma, L) = (4, 4, 10, 10, 4, 200)$, which gives a threshold of 3.18dB and the gap to Shannon limit is 0.16dB. With rate 0.6, the threshold of the optimized irregular code proposed in [58] is 3.60dB, and the gap to Shannon limit is 0.52dB. Thus, compared to optimized irregular LDPC relay codes, a better code design is possible with the SC-LDPC GMAC code with reduced optimization complexity. The above results are summarized in Table 3.2.

TABLE 3.2: Comparison between the optimized irregular LDPC GMAC code [58] and SC-LDPC GMAC code. SL and TH stand for the Shannon limit and threshold, respectively.

GMAC Codes	Rate	SL (dB)	TH (dB)	Gap (dB)
LDPC GMAC Code [58]	0.5	1.99dB	2.18dB	0.19dB
SC-LDPC GMAC code	0.4967	1.97dB	2.15dB	0.18dB
LDPC GMAC Code [58]	0.6	3.08dB	3.60dB	0.52dB
SC-LDPC GMAC code	0.5955	3.02dB	3.18dB	0.16dB

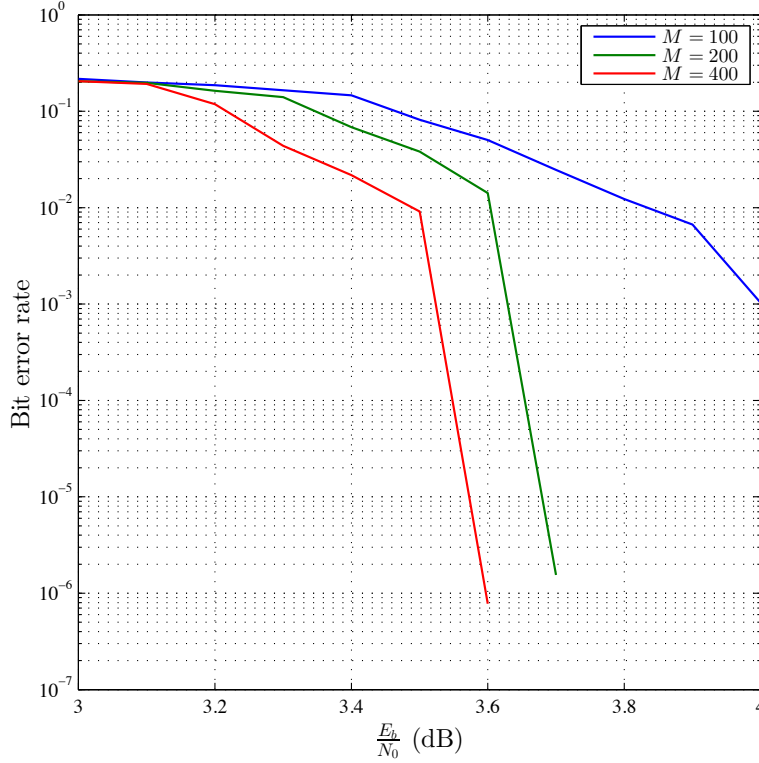


FIGURE 3.8: Bit error rate performance of the SC-LDPC GMAC code. In the simulation, we consider $d_{v_A} = d_{v_B} = 3$, $d_{c_A} = d_{c_B} = 6$, $\gamma = 3$ and $L = 200$. From density evolution, we get an asymptotic threshold 2.15dB, while the Shannon limit is 1.97dB.

Finite Length Performance

We now present the finite-length performance of the SC-LDPC GMAC code. In Fig. 3.8, we show the bit error probability of $(d_{v_A}, d_{v_B}, d_{c_A}, d_{c_B}, \gamma, L) = (3, 3, 6, 6, 3, 200)$ SC-LDPC GMAC code. Since we observed almost similar performance for both users, we only plot the finite-length performance of user-A. In the simulation, we consider $M = \{100, 200, 400\}$, which gives the corresponding block length for each user = $\{20000, 40000, 80000\}$. From the results, we observe that the finite-length performance of the proposed SC-LDPC GMAC code exhibits a threshold close to 3.1dB. It is worth mentioning that with the reported block length, the LDPC GMAC code in [58] gives better finite-length performance than the performances shown in Fig. 3.8. It is expected since finite-length performance of spatially coupled codes is not good for $M < 10000$.

3.4 Conclusion

In this chapter, we have presented the tools, developed for this thesis, which can be utilized to design and analysis of both block and anytime codes. Firstly, we have proposed an algorithm to estimate the finite-length behavior of LDPC codes in the waterfall region. From the comparison with simulation results, we have found that our algorithm has a good prediction accuracy for various channels without knowing the threshold or any other scaling parameters. In Chapter 4, we use the proposed algorithm to estimate the finite-length performance of anytime code.

Next, we have presented density evolution techniques to find the decoding threshold of SC-LDPC codes over BIAWGN channels. We have observed that although MET density evolution results in an accurate analysis, it suffers from high complexity and limited practicality. On the other hand, although our proposed density evolution techniques are slightly less accurate, they are easy to implement and have broader applications. Based on the proposed density evolution technique, we show the asymptotic analysis of anytime code over BIAWGN channel in Chapter 4.

Finally, we have presented the design and analysis of SC-LDPC MAC code while 2-user GMAC is considered. Through asymptotic results, we have observed that presented SC-LDPC MAC code exhibits a capacity approaching performance over BIAWGN channel. For a given 2-user GMAC, we have presented a simple optimization procedure. We have also shown that compare to the existing optimized MAC code, presented SC-LDPC MAC code leads to a better performance. Based on the presented low complexity analysis, we analyse anytime code over GMAC in Chapter 6.

Chapter 4

Anytime Codes for Point-to-Point Communication

In this chapter, we propose anytime channel code for point-to-point communication. The anytime characteristics of the proposed code is shown through density evolution analysis. Adaptive window decoding algorithms are proposed to ensure low complexity anytime decoder. We also present finite-length performance estimation algorithm and delay-exponent analysis over BEC.

This chapter is organized as follows. We first describe the system model for the P2P anytime communication in Section 4.1. In the next section, the design of anytime codes based on spatially coupled codes is proposed. In Section 4.3, the anytime characteristics of the proposed anytime codes are investigated over BEC. We present the algorithm for finite length performance estimation and propose a low complexity anytime decoder. We also present simulation results along with the performance comparison between proposed and existing anytime codes. We extend our analysis to BIAWGN channels in Section 4.4, where we show the anytime transmission properties of the proposed codes and develop the adaptive window decoding algorithms under belief propagation decoding.

4.1 System Model

Consider a streaming source, where at time t , a K -bit message $\mathbf{m}_t \in \{0, 1\}^K$ is produced. At time instant t , the anytime channel encoder produces a M -bit channel input $\mathbf{u}_t \in \{0, 1\}^M$ as a function of the available source information $\mathbf{m}_1^t =$

$[\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_t]$. For $t = 1, 2, \dots$, we define the anytime channel encoder by the function $\mathcal{E}_t(\mathbf{m}_1^t)$. The rate of the anytime channel encoder is $R = \frac{K}{M}$. The code sequence $\mathbf{u}_t$ is then transmitted over a noisy channel. The anytime channel decoder receives $\mathbf{v}_t$ (a noisy version of $\mathbf{u}_t$) at every time step t . The decoder estimates the transmitted messages using the decoding function $\hat{\mathbf{m}}_1^t(t) = \mathcal{D}_t(\mathbf{v}_1^t)$, i.e., the decoder produces an estimate $\hat{\mathbf{m}}_1^t(t)$ based on the current as well as all previously received messages. A block diagram of anytime transmission is illustrated in Fig. 4.1.

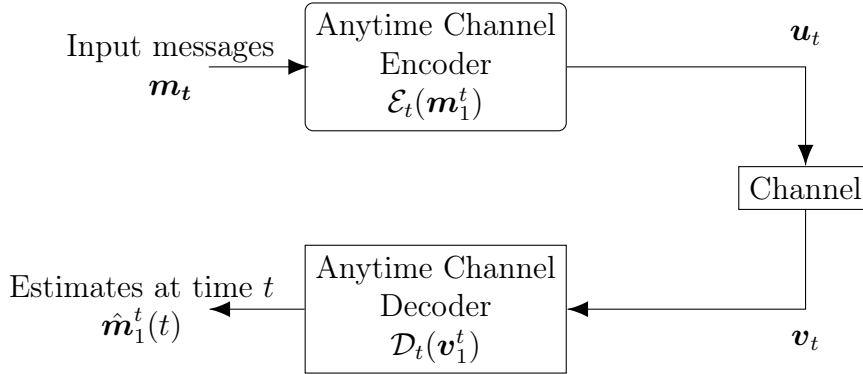


FIGURE 4.1: A block diagram illustrating anytime communication.

We now consider the estimate $\hat{\mathbf{m}}_i(t)$ of the message $\mathbf{m}_i$ at time $t \geq i$; the corresponding decoding delay for message $\mathbf{m}_i$ is $d = t - i$. The probability of error for the message i at time t is $P_e(t - i) = \Pr(\hat{\mathbf{m}}_i(t) \neq \mathbf{m}_i)$. For a given channel, the encoder-decoder pair $(\mathcal{E}_t, \mathcal{D}_t)$ is called *anytime code* if there exists a finite $\beta > 0$ such that [12]:

$$P_e(t - i) \leq \beta e^{-\alpha(t-i)}, \quad \forall (t - i) \geq 0, \quad (4.1)$$

where $\alpha > 0$ is known as the *anytime reliability*, *delay-exponent* or *anytime exponent*. The anytime reliability α specifies how fast the reliability of each message improves with delay. The inequality in (4.1) is the main property of an anytime code. Eq. (4.1) implies that the decoder is capable of estimating the message $\mathbf{m}_i$ at any delay $d \geq 0$ and the probability of error for that message will decay exponentially as the delay d approaches infinity. Control applications require the error probability of each message to decay to zero when the delay approaches infinity. In practice, due to complexity and hardware constraints, a message is declared reliable when the error probability is below a certain threshold (for example 10^{-10}). This threshold varies with applications. Thus, depending on a specific application, the anytime code should be designed with a given target threshold while maximizing the delay-exponent.

4.2 Spatially Coupled Codes as Anytime Codes

To achieve the anytime coding properties as described in Chapter 1, we bring the following modifications on the SC-LDPC code.

1. We allow large chain length L and large coupling length γ , so that the reliability of each message can be improved with large delays.
2. Instead of a uniform distribution of each connection from variable nodes to check nodes, we choose a distribution that favours connections to nearby nodes. Specifically, a variable node at position i is more likely to connect to check nodes close to position i . Equivalently, there will be fewer connections with check nodes that are positioned farther away from position i . In this way, we can get a quick improvement of error performance with small delay. This type of distribution can be achieved, for example, by using an exponential distribution¹. In this case, the probability of a connection between a variable node (VN) at position i and a check node (CN) at position j ($j \geq i$) can be written by:

$$P_r(k) = \frac{e^{-k\lambda}(1 - e^{-\lambda})}{1 - e^{-\gamma\lambda}}, \quad (4.2)$$

where λ is known as exponential rate parameter and $k = j - i$ is the distance between the connected check node and variable node. For $\gamma = \infty$, (4.2) becomes

$$P_r(k) = e^{-k\lambda}(1 - e^{-\lambda}).$$

We refer to the above modified spatially coupled codes as (d_v, d_c, λ) -anytime SCLDPC code. In the following sections, we present theoretical analysis and simulation results for the proposed anytime code. We also denote M as the encoded message length at each time unit and t to be the decoding time instant.

¹The rationale behind choosing the exponential distribution is that the resultant code allows a tractable analytical analysis.

4.3 Code performance over BEC

4.3.1 Asymptotic Analysis

For density evolution of anytime SC-LDPC codes, we assume an infinite number of variable nodes (i.e., $M = \infty$) at each position. Let $x^{(\ell)}(i, t)$ be the erasure probability of a message outgoing from a VN at position i in the ℓ^{th} iteration at time t . Applying the derivation techniques in [32], we get the following density evolution equation for any $t \geq i$ and $\gamma = \infty$.

$$x^{(\ell)}(i, t) = \epsilon \left(1 - \sum_{j=0}^{t-i} P_r(j) \left(1 - \sum_{k=0}^{\infty} P_r(k) x^{(\ell-1)}(i+j-k, t) \right)^{d_c-1} \right)^{d_v-1}, \quad (4.3)$$

where $x^{(\ell)}(i, t) = 0, \forall \ell$ when $i \leq 0$ and $x^{(0)}(i, t) = \epsilon$ when $0 < i \leq t$. As mentioned earlier, each variable node is connected probabilistically to the check nodes and hence the check nodes do not maintain a fixed degree. Thus the use of fixed check node degree in (4.3) result in an approximate density evolution. From (4.3), the probability that the variable node at position i has been erased can be computed by considering all of its connected edges. The erasure probability $P_e(i, t)$ of message $\mathbf{m}_i$ is

$$P_e(i, t) = \epsilon \left(1 - \sum_{j=0}^{t-i} P_r(j) \left(1 - \sum_{k=0}^{\infty} P_r(k) x^{(\infty)}(i+j-k, t) \right)^{d_c-1} \right)^{d_v}. \quad (4.4)$$

We present the numerical results from the above modified density evolution analysis. We consider $(d_v, d_c, \lambda) = (3, 6, 0.1)$ -anytime SC-LDPC code with $\gamma = \infty$. In Fig. 4.2, we present the erasure probabilities of three messages versus the decoding time instant. We observe that the erasure probability of each message decays exponentially with the decoding delay. Furthermore, the erasure performance of a given message is a time-shifted version of the others. Hence, the erasure performance of a message depends only on the decoding delay, not on its index. From the slope of the erasure probability performance shown in Fig. 4.2, we can numerically determine the achieved delay-exponent α . The anytime property of the proposed code can be proved by the following theorem.

Theorem 4.1. *Consider the anytime SC-LDPC code with exponential distribution of coupling connections as described in Section 4.2. The delay-exponent of the proposed (d_v, d_c, λ) -anytime SC-LDPC code is given by*

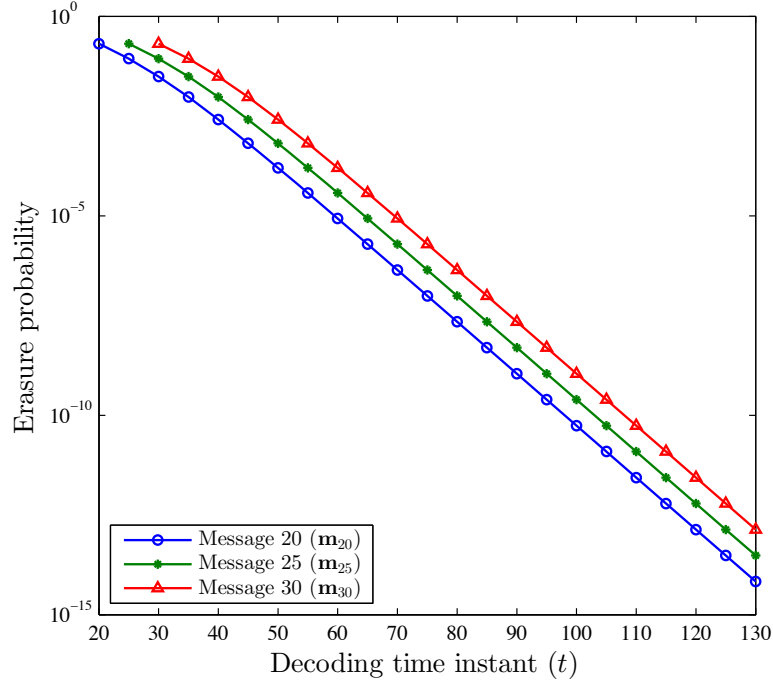


FIGURE 4.2: Asymptotic performance of proposed anytime code over binary erasure channel ($\epsilon = 0.3$).

$$\alpha = \lambda d_v, \quad (4.5)$$

when $\epsilon \leq \bar{\epsilon}_b$, where

$$\bar{\epsilon}_b \triangleq \max_{\beta \geq \epsilon} \min_{d \geq 0} \beta \left(e^{-\lambda} + \beta(d_c - 1)e^{\lambda(2-d_v)(d+1)}(1 - e^{-\lambda}) \left(\frac{(1 - e^{-\lambda})(1 - e^{\lambda(d_v-2)(d+2)})}{(1 - e^{-\lambda d_v})(1 - e^{\lambda(d_v-2)})} \right. \right. \\ \left. \left. + \left(e^{\lambda(d_v-1)} - \frac{1 - e^{-\lambda}}{1 - e^{-\lambda d_v}} \right) \frac{1 - e^{-2\lambda(d+2)}}{1 - e^{-2\lambda}} \right) \right)^{-d_v}.$$

The proof of the theorem is given in the Appendix A.1. It can be shown numerically that the proposed anytime code achieves the delay-exponent $\alpha = \lambda d_v$ whenever ϵ is below a threshold $\bar{\epsilon}$. We can consider $\bar{\epsilon}_b$ as a bound on the threshold $\bar{\epsilon}$. It is worth mentioning that there exists a large gap between $\bar{\epsilon}$ and $\bar{\epsilon}_b$ due to the limited tightness of the bounds used in the proof. For example, with a (3,6)-anytime SC-LDPC code and $\lambda = 0.1$, the delay-exponent is $\alpha = 0.3$, where $\bar{\epsilon}$ and $\bar{\epsilon}_b$ are 0.487 and 0.093, respectively. In Fig. 4.3, we compare the theoretical exponent

with the exponent obtained from the numerical results of density evolution. From Fig. 4.3, it is clear that the theoretically determined delay-exponent $\alpha = \lambda d_v$ closely matches the exponent of the decoded erasure probability measured from density evolution analysis results. From (4.5), we also observe that the delay-exponent increases linearly with the increment of λ or d_v . However, in practice (i.e., for the finite-length case), increasing d_v may introduce short cycles in the code. On the other hand, increasing λ leads to fewer connections to the distant protographs. Thus, increasing d_v or λ may result in an error floor at large delays. The later effect is shown in Section 4.3.4 through simulation results.

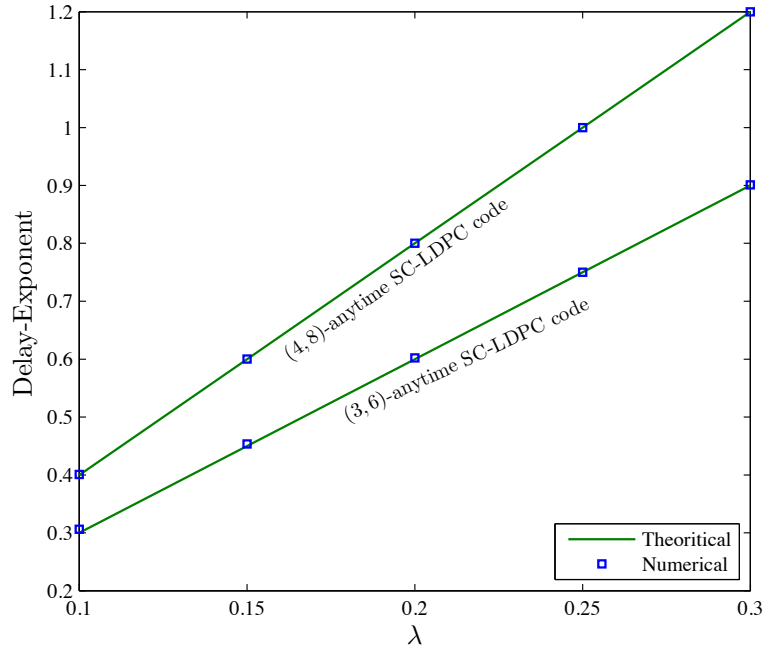


FIGURE 4.3: Comparison between the theoretical exponent ($\alpha = \lambda d_v$) and the numerical exponent.

Remarks: In the above delay-exponent analysis, infinite number of channel use in a message is considered. Thus theoretically, an infinite delay-exponent is achievable. The finite delay-exponent in the above theorem is due to the structure of the code. For the anytime code proposed in [10], the variable node degree grows to infinity, and the delay-exponent is limited by the channel coefficient. However, in our case, the variable node degree is fixed to a finite value, and the delay-exponent is limited by the parameters λ and d_v .

4.3.2 Finite-Length Estimation

In this section, we present an algorithm to estimate the finite-length behaviour of the proposed anytime SCLDPC code. For this purpose, we utilize the estimation algorithm shown in Chapter 3, which is developed to estimate the finite-length performance of LDPC codes.

Recall that, we define a random variable E , the empirical bit erasure probability observed by a codeword. Then NE has a binomial distribution $B(N, \epsilon)$ [54]. For large N , the distribution of E can be approximated by a Gaussian distribution $\mathcal{N}(\epsilon, \frac{\epsilon(1-\epsilon)}{N})$ with mean ϵ and variance $\frac{\epsilon(1-\epsilon)}{N}$. Conditioning on a channel realization (realised erasure probability), the proposed algorithm first finds the conditional decoded erasure probability from density evolution. Then the erasure probability can be obtained by marginalizing over the channel realizations. Using [63], we estimate the bit erasure rate of finite-length anytime SCLDPC codes as a function of the delay. To estimate the erasure probability of message $\mathbf{m}_i$, we assume that $\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_{i-1}$ messages are recovered. Thus, the erasure probability of message $\mathbf{m}_i$ at decoding time instant $t + i$ is equal to the erasure probability of message $\mathbf{m}_1$ at decoding time instant $t + 1$.

- Let $P_{e|k}(1, t)$ denote the erasure probability of message $\mathbf{m}_1$ after decoding at time t , conditioning on $k \in [0, (t - i + 1)M]$ erasures realized at the channel. We determine $P_{e|k}(1, t)$ via the following steps.
 - For k erasures realized at the channel, the empirical channel erasure probability is $\delta_k = \frac{k}{(t-i+1)M}$.
 - Update $x_k^{(\ell)}(1, t)$ using the density evolution update equation as mentioned in Subsection 4.3.1, where ϵ and $x^{(\ell)}(i, t)$ are replaced by δ_k and $x_k^{(\ell)}(1, t)$, respectively. Here $x_k^{(\ell)}(1, t)$ denotes the corresponding erasure probability conditioning on k erasures realized at the channel.
 - Find the conditional erasure probability $P_{e|k}(1, t)$ using (4.4).
- Marginalize $P_{e|k}(1, t)$ over k to get the erasure probability of $\mathbf{m}_1$.

$$P_e(1, t) = \sum_k P_b(k) P_{e|k}(1, t), \quad (4.6)$$

where $P_b(k)$ is the probability of having k erasures from the channel, which can be obtained from the binomial distribution $B((t - i + 1)M, \epsilon)$ of realized channel erasure probability.

Now we present the results according to the aforementioned finite-length analysis. To estimate the performance of our proposed anytime code, we consider $(d_v, d_c, \lambda) = (3, 6, 0.1)$ -anytime SC-LDPC code with $M = 32$. Fig. 4.4 illustrates the estimation and simulation performance of the proposed anytime code. It can be seen that there exists a slight gap between our prediction and the simulation results, since the finite-length analysis does not consider cycles in the code and due to the assumption that all the previous messages are decoded correctly. Otherwise, good prediction on erasure performance can be obtained from the proposed estimation algorithm.

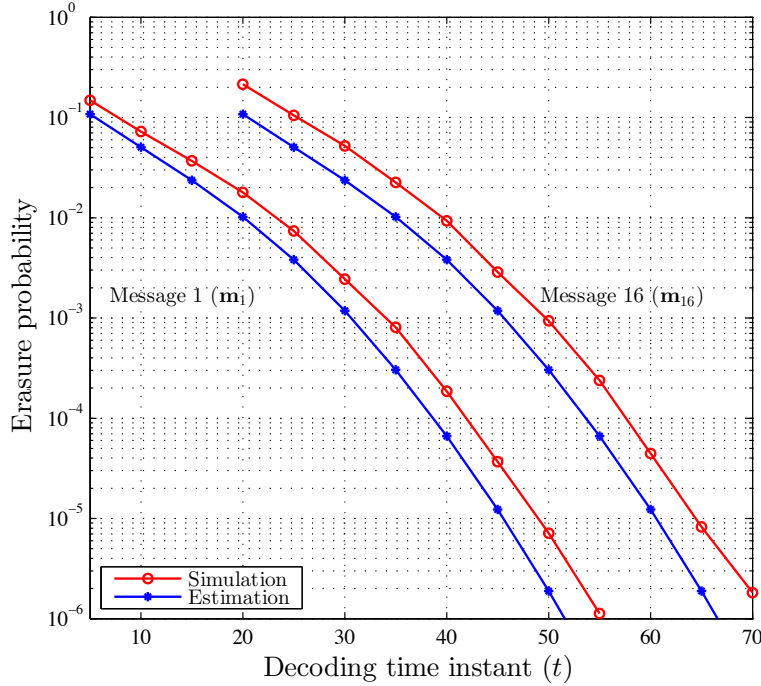


FIGURE 4.4: Comparison between the simulation results and our estimation over BEC ($\epsilon = 0.4$) with $M = 32$.

4.3.3 Adaptive Window Decoding

In this section, we present a low complexity adaptive window decoding (AWD) algorithm for anytime receivers, where the window size depends only on the number of erased bits. Note that the adaptive window decoding technique for the BEC is similar to the peeling decoder [64]. However, for the sake of completeness and to help explaining the AWD extension for BIAWGN channels, we describe the AWD for BEC in the following.

According to the anytime code construction mentioned in Section 4.2, the resultant code structure leads to a lower triangular unterminated check matrix $\mathbf{H}$. The

lower triangular property of the check matrix enables us to perform decoding with a portion of $\mathbf{H}$, i.e., a portion which is associated with the received bits. We denote the active portion of the check matrix $\mathbf{H}$ at time t as $\mathbf{H}_t$. Unterminated check matrix $\mathbf{H}$ and active check matrix $\mathbf{H}_t$ are illustrated in Fig. 4.5.

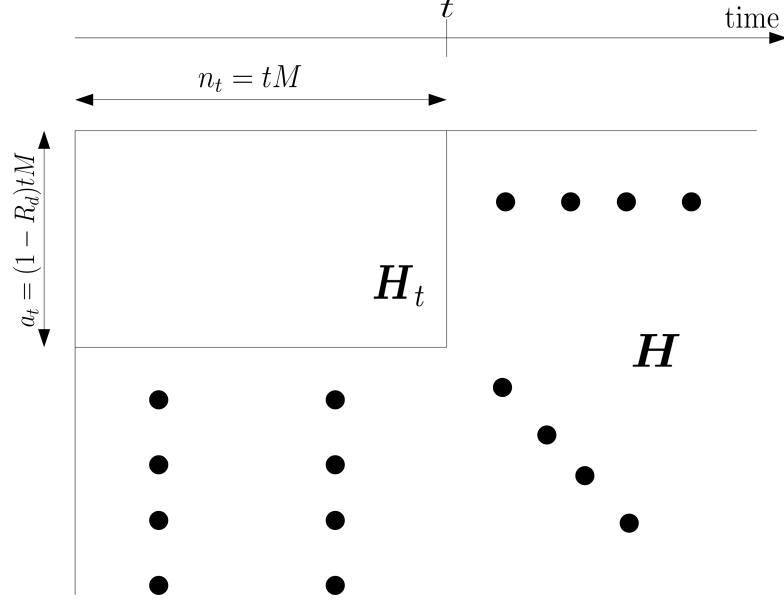


FIGURE 4.5: Illustration of unterminated check matrix $\mathbf{H}$ and active check matrix $\mathbf{H}_t$.

Let $a_t \times n_t$ be the size of the active check matrix $\mathbf{H}_t$. We define the parity vector $\mathbf{p}_t = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{a_t}] \in \{0, 1\}^{a_t}$, where $\mathbf{p}_i$ is the parity of coded bits associated to the i^{th} row of the check matrix. We initialize $\mathbf{p}_t$ with all zeros. At each decoding time instant, the adaptive window decoder removes the recovered coded bits and the corresponding columns of $\mathbf{H}_t$. If the value of a recovered coded bit j is 1, then we flip the value of associated parity components, i.e., $\mathbf{p}_i = \mathbf{p}_i \oplus [\mathbf{H}_t]_{ij}b_j$, where $[\mathbf{H}_t]_{ij}$ denotes the entry of $\mathbf{H}_t$ at the i^{th} row and j^{th} column and $b_j (j \in \{1, 2, \dots, n_t\})$ denotes the value of the j^{th} coded bit. Then the all-zero rows of the check matrix are removed. Therefore we get a reduced size check matrix $\mathbf{H}_r$ and the decoding algorithm runs on the corresponding Tanner graph of $\mathbf{H}_r$. From the conventional erasure decoding algorithm [17], the check node update rule is modified in the following manner. In the first iteration, we recover all the variable nodes that are connected with a check node of degree one. Then at each iteration, a check node C (let C corresponds to i^{th} row of $\mathbf{H}_t$) sends to a connected variable node V an erasure if it receives at least one erasure from the connected variable nodes other than V ; otherwise C sends to V the sum (mod 2) of $\mathbf{p}_i$ and incoming messages from all connected variable nodes other than V . The adaptive window decoding scheme for the BEC is summarized in Algorithm 1.

Algorithm 1 Adaptive window decoding algorithm for BEC

```

for  $t = 1, 2, \dots, \infty$  do
  1. Take first  $tM$  columns and first  $(1 - R_d)tM$  rows from the parity check
    matrix  $\mathbf{H}$  to form  $\mathbf{H}_t$  and initialize  $\mathbf{p}_t$  with all zeros.
  2. Modify  $\mathbf{H}_t$  and  $\mathbf{p}_t$  based on recovered coded bits.
  for  $i = 1$  to  $tM$  do
    if coded bit- $i$  is recovered then
      if coded bit- $i = 1$  then
        Flip the values of parity components associated with coded bit- $i$ .
      end if
      Remove the coded bit- $i$  and corresponding column of  $\mathbf{H}_t$ .
    end if
  end for
  3. Remove the rows with all zeros from the check matrix to obtain  $\mathbf{H}_r$ .
  4. Run the decoder with erased bits, parity vectors and reduced check matrix
     $\mathbf{H}_r$ .
  5. Find new recovered bits based on the information returned by the decoder.
end for

```

The aforementioned adaptive decoding algorithm is optimal for the BEC since the decoder can accurately detect recovered bits. This algorithm gives a significant lower decoding complexity compared to the expanding window decoding technique, while the error performance remains the same. In the case of expanding window decoding, the complexity of the decoder scales linearly with Mt . On the other hand, the average complexity of our algorithm scales linearly with $\mathcal{W}_{\text{av}}$, where $\mathcal{W}_{\text{av}}$ is the average decoding window length that contains the erasure bits of the currently received message and all the non-decodable bits among the previously received messages. For message size $M = 40$, Fig. 4.6 shows the comparison between expanding window decoding (EWD) and AWD techniques. For AWD, we observe that the average decoding window length approaches a constant ($\mathcal{W}_{\text{av}} \simeq 212$ for $\epsilon = 0.4$) while the decoding window length grows linearly for EWD technique. As expected, $\mathcal{W}_{\text{av}}$ increases as the channel erasure probability increases. We noticed that $\mathcal{W}_{\text{av}}$ is finite whenever the erasure probability of each message decays with delay. In Fig. 4.6, we also present the estimated decoding window length for the AWD technique by utilizing the finite-length performance estimation as described in earlier subsection. Using (4.6), at time t , the estimated decoding window length is $M \sum_{i=1}^{t-1} P_e(i, t-1) + M\epsilon$.

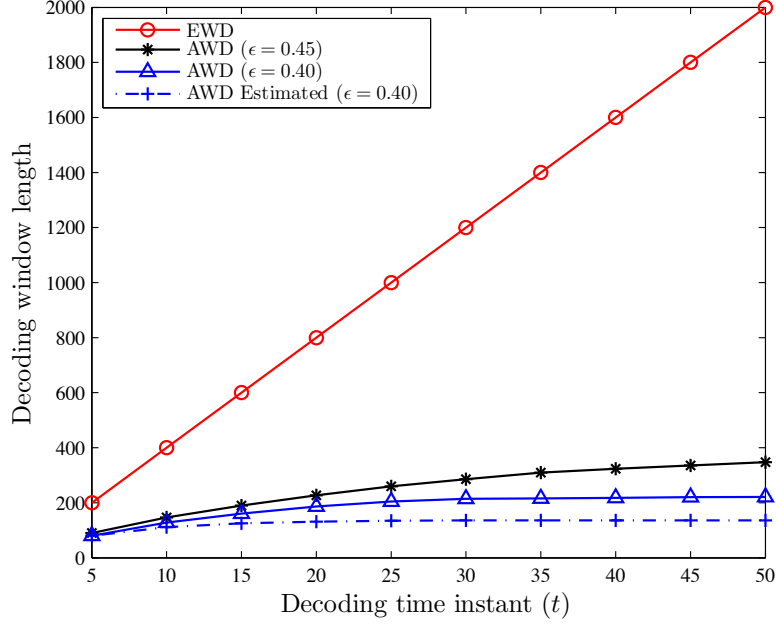


FIGURE 4.6: Comparison between the expanding window decoding (EWD) and adaptive window decoding (AWD) over BEC.

4.3.4 Numerical Results

In this section, we present the error performance through simulation results and show the effects of different design parameters on the proposed anytime code. For the simulations, we construct a $(d_v, d_c, \lambda) = (3, 6, \lambda)$ -anytime SC-LDPC code as described in Section 4.2. Note that we have also simulated our proposed code using $(d_v, d_c, \lambda) = (4, 8, \lambda)$ -anytime SC-LDPC code and observed the same characteristics as explained in the following sections. In the finite-length code construction, we do not allow the check nodes to have a higher degree than d_c .

Comparison

We have compared the performance of our proposed anytime code with the existing anytime code [10] over a binary erasure channel. For the code of [10], the finite-length performance of a particular message depends on its index/position. More precisely, the level of error floor of a message gets worse as the index of the message increases. For our proposed code, except for the first few messages, the finite-length performance of a particular message is independent of its index/position. We have observed that the performances of the first few messages are slightly better than the later ones due to the boundary effect.

For numerical comparison, we constructed the code of [10] by choosing sub-matrices $\mathbf{B}_0 = [1 \ 1]$ and $\mathbf{B}_i = [1 \ 0]$, where $i > 0$. For our proposed code, we consider exponential rate parameter $\lambda = 0.1$ and coupling length $\gamma = 50$. The design rate for the both codes is $R_d = \frac{1}{2}$. In the simulation, we consider $\epsilon = 0.4$ and $M = \{24, 40\}$. Fig. 4.7 shows a comparison between our proposed code and a systematic LDPC convolutional code proposed in [10], where we plot the average erasure performance of message $\mathbf{m}_6$ versus the decoding delay². Each curve is obtained by averaging over 10^6 simulation runs. Although the latter code provides better performance at low delays, it exhibits a very high error floor where the error probability does not decay with the delay. It is worth mentioning that the constructed code with above sub-matrices [10] leads to a good performance for low channel erasure probability ($\epsilon = 0.3$) and/or larger message length M . Thus, it may be possible to modify the code for worse channels like $\epsilon = 0.4$, to have better performance than that reported in Fig. 4.7. However, to the best of our knowledge such a modification will result in a complex code design. On the other hand, in Fig. 4.7, our proposed code maintains an exponential decay of the erasure probability at large delays. Besides, our code also gives a good performance for lower channel erasure probability.

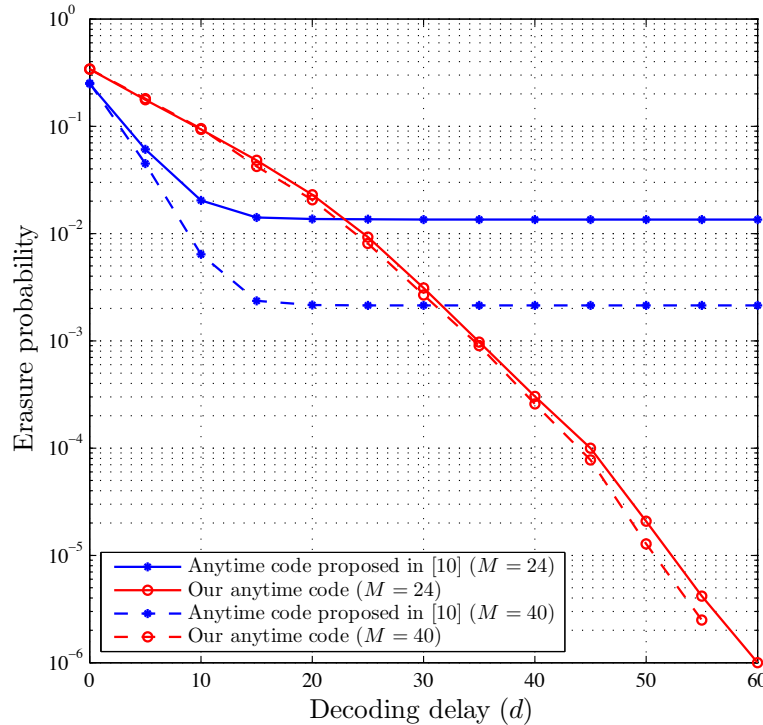


FIGURE 4.7: Performance comparison between our proposed anytime code and systematic LDPC convolutional code [10] with $\epsilon = 0.4$.

²For our proposed code, we observed that the performance gets independent of the position from message $\mathbf{m}_6$. Hence, we choose to compare the results for $\mathbf{m}_6$.

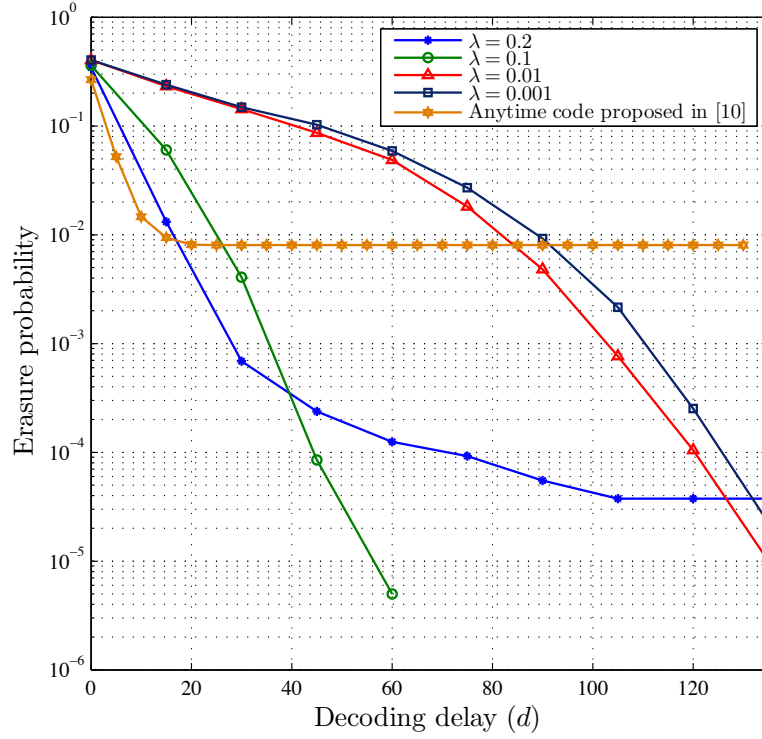


FIGURE 4.8: Effect of λ with $\gamma = \infty$ and $\epsilon = 0.42$. Performance of systematic LDPC convolutional code [10] with sub-matrices $B_0 = [1 \ 1]$ and $B_i = [1 \ 0]$, where $i > 0$ is also shown in the figures. We show the average erasure performance of message $\mathbf{m}_6$ with $M = 40$.

Effect of Exponential Rate Parameter λ

Fig. 4.8 shows the behavior of the erasure probability of a message for different values of λ . For larger λ , connections or couplings are more concentrated on nearby protographs. As a result, each message is recovered quickly as soon as the nearest messages are received. However, the decay of the error probability at large delays is slow as there are few connections with protographs farther in the future. On the other hand, with small λ , although the decay of the probability of error is slow at first, the error performance improves significantly at large delays. Fig. 4.8 illustrates the trade-off between λ and decay of error probability. From the results, it is clear that $\lambda = 0.1$ is a good choice at high channel erasure probability and low error floor requirement, since the decoded erasure probability decays reasonably fast while exhibiting no error floor at erasure probability 10^{-5} .

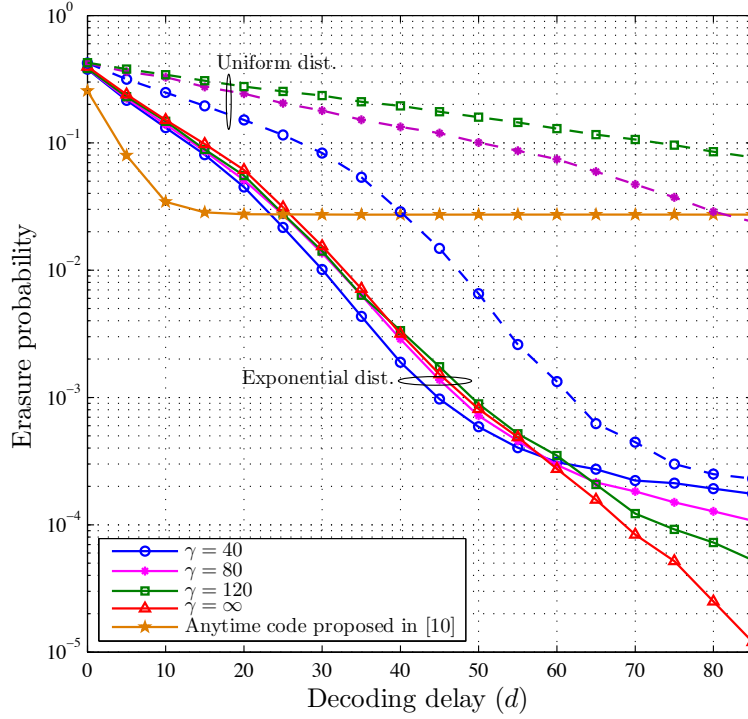


FIGURE 4.9: Effect of γ at $\epsilon = 0.44$, where solid line and dashed line are for the exponential ($\lambda = 0.1$) and uniform distribution, respectively. Performance of systematic LDPC convolutional code [10] with sub-matrices $B_0 = [1 \ 1]$ and $B_i = [1 \ 0]$, where $i > 0$ is also shown in the figures. We show the average erasure performance of message $\mathbf{m}_6$ with $M = 40$.

Varying Coupling Length γ

We now vary the coupling length while keeping the other parameters fixed. Fig. 4.9 shows the simulation results for different coupling length. When γ is small, the resultant code becomes a convolutional code with small memory length and thus the error performance levels out with increasing delay. On the other hand, a large γ enables a wider coupling connections between the protographs. From the simulation results, it is clear that an infinite coupling length is necessary for our proposed code. To support the exponential distribution, we also present the performance of our proposed code with uniform distribution of coupling connection. The proposed code with uniform distribution exhibits a very slow decay of erasure probability compared with the code with exponential distribution. Moreover, a uniform distribution is not feasible with $\gamma = \infty$. Thus, a uniform distribution of coupling connections in the proposed code design is not suitable for the anytime coding scenario.

4.4 Extension to the BIAWGN Channel Scenario

4.4.1 Asymptotic Analysis

In this section, we investigate the anytime characteristics of the proposed code over BIAWGN channels through density evolution. Following the low complexity density evolution of SC-LDPC codes as described in Chapter 3 (see section 3.2), we implement the density evolution of the proposed anytime code. Let us define the following notations:

- $x^{(\ell)}(i, t)$ and $y^{(\ell)}(i, t)$ are the mean values of LLR messages outgoing from a VN and a CN, respectively at position i in the ℓ^{th} iteration when t messages are received.
- $P_e(i, t)$ is the error probability of a VN at position i at decoding time instant t .

Using the above definitions, we get the following density evolution update equations for our proposed (d_v, d_c) -anytime SC-LDPC code:

- Check node update:

$$y^{(\ell)}(i, t) = \phi^{-1} \left[1 - \left\{ 1 - \phi \left(\mathcal{J}^{-1} \left(\sum_{k=0}^{\infty} P_r(k) \mathcal{J} (x^{(\ell)}(i - k, t)) \right) \right) \right\}^{d_c - 1} \right],$$

where $x^{(\ell)}(i, t) = \infty, \forall \ell$ when $i \leq 0$ and $x^{(0)}(i, t) = \frac{2}{\sigma_n^2}$ when $0 < i \leq t$.

- Variable node update:

$$x^{(\ell+1)}(i, t) = \frac{2}{\sigma_n^2} + (d_v - 1) \mathcal{J}^{-1} \left(\sum_{k=0}^{\infty} P_r(k) \mathcal{J} (y^{(\ell)}(i + k, t)) \right).$$

- Bit error rate:

$$P_e(i, t) = Q \left(\sqrt{\frac{1}{2} \left(\frac{2}{\sigma_n^2} + d_v \mathcal{J}^{-1} \left(\sum_{k=0}^{\infty} P_r(k) \mathcal{J} (y^{(\infty)}(i + k, t)) \right) \right)} \right).$$

Using the above described density evolution analysis, we plot the bit error rate performance in Fig. 4.10. We consider a $(d_v, d_c, \lambda) = (3, 6, 0.1)$ -anytime SC-LDPC

code. Similar to the performance over the BEC, we observe the anytime behaviour of the proposed code over the BIAWGN channel.

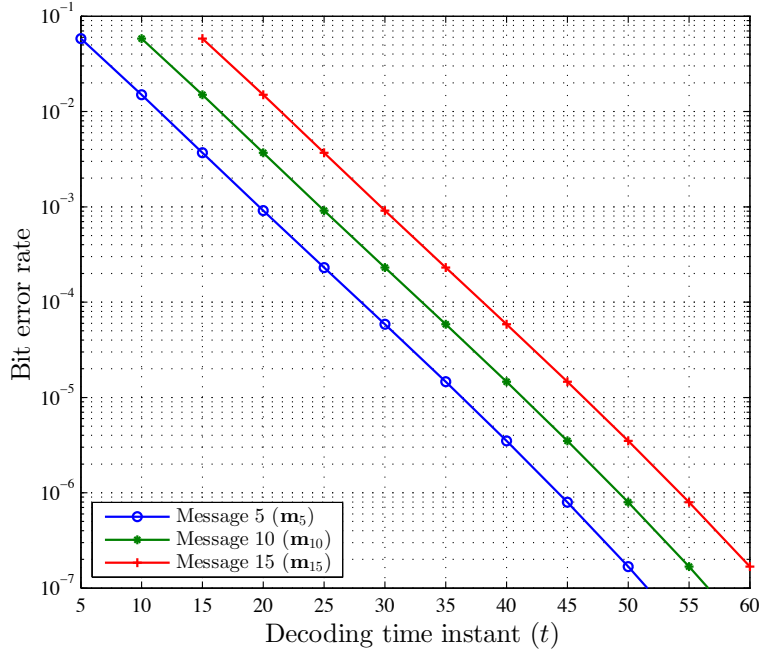


FIGURE 4.10: Asymptotic performance of proposed anytime code over BIAWGN channel ($\frac{E_b}{N_0} = 10 \log \frac{1}{\sigma_n^2} = 3\text{dB}$).

4.4.2 Performance of Finite-Length Codes

Through simulation results, we now present the finite-length error performance of the proposed anytime code over BIAWGN channels and compare it with the anytime code of [10]. For our code, we consider a $(d_v, d_c, \lambda) = (3, 6, 0.1)$ -anytime SC-LDPC code with coupling length $\gamma = \infty$. We construct the anytime code of [10] with sub-matrices $\mathbf{B}_0 = \begin{bmatrix} 1 & 1 \end{bmatrix}$ and $\mathbf{B}_i = \begin{bmatrix} 1 & 0 \end{bmatrix}$, where $i > 0$. In the simulation, we consider $M = \{24, 40\}$ and a BIAWGN channel with $\frac{E_b}{N_0} = 2\text{dB}$. The bit error rate performance comparison between the two anytime codes is shown in Fig. 4.11, which gives a performance characteristics similar to that of the performance over the BEC. In the figure, we show the average error performance of message $\mathbf{m}_6$, where each curve is obtained by averaging over 10^6 simulation runs.

4.4.3 Adaptive Window Decoding

Similar to the approach described in Subsection 4.3.3, we propose adaptive window decoding algorithms for BIAWGN channels, where the decoders deal with the

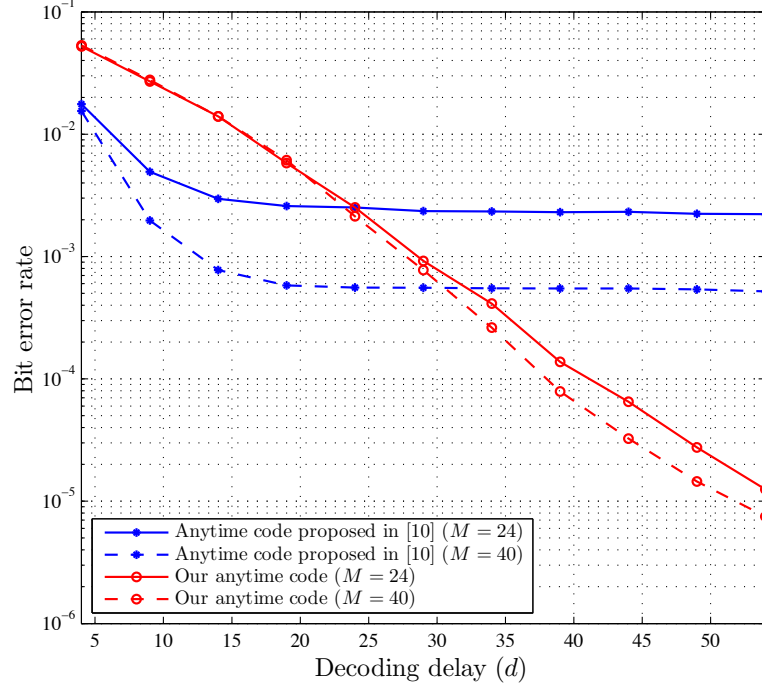


FIGURE 4.11: Bit error rate comparison between our proposed anytime code and systematic LDPC convolutional code [10] at $\frac{E_b}{N_0} = 2\text{dB}$.

unrecovered coded bits only. In the case of a BEC, a bit can be declared as recovered with full reliability. However, this is not the case for BIAWGN channels. We declare a code bit as recovered if a certain reliability threshold is met, and a recovered bit can be declared unrecovered at a later stage if the bit does not satisfy certain conditions (these conditions will be defined at a later stage). This results in a shrinkage and expansion of the decoding window length based on the state of recovered bits.

Recall that $\mathbf{H}_t$ is the active check matrix of size $a_t \times n_t$ and $\mathbf{p}_t$ is the vector of parity components. At the first decoding time instant (i.e., at $t = 1$), we run the conventional BP decoder with all received bits, their corresponding channel LLRs and active check matrix $\mathbf{H}_1$. From $t = 2$, we apply the adaptive window decoding technique similar to the algorithm as shown in Subsection 4.3.3. Using the information returned from the decoder, we first search for the recovered variable nodes or coded bits. At each time step, we remove the recovered coded bits and the corresponding columns of $\mathbf{H}_t$. If the value of the recovered bit is 1, we flip the value of the associated parity components. Rows of the matrix $\mathbf{H}_t$ with all zeros are also removed to obtain the reduced check matrix $\mathbf{H}_r$. The extrinsic message sent from check node j to variable node i at iteration ℓ is given by³

³Note that, we consider the positions of the variable nodes and check nodes with respect to the columns and rows of the active check matrix $\mathbf{H}_t$.

$$G_{j,i}^{(\ell)} = 2(1 - 2\mathbf{p}_j) \tanh^{-1} \left(\prod_{i' \in \mathbb{S}_v(j), i' \neq i} \tanh \left(\frac{F_{j,i'}^{(\ell)}}{2} \right) \right).$$

We run the modified BP decoder with $\mathbf{H}_r$, channel LLRs of unrecovered bits and $\mathbf{p}_t$. Now the question is how to declare a coded bit as recovered! To illustrate the procedure, we use the following notations:

- $\text{LLR}_d(i)$ denotes the LLR value returned by the decoder for coded bit i . We calculate $\text{LLR}_d(i)$ at the last iteration of the BP decoding process by,

$$\text{LLR}_d(i) = \sum_{j \in \mathbb{S}_c(i)} G_{j,i}^{(\ell_{\max})} + r_i,$$

where $\ell_{\max}$ is the maximum iteration number.

- Let $\mathbf{s}_t = [\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{a_t}] \in \{0, 1\}^{a_t}$ be the vector of syndrome components. A syndrome component $\mathbf{s}_i$ corresponds to the i^{th} row of $\mathbf{H}_t$, which determines whether the i^{th} check equation is satisfied. Later, two methods are proposed to determine the syndrome component.
- Let LLR_{th} be a predefined LLR threshold.

At each time step, after the decoding operation, we determine whether a bit is recovered based on the LLR value returned by the decoder, the syndrome components and the LLR threshold. We consider a coded bit i as recovered, if (i) $|\text{LLR}_d(i)| > \text{LLR}_{\text{th}}$ and (ii) all the syndrome components that involve bit i are zero (i.e., $\mathbf{s}_j = 0$ for all $j \in \mathbb{S}_c(i)$). The later condition ensures that all the check equations involving bit i are satisfied. This condition causes the backward expansion of the decoding window length by changing the status of a recovered bits to unrecovered. We propose the following two methods to find the value of the syndrome components.

Method I (AWD-I)

We first make a hard decision on the LLR value returned by the decoder. Thus, b_i , the value of coded bit i , is given by

$$b_i = \begin{cases} 0 & \text{if } \text{LLR}_d(i) \geq 0 \\ 1 & \text{otherwise} \end{cases}.$$

Let $\mathbf{b}_t = [b_1, b_2, \dots, b_n] \in \{0, 1\}^n$ be the hard decision on the codeword vector. Then we calculate the syndrome components by $\mathbf{H}_t \mathbf{b}_t^T = \mathbf{s}_t$.

Method II (AWD-II)

We define $\text{LLR}_s(j)$ as the LLR value associated with the j^{th} row of check matrix $\mathbf{H}_t$. If the j^{th} row of $\mathbf{H}_t$ is present in $\mathbf{H}_r$, then the value of $\text{LLR}_s(j)$ is calculated in the last iteration of the modified BP decoder by the following equation:

$$\text{LLR}_s(j) = 2(1 - 2\mathbf{p}_j) \tanh^{-1} \left(\prod_{i \in \mathbb{S}_v(j)} \tanh \left(\frac{F_{j,i}^{(\ell_{\max})}}{2} \right) \right).$$

We find the value of a syndrome component $\mathbf{s}_j$ based on $\text{LLR}_s(j)$ and an arbitrary chosen threshold LLR_{chk} such that,

$$\mathbf{s}_j = \begin{cases} 0 & \text{if } \text{LLR}_s(j) > \text{LLR}_{\text{chk}} \\ 1 & \text{otherwise} \end{cases}.$$

The adaptive window decoding scheme for the BIAWGN channel is summarized in Algorithm 2. In Fig. 4.12 and Fig. 4.13, we present the performance and complexity comparison between expanding window decoding (EWD) [10] and adaptive window decoding techniques (AWD-I and AWD-II) over BIAWGN channels. Similar to the AWD over BEC, we observe a finite average decoding window length with the AWD over BIAWGN channel. However, for BIAWGN channels, the performance of AWD is worse than that of the expanding window decoding technique. With the same LLR_{th} , AWD-II achieves a better error performance with higher complexity. We have observed that increasing LLR_{th} enhances the performance of the adaptive window decoder at the cost of increasing the average decoding window length. With the adaptive window decoder, the probability density function

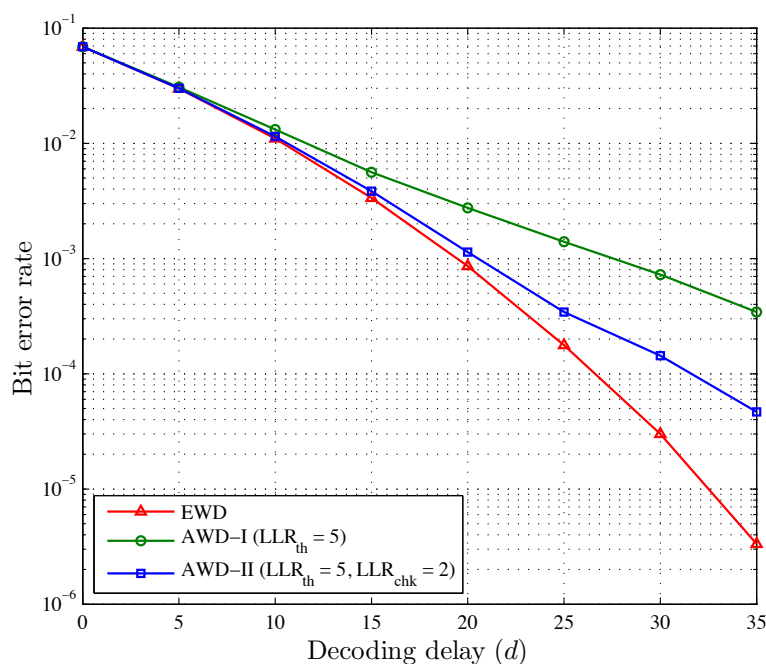


FIGURE 4.12: Performance comparison between expanding window decoding (EWD) and adaptive window decoding algorithms (AWD-I and AWD-II) with $M = 30$ and $\frac{E_b}{N_0} = 3\text{dB}$.

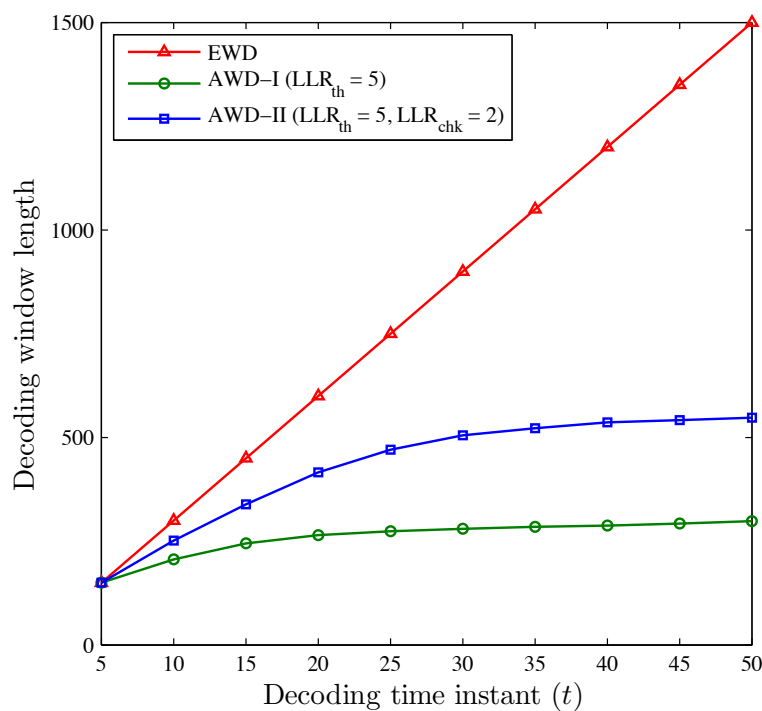


FIGURE 4.13: Complexity comparison between expanding window decoding (EWD) and adaptive window decoding algorithms (AWD-I and AWD-II) with $M = 30$ and $\frac{E_b}{N_0} = 3\text{dB}$.

Algorithm 2 Adaptive Window Decoding Algorithm for BIAWGN Channel

```

for  $t = 2, 3, \dots, \infty$  do
  1. Take first  $tM$  columns and first  $(1 - R_d)tM$  rows from the parity check
    matrix  $\mathbf{H}$  to form  $\mathbf{H}_t$  and initialize  $\mathbf{p}_t$  with all zeros.
  2. Modify  $\mathbf{H}_t$  and  $\mathbf{p}_t$  based on recovered coded bits.
  for  $i = 1$  to  $(t - 1)M$  do
    if coded bit- $i$  is recovered then
      if coded bit- $i = 1$  then
        Flip the values of parity bits associated with coded bit  $i$ .
      end if
      Remove the corresponding column of  $\mathbf{H}_t$  and coded bit  $i$ .
    end if
  end for
  3. Remove the rows with all zeros from the check matrix to obtain  $\mathbf{H}_r$ .
  4. Run the modified BP decoder with  $\mathbf{H}_r$ , channel LLRs of unrecovered bits
    and parity vectors.
  5. Find the syndrome components.
  6. Search for the recovered coded bits (for  $i = 1, 2, \dots, tM$ ) using returned
    LLR value and syndrome components.
end for

```

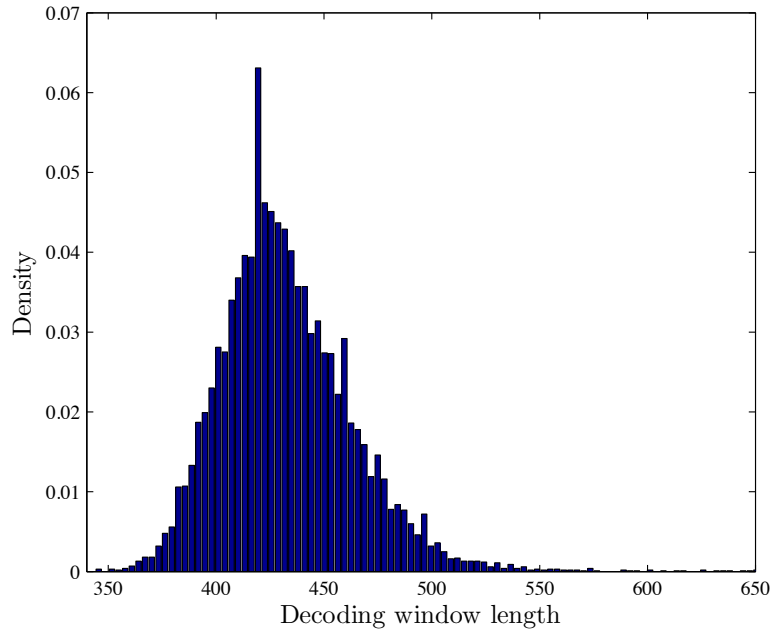


FIGURE 4.14: Probability density function of decoding window length using AWD-I at decoding time instant $t = 60$ with $\text{LLR}_{\text{th}} = 10$, $\frac{E_b}{N_0} = 3\text{dB}$ and $M = 30$.

(pdf) of the decoding window length at decoding time instant $t = 60$ is shown in Fig. 4.14, which is measured over 10,000 samples. From Fig. 4.14, we observe that the pdf is concentrated around the decoding window length of 440, while the probability of getting a decoding window length of over 550 is significantly small. Note that to obtain a fixed complexity decoder, one may fix the maximum decoding window length. Depending on the channel parameter, fixing the maximum decoding window size may affect the code performance. For example, with worse channel condition, a fixed maximum window size may drastically degrade the code performance. Limiting the maximum window length will eventually leave errors behind, and thus error floor may occur.

4.5 Conclusion

In this chapter, we have introduced an efficient linear anytime channel code based on spatially coupled codes. Through the asymptotic density evolution analysis, we observe the anytime properties of the proposed code over BEC and BIAWGN channels. A theoretical investigation of the delay-exponent of the proposed anytime code has been done for the BEC. Anytime decoders with reduced complexity are presented in this chapter. An algorithm is presented to estimate the finite-length performance of the proposed anytime codes. Through simulation results, we have shown that, our proposed code performs better than existing linear anytime codes. Moreover, one can optimize the anytime code by varying the design parameters.

Chapter 5

Anytime Codes for Relay Channels

In this chapter, we study the design and analytical analysis of anytime codes for a three-terminal relay network. With decode-and-forward relaying protocol, we propose bilayer anytime codes for the relay network. Considering two different transmission schemes in the relay-destination link, we investigate the anytime characteristics of the proposed code through density evolution analysis. For both transmission schemes, we analytically derive the delay-exponents using mathematical induction technique. We show that the analytical delay-exponent has a close match with the delay-exponent obtained from numerical results. Through simulation results, we also show the finite-length performance of the proposed bilayer anytime code.

Remainder of this chapter is organized as follows. In Section 5.1, we describe the system model of the anytime transmission over the three terminal relay network. The proposed bilayer anytime code is described in Section 5.2. Considering a protected transmission scheme in the relay-destination link, density evolution analysis, relay's delay effect and delay exponent analysis are shown in Section 5.3. With unprotected relaying in the relay-destination link, similar analysis is carried out in Section 5.4.

5.1 System Model

The source produces a K_s -bit message $\mathbf{m}_t \in \{0, 1\}^{K_s}$ at time $t = 1, 2, \dots$. Then the anytime channel encoder at the source produces a M_s -bit channel input $\mathbf{u}_t \in \{0, 1\}^{M_s}$ as a function of all available source messages $\mathbf{m}_1^t = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_t]$.

The channel encoder is defined by the function $\mathcal{E}_t^s(\mathbf{m}_1^t)$ and the code rate is $R_s = \frac{K_s}{M_s}$ (bits/channel use). The encoded message $\mathbf{u}_t$ is broadcasted to the relay and destination terminals. Let $\bar{\mathbf{v}}_t$ and $\hat{\mathbf{v}}_t$ be the channel outputs corresponding to $\mathbf{u}_t$ received by the relay and destination terminals, respectively.

In contrast to the traditional DF relaying scenario, it is not reasonable to assume that the relay recovers the received message instantly. The system design needs to take into account the trade-off between delay and reliability at the relay terminal. Let d_r denote the delay between decoding and encoding at the relay. At time $t = d_r + 1, d_r + 2, \dots$, the anytime channel decoder at the relay produces estimates $\bar{\mathbf{u}}_1^t(t)$ of $\mathbf{u}_1^t$ using the decoding function $\mathcal{D}_t^r(\bar{\mathbf{v}}_1^t) = \bar{\mathbf{u}}_1^t(t)$. Then the relay's anytime channel encoder produces a M_r -bit channel input $\mathbf{p}_t$ using the encoding function $\mathcal{E}_t^r(\bar{\mathbf{u}}_1^{t-d_r}(t))$ and transmits $\mathbf{p}_t$ to the destination. Let $\hat{\mathbf{p}}_t$ denote the message received by the destination. Thus, at time $t > d_r$, the destination computes an estimate of $\mathbf{m}_1^t$ using $\hat{\mathbf{m}}_1^t(t) = \mathcal{D}_t^d(\hat{\mathbf{v}}_1^t, \hat{\mathbf{p}}_{d_r+1}^t)$.

We now consider the estimate $\hat{\mathbf{m}}_i(t)$ of message $\mathbf{m}_i$ at time $t \geq i$ at the destination; the corresponding decoding delay for message $\mathbf{m}_i$ is $d = t - i$. The probability of error for the message i at time t will be $P_e(i, t) = \Pr(\hat{\mathbf{m}}_i(t) \neq \mathbf{m}_i)$. The requirements of the *anytime code* are satisfied if there exists a finite $\beta > 0$ such that [12]:

$$P_e(i, t) \leq \beta e^{-\alpha(t-i)}, \quad \forall (t-i) \geq 0. \quad (5.1)$$

5.2 Proposed Bilayer Anytime SC-LDPC Code

In this section, we propose a design of anytime codes for the relay channel. The source uses a $(d_{v_1}, d_{c_1}, \lambda_1)$ -anytime SC-LDPC code $\mathcal{C}_s$ to produce an encoded message $\mathbf{u}_t$ at time $t = 1, 2, \dots$. The source then broadcasts the encoded messages to the relay and destination terminals. Let M_s and $\frac{d_{v_1}}{d_{c_1}}M_s$ be the number of variable and check nodes, respectively at each position of code $\mathcal{C}_s$.

At time $t = d_r + 1, d_r + 2, \dots$, the relay produces $\bar{\mathbf{u}}_1^t(t)$, which is an estimate of $\mathbf{u}_1^t$ and then generates $\mathbf{p}_t$ in the following manner. Along with code $\mathcal{C}_s$, we first place M_r additional check nodes, known as relay check nodes, at each position of the Tanner graph. Then we add d_{v_2} additional edges to each variable node of code $\mathcal{C}_s$. Each of the additional d_{v_2} edges of a variable node at position i is connected to the relay check nodes in the range $[i + d_r, i + d_r + 1, \dots, \infty]$, where the connections follow an exponential distribution with parameter λ_2 . Thus, the relay check nodes

have degree $d_{c_2} = d_{v_2} \frac{M_s}{M_r}$. Then we include M_r new variable nodes at each position of the Tanner graph, where a new variable node at position i is connected to a relay check node at position i through a deterministic edge. The value of each new variable node is chosen to satisfy the corresponding relay check equation. We refer to the new variable nodes as extra parity bits. The extra parity bits at position i constitute $\mathbf{p}_i$. We refer to the overall code as a $(d_{v_1}, d_{c_1}, \lambda_1, d_{v_2}, d_{c_2}, \lambda_2)$ -bilayer anytime SC-LDPC code. Such a bilayer anytime code is depicted in Fig. 5.1.

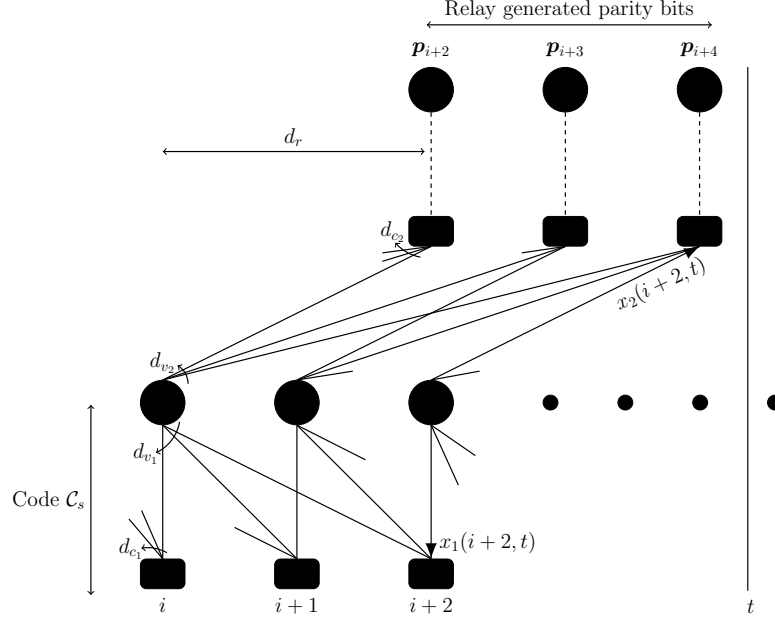


FIGURE 5.1: Proposed $(d_{v_1}, d_{c_1}, \lambda_1, d_{v_2}, d_{c_2}, \lambda_2)$ -bilayer anytime SC-LDPC code. Degrees of the nodes are depicted on the left side, while the update messages used in the density evolution analysis are depicted on the right side.

With the above bilayer anytime code, we present two relaying schemes for the rd link. (i) We assume that the relay encodes the extra parity bits using a capacity approaching code of rate $R_{rd} = 1 - \epsilon_{rd}$ and sends the encoded bits to the destination. This encoding step ensures reliable transmission of extra parity bits to the destination when M_s approaches infinity. The destination first decodes the extra parity bits and then decodes the source's message with the help of the extra parity bits. We refer to this scheme as the coded relaying scheme. (ii) In the second case, the parity bits are directly used as input to the rd link. One can consider scheme (ii) as a special case of scheme (i), where the extra parity bits are transmitted over the rd link with a code of rate 1. Scheme (ii) can generate $\frac{1}{R_{rd}}$ times as many extra parity bits compared to scheme (i). We refer to this scheme as the uncoded relaying scheme. The asymptotic analyses and results for scheme (i) and scheme (ii) are presented in the following sections.

5.3 Coded Relaying Scheme

In this section, we present the asymptotic analysis and numerical results of the proposed bilayer anytime code, where the extra parity bits generated by the relay are protected by an independent capacity approaching code. We present the anytime characteristics of this scheme through the density evolution analysis and observe the effect of the relay's delay on the overall performance. We also derive the delay exponent of this scheme through mathematical induction and compare the theoretical delay exponent with the numerical delay exponent.

5.3.1 Density Evolution

As discussed earlier, the relay cannot recover the received message instantly due to the anytime characteristics of code $\mathcal{C}_s$. At the relay, at time $t = d_r + 1, d_r + 2, \dots$, the erasure probability of a message i is $P_e^{sr}(i, t) = \Pr(\mathbf{u}_i \neq \bar{\mathbf{u}}_i(t))$, which can be calculated by finding $P_e(i, t)$ from (4.4). Since the relay generates extra parity bits $\mathbf{p}_t$ based on the messages $\bar{\mathbf{u}}_1^{t-d_r}(t)$, $\mathbf{p}_t$ may contain erased bits. The erasure probability of the generated extra parity bits depends on the value of d_r and decays exponentially with increasing d_r . Let $\xi_r(t)$ denote the erasure probability of the elements of $\mathbf{p}_t$, then

$$\xi_r(t) = 1 - \left(1 - \sum_{j=0}^{\infty} P_{r_2}(j) P_e^{sr}(t - j - d_r, t) \right)^{d_{c_2}}, \quad (5.2)$$

where we consider $P_e^{sr}(i, t) = 0$ if $i \leq 0$. Let us define the following notation for the density evolution analysis:

- $x_1^{(\ell)}(i, t)$ be the erasure probability of a message outgoing from a VN at position i to the CNs of code $\mathcal{C}_{sr}$ in the ℓ^{th} iteration, at time t .
- $x_2^{(\ell)}(i, t)$ is the erasure probability of a message outgoing from a VN at position i to the relay check nodes in the ℓ^{th} iteration, at time t
- $P_e(i, t)$ is the erasure probability of message i at decoding time instant t .

Following the derivation of the density evolution for bilayer LDPC convolutional codes [65], the density evolution for the proposed code is presented in (5.3), where $f_1(i, t)$ represents

- $x_1^{(\ell)}(i, t)$ when $q_1 = d_{v_1} - 1$ and $q_2 = d_{v_2}$,
- $x_2^{(\ell)}(i, t)$ when $q_1 = d_{v_1}$ and $q_2 = d_{v_2} - 1$,
- $P_e(i, t)$ when $q_1 = d_{v_1}$, $q_2 = d_{v_2}$ and $\ell = \infty$.

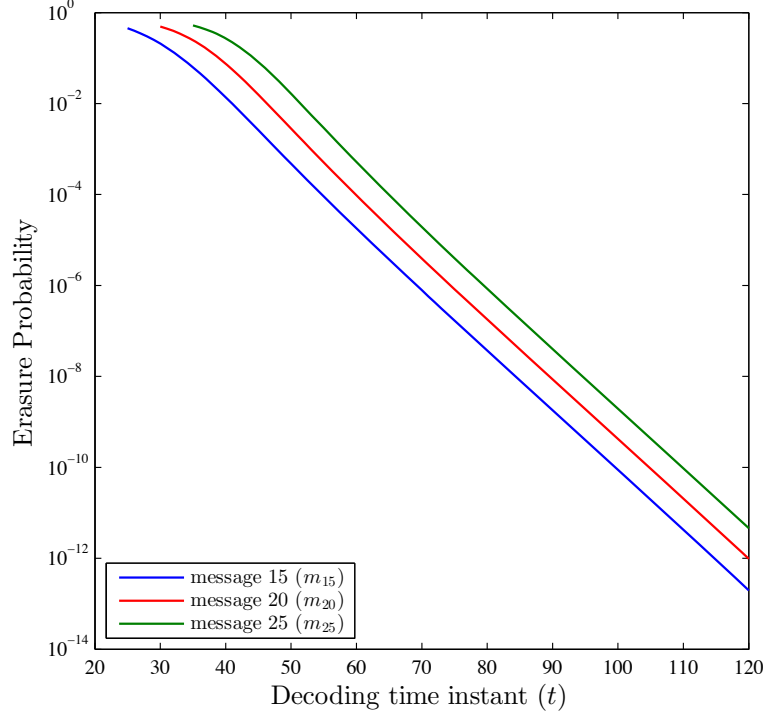


FIGURE 5.2: Asymptotic performances of three messages ($\mathbf{m}_{15}, \mathbf{m}_{20}, \mathbf{m}_{25}$) versus decoding time instant t with $d_r = 10$.

$$\begin{aligned}
 f_1(i, t) = & \epsilon_{sd} \left(1 - \sum_{j=0}^{t-i} P_{r_1}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_1}(k) x_1^{(\ell-1)}(i+j-k, t) \right)^{d_{c_1}-1} \right)^{q_1} \\
 & \cdot \left(1 - (1 - \xi_r(i + d_r)) \sum_{j=0}^{t-i-d_r} P_{r_2}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_2}(k) x_2^{(\ell-1)}(i+j-k, t) \right)^{d_{c_2}-1} \right)^{q_2}.
 \end{aligned} \tag{5.3}$$

Recall from the code definition that $P_{r_1}(k) = e^{-k\lambda_1}(1 - e^{-\lambda_1})$ and $P_{r_2}(k) = e^{-k\lambda_2}(1 - e^{-\lambda_2})$, respectively. In the above density evolution, we consider $x_1^{(\ell)}(i, t) = x_2^{(\ell)}(i, t) = 0, \forall \ell$ when $i \leq 0$. Finally, we initialize the density evolution with $x_1^{(0)}(i, t) = x_2^{(0)}(i, t) = \epsilon_{sd}$ when $0 < i \leq t$. We now present the results from the above density evolution analysis. We consider a $(d_{v_1}, d_{c_1}, \lambda_1, d_{v_2}, d_{c_2}, \lambda_2) =$

(3, 6, 0.1, 2, 6, 0.1)-bilayer anytime code with $\epsilon_{sd} = 0.7$ and $\epsilon_{sr} = 0.4$. Fig. 5.2 shows the erasure probabilities of three messages at the destination versus the decoding time instant. For each of the messages, we observe an exponential decay of the erasure probability with increasing delay and the erasure performance of a given message is a shifted version of another. This shows that the erasure performance of a message is only dependent on the decoding delay and is independent of its absolute position. Fig. 5.3 shows how the exponential decay of the erasure probability is affected by the relay's delay d_r . We observe erasure floors at small relay's delays ($d_r = 6$), since the erasure probability of the relayed parity bits is large at small delays and hence, little additional information is provided by the relay. We notice that there exists a minimum value of d_r for which the decoded erasure probability decays exponentially. For example, with the above code and channel conditions, the decoded erasure probability decays exponentially when $d_r > 7$. We also notice that the erasure probability decreases faster with increasing d_r , since the erasure probability of the relayed parity bits approaches zero as d_r increases. However, a choice of large d_r prolongs the waiting period to start decoding at the destination. Thus, for a given bilayer anytime code and channel condition, an appropriate choice of d_r is necessary. In Fig. 5.4, we show how the erasure probability at the destination varies with the relay's delay d_r , while the erasure probability is measured at a fixed decoding delay d . We notice that for a given d , there exists an optimum d_r for which the erasure probability is lowest. We also notice that the optimum d_r increases slowly with increasing d .

5.3.2 Delay-Exponent Analysis

We now analytically find the upper bound of the delay exponent of the proposed bilayer anytime code. We also compare the analytical delay-exponent with *numerical* delay-exponent, where we determine the numerical delay-exponent from the slope of the asymptotic performance obtained from density evolution results (for example the result shown in Fig. 5.2). For the delay exponent analysis, we neglect the erasure probability caused at the sr link due to the finite relay's delay¹. Note that due to the erasure-free assumptions, the following theorem gives an upper bound on the delay-exponent achieved by the proposed bilayer anytime code.

Theorem 5.1. *Consider the $(d_{v_1}, d_{c_1}, \lambda_1, d_{v_2}, d_{c_2}, \lambda_2)$ -bilayer anytime SC-LDPC code as described in Section 5.2. By assuming an erasure-free sr link with protected*

¹This erasure-free assumption becomes valid for sufficiently large d_r , where the erasure probability $P_e^{sr}(i, t)$ approaches zero.

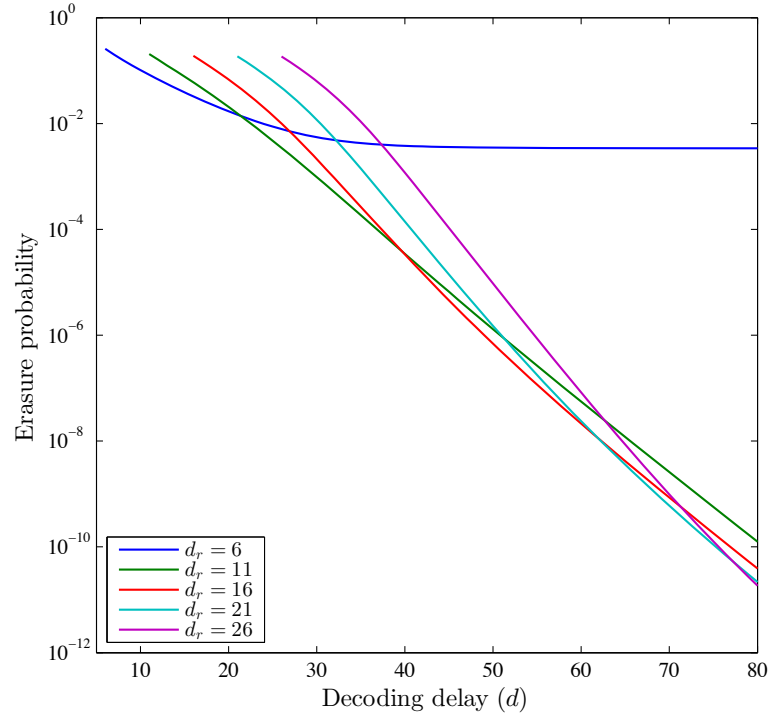


FIGURE 5.3: Effect of relay's delay d_r on the decaying behaviour of erasure probability for coded relaying scheme.

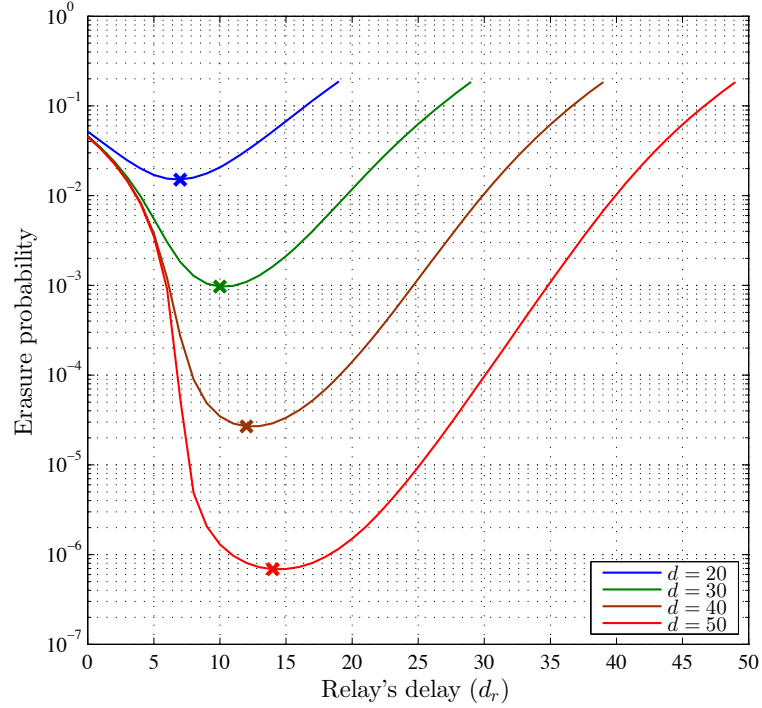


FIGURE 5.4: Erasure probability of a message at a fixed decoding delay d , while varying relay's delay d_r for coded relaying scheme. We observe the performance for $d = \{20, 30, 40, 50\}$. Optimum d_r for each d is indicated by ' $\times$ '.

transmission in the rd link, an upper bound to the delay-exponent of the proposed bilayer code is given by

$$\alpha = \lambda_1 d_{v_1} + \lambda_2 d_{v_2}, \quad (5.4)$$

when $\epsilon_{sd} \leq \hat{\epsilon}_b$, where

$$\hat{\epsilon}_b \triangleq \max_{\beta > \epsilon_{sd}} \min_{d \geq 0} \beta (e^{-\lambda_1} + \mathcal{G}_1)^{-d_{v_1}} (e^{-\lambda_2} + \mathcal{G}_2)^{-d_{v_2}}.$$

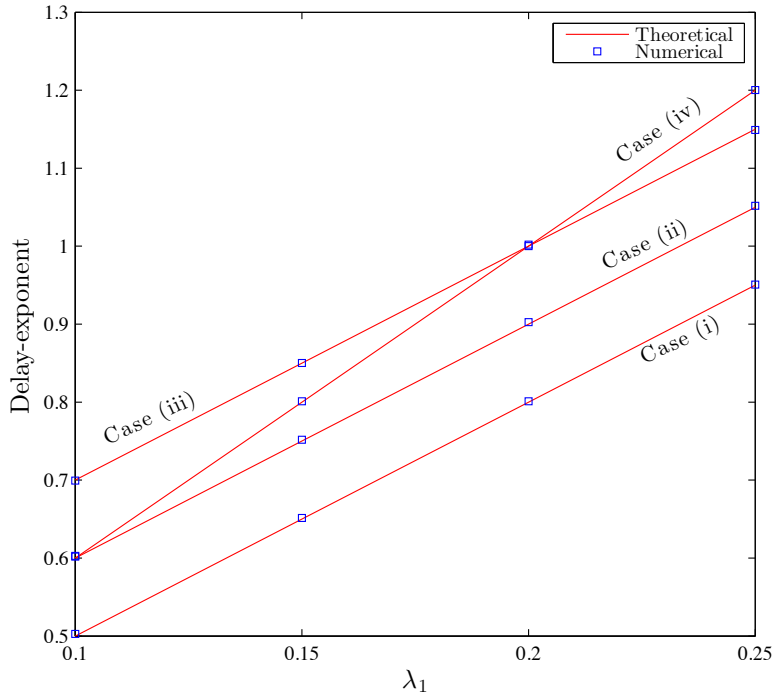


FIGURE 5.5: Comparison between theoretical and numerical delay-exponents for four cases: (i) $d_{v_1} = 3, d_{v_2} = 2, \lambda_2 = 0.1$ (ii) $d_{v_1} = 3, d_{v_2} = 3, \lambda_2 = 0.1$ (iii) $d_{v_1} = 3, d_{v_2} = 2, \lambda_2 = 0.2$ (iv) $d_{v_1} = 4, d_{v_2} = 2, \lambda_2 = 0.1$. For all cases, we consider $d_{c_1} = 6, d_{c_2} = 8$. For all cases, we set $d_r = 30$.

The definitions of $(\mathcal{G}_1, \mathcal{G}_2)$ and the proof of the theorem are given in Appendix A.2. Similar to the delay-exponent analysis of the P2P communication scenario, it can be shown numerically that the proposed anytime code achieves the delay-exponent $\alpha = \lambda_1 d_{v_1} + \lambda_2 d_{v_2}$ whenever ϵ_{sd} is below a threshold $\hat{\epsilon}_{sd}$. We can consider $\hat{\epsilon}_b$ as a bound on the threshold $\hat{\epsilon}_{sd}$. It is worth mentioning that there exists a large gap between $\hat{\epsilon}_{sd}$ and $\hat{\epsilon}_b$ due to the limited tightness of the bounds used in the proof. For example, with a $(3, 6, 0.1, 2, 8, 0.1)$ -bilayer anytime SC-LDPC code, both the theoretical and numerical delay-exponent is $\alpha = 0.5$ while $\hat{\epsilon}_{sd}$ and $\hat{\epsilon}_b$ are 0.732 and 0.115, respectively. In Fig. 5.5, we compare the above theoretical delay-exponent

with the numerical delay-exponent obtained from the density evolution results. From Fig. 5.5, it is clear that the theoretical delay-exponent closely matches the numerical delay-exponent. In (5.4), we observe that asymptotically the delay-exponent increases linearly with the increment of any parameter (d_{v_1} , d_{v_2} , λ_1 and λ_2). However, in practice (i.e., for finite-length case), increasing d_{v_1} or d_{v_2} may introduce short cycles in the code. On the other hand, increasing λ_1 or λ_2 leads to fewer connections to distant protographs. Thus, increasing d_{v_1} , d_{v_2} , λ_1 , or λ_2 may result in an erasure floor at large delays in finite-length codes.

5.4 Uncoded Relaying Scheme

For the coded relaying scheme, we assumed that the relay protects extra parity bits using a capacity approaching block code, which ensures reliable transmission over the rd link. However, this assumption is not realistic in delay sensitive applications, since reliable transmission using the block codes is only feasible for very large message length. In this section, the extra parity bits are directly transmitted over the rd link². For this uncoded scheme, the relay generates more parity bits compared to the coded scheme. However, these parity bits may be erased at the destination due to the rd link. Compared to the coded relaying scheme, the uncoded relaying scheme reduces the computational complexity at the relay by performing uncoded transmission to the destination. In the following, we present the theoretical analysis and numerical results for the uncoded relaying scheme.

5.4.1 Density Evolution

In contrast to the asymptotic analysis in Section 5.3, we take the channel erasure probability of the rd link into account in this asymptotic analysis. At time $t = d_r + 1, d_r + 2, \dots$, let $\xi_{srd}(t)$ denote the erasure probability of the extra parity bits in $\mathbf{p}_t$ at the destination due to the sr link and the rd link. Since the relay transmits extra parity bits without any protection to the destination, $\xi_{srd}(t)$ is calculated as $\xi_{srd}(t) = \xi_r(t) + \epsilon_{rd} - \xi_r(t)\epsilon_{rd}$. With the uncoded relaying scheme, the density evolution equation is presented in (5.5), where $f_2(i, t)$ represents

- $x_1^{(\ell)}(i, t)$ when $q_1 = d_{v_1} - 1$ and $q_2 = d_{v_2}$,

²The extra parity bits, generated by the relay, may contain erasure events due to the sr link. For sake of simplicity, we assume that the destination receives an erasure bit if the relay transmits an erased extra parity bit.

- $x_2^{(\ell)}(i, t)$ when $q_1 = d_{v_1}$ and $q_2 = d_{v_2} - 1$,
- $P_e(i, t)$ when $q_1 = d_{v_1}$, $q_2 = d_{v_2}$ and $\ell = \infty$.

$$f_2(i, t) = \epsilon_{sd} \left(1 - \sum_{j=0}^{t-i} P_{r_1}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_1}(k) x_1^{(\ell-1)}(i+j-k, t) \right)^{d_{c_1}-1} \right)^{q_1} \cdot \left(1 - (1 - \xi_{srd}(i + d_r)) \sum_{j=0}^{t-i-d_r} P_{r_2}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_2}(k) x_2^{(\ell-1)}(i+j-k, t) \right)^{d_{c_2}-1} \right)^{q_2}. \quad (5.5)$$

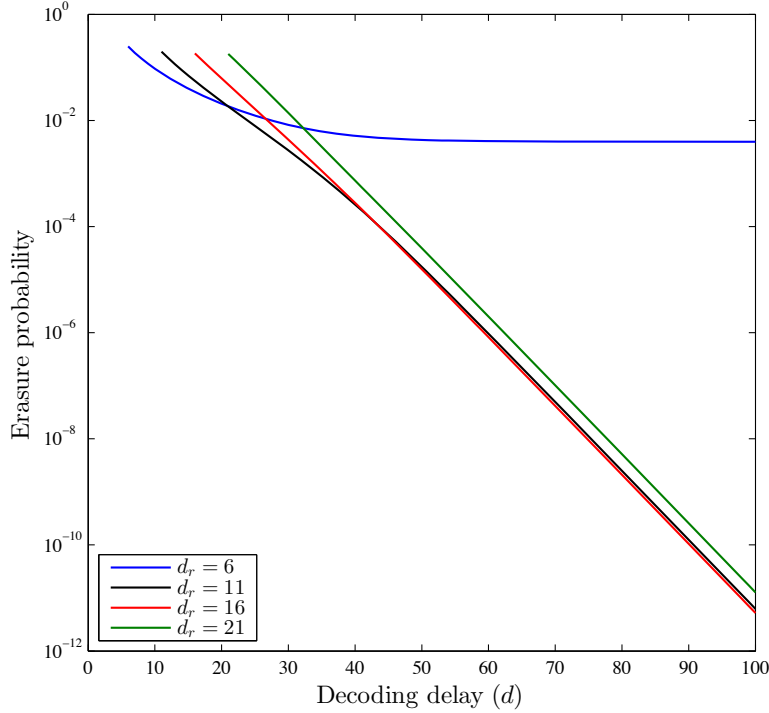


FIGURE 5.6: Asymptotic performance of the proposed bilayer anytime code while considering the uncoded relaying scheme. We consider a $(d_{v_1}, d_{c_1}, \lambda_1, d_{v_2}, d_{c_2}, \lambda_2) = (3, 6, 0.1, 2, 3, 0.2)$ -bilayer anytime code with $\epsilon_{sd} = 0.7$, $\epsilon_{sr} = 0.4$ and $\epsilon_{rd} = 0.4$.

In the above density evolution, we consider $x_1^{(\ell)}(i, t) = x_2^{(\ell)}(i, t) = 0, \forall \ell$ when $i \leq 0$. Finally, we initialize the density evolution with $x_1^{(0)}(i, t) = x_2^{(0)}(i, t) = \epsilon_{sd}$ for $0 < i \leq t$. Fig. 5.6 presents the results from the above density evolution analysis. We consider a $(d_{v_1}, d_{c_1}, \lambda_1, d_{v_2}, d_{c_2}, \lambda_2) = (3, 6, 0.1, 2, 3, 0.2)$ -bilayer anytime code with $\epsilon_{sd} = 0.7$, $\epsilon_{sr} = 0.4$ and $\epsilon_{rd} = 0.4$. Similar to the coded scheme, we observed the exponential decay of erasure probability at sufficiently large relay's delay. We

also observed that the performance of a given message is the shifted version of another. For different relay's delay d_r , Fig. 5.6 shows the erasure probability of a message at the destination while varying the decoding delay. Erasure floor appears for small delays ($d_r = 5$), since the erasure probability of the extra parity bits is large at small delays. However, for the uncoded scheme, when the relay's delay exceeds a certain threshold, the decoded erasure probability decays at a fixed exponential rate. This characteristic is different from the coded scheme where the rate of decay increases with the relay's delay, since at the destination, the erasure probability of extra parity bits saturates to ϵ_{rd} .

5.4.2 Delay-Exponent Analysis

The following theorem gives an analytical formula for the delay exponent of the above described uncoded relaying scheme.

Theorem 5.2. *Consider the $(d_{v_1}, d_{c_1}, \lambda_1, d_{v_2}, d_{c_2}, \lambda_2)$ -bilayer anytime SC-LDPC code with uncoded transmission in the rd link. The delay-exponent of the proposed bilayer code is given by:*

$$\alpha = \lambda_1 d_{v_1}, \quad (5.6)$$

when $\epsilon_{sd} \leq \tilde{\epsilon}_b$, where

$$\tilde{\epsilon}_b \triangleq \max_{\beta > \epsilon_{sd}} \min_{d \geq 0} \beta (e^{-\lambda_1} + \mathcal{G}_1)^{-d_{v_1}} (\xi_{srd} (1 - e^{-\lambda_2(d+2-d_r)}) + \mathcal{G}_3)^{-d_{v_2}}.$$

The proof of the theorem and the definition of $\mathcal{G}_3$ are given in Appendix A.3. Similar to the coded relaying scheme, numerically we observe that the delay-exponent $\alpha = \lambda_1 d_{v_1}$ whenever ϵ_{sd} is below a threshold $\tilde{\epsilon}_{sd}$. We can consider $\tilde{\epsilon}_b$ as a bound on the threshold $\tilde{\epsilon}_{sd}$. Again, we observe a large gap between $\tilde{\epsilon}_{sd}$ and $\tilde{\epsilon}_b$ due to the limited tightness of the bounds used in the proof. From the above theoretical exponent, we observe that as long as the threshold condition is satisfied, the delay exponent α only depends on the code $\mathcal{C}_{sr}$ and does not depend on other parameters (i.e., λ_2 or d_{v_2}). The close match between this theoretical delay exponent and the numerical delay exponent³ is observed in Fig. 5.7. The

³The numerical delay-exponent can be determined from the erasure probability performance of density evolution results (for example results shown in Fig. 5.6).

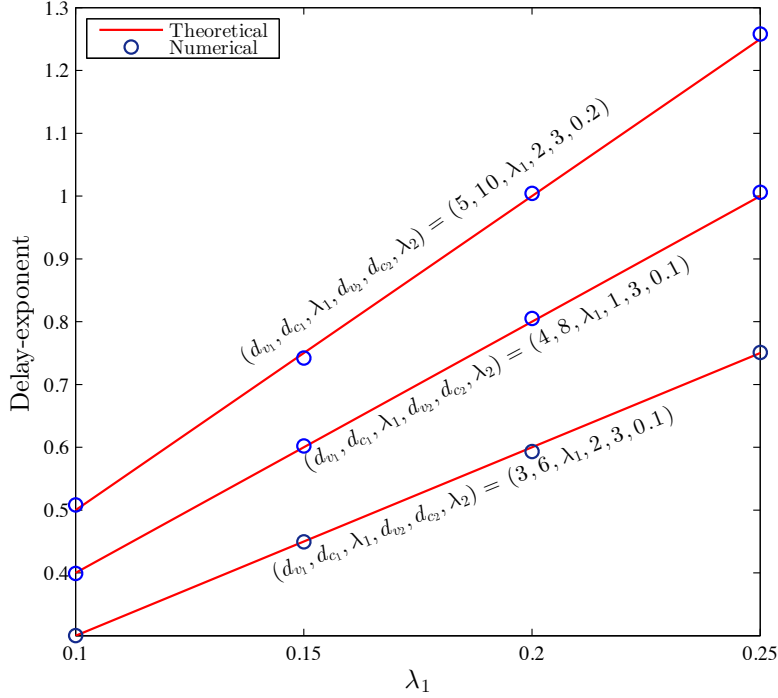


FIGURE 5.7: Comparison between theoretical and numerical delay-exponents of uncoded relaying scheme.

results shown in Fig. 5.7 also verify that the delay-exponent of the uncoded relaying scheme only depends on λ_1 and d_{v_1} and is independent of all other code design parameters. We notice that the delay-exponent achieved from the uncoded scheme is similar to the P2P (d_v, d_c, λ) -anytime SC-LDPC code. However, a better threshold compared to the delay exponent of the P2P anytime code can be obtained. For example, both P2P $(d_v, d_c, \lambda) = (3, 6, 0.1)$ -anytime SC-LDPC code and $(d_{v_1}, d_{c_1}, \lambda_1, d_{v_2}, d_{c_2}, \lambda_2) = (3, 6, 0.1, 2, 3, 0.2)$ -bilayer anytime code have a delay-exponent $\alpha = 0.3$, while we observe a threshold of 0.487 for the P2P anytime code and a threshold of 0.721 for the bilayer anytime code. On the other hand, the uncoded relaying scheme obtains a lower delay-exponent compared to the coded relaying scheme. This is because the erasure probability of the extra parity bits is large due the rd link and this erasure probability does not decay at the destination. In the following, we propose a modified design to improve the performance of the uncoded relaying scheme.

5.4.3 Modified design

In Fig. 5.8, we show a modification in the Tanner graph to improve the performance of the uncoded scheme. Along with the deterministic edge between the relay check

[illegible]

Let $\xi_r(i, t)$ be the erasure probability of $\mathbf{p}_i(t)$, which can be calculated by,

where we consider $P_e^{sr}(i, t) = 0$ if $i \leq 0$ and $\xi_r(i, t) = 0$ if $i < d_r$. In (5.7), $P_{r_3}(k) = e^{-k\lambda_3}(1 - e^{-\lambda_3})$ is the edge distribution probability used in the extra parity generation. Due to the relay's delay and the rd link, at time $t = d_r + 1, d_r + 2, \dots$,

the erasure probability of extra parity bits from the relay to the destination is given by $\xi_{srd}(t) = \xi_r(t, t) + \epsilon_{rd} - \xi_r(t, t)\epsilon_{rd}$.⁴

For asymptotic analysis of this modified design, we perform the following four edge-type density evolution analysis. Along with $x_1^{(\ell)}(i, t)$ and $x_2^{(\ell)}(i, t)$, we also introduce two following notation:

- $x_3^{(\ell)}(i, t)$ be the erasure probability of a message outgoing from i^{th} extra parity bit along the deterministic edge in the ℓ^{th} iteration, at time t .
- $x_4^{(\ell)}(i, t)$ be the erasure probability of a message outgoing from i^{th} extra parity bit along the probabilistic edges in the ℓ^{th} iteration, at time t .

The density evolution of the modified design can be obtained by tracking the message passing on four types of edges at each iteration, where the outgoing erasure probability on one edge type in the current iteration depends on the incoming erasure probabilities in the previous iteration. The density evolution update equations are given in (5.8), (5.9) and (5.10). In (5.8), $f_3(i, t)$ represents

- $x_1^{(\ell)}(i, t)$ when $q_1 = d_{v_1} - 1$ and $q_2 = d_{v_2}$,
- $x_2^{(\ell)}(i, t)$ when $q_1 = d_{v_1}$ and $q_2 = d_{v_2} - 1$,
- $P_e(i, t)$ when $q_1 = d_{v_1}$, $q_2 = d_{v_2}$ and $\ell = \infty$.

$$f_3(i, t) = \epsilon_{sd} \left(1 - \sum_{j=0}^{t-i} P_{r_1}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_1}(k) x_1^{(\ell-1)}(i+j-k, t) \right)^{d_{c_1}-1} \right)^{q_1} \left(1 - \sum_{j=0}^{t-i-d_r} P_{r_2}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_2}(k) x_2^{(\ell-1)}(i+j-k, t) \right)^{d_{c_2}-1} \left(1 - x_3^{(\ell-1)}(i+j+d_r, t) \right) \cdot \left(1 - \sum_{k=1}^{\infty} P_{r_3}(k-1) x_4^{(\ell-1)}(i+j-k+d_r, t) \right)^{d_{c_3}} \right)^{q_2}, \quad (5.8)$$

$$x_3^{(\ell)}(i, t) = \xi_{srd}(i) \left(1 - \sum_{j=1}^{t-i-d_r} P_{r_3}(j-1) \left(1 - \sum_{k=1}^{\infty} P_{r_3}(k-1) x_4^{(\ell-1)}(i+j-k, t) \right)^{d_{c_3}-1} \right)$$

⁴We observed that typically $\epsilon_{rd} \gg \xi_r(t, t)$, which results $\xi_{srd}(t) \approx \epsilon_{rd}$.

$$\cdot \left(1 - x_3^{(\ell-1)}(i+j, t)\right) \left(1 - \sum_{k=0}^{\infty} P_{r_2}(k) x_2^{(\ell-1)}(i+j-k-d_r, t)\right)^{d_{c_2}} \Big)^{d_{v_3}}, \quad (5.9)$$

$$\begin{aligned} x_4^{(\ell)}(i, t) &= \xi_{srd}(i) \left(1 - \sum_{j=1}^{t-i-d_r} P_{r_3}(j-1) \left(1 - \sum_{k=1}^{\infty} P_{r_3}(k-1) x_4^{(\ell-1)}(i+j-k, t)\right)^{d_{c_3}-1} \right. \\ &\cdot \left. \left(1 - x_3^{(\ell-1)}(i+j, t)\right) \left(1 - \sum_{k=0}^{\infty} P_{r_2}(k) x_2^{(\ell-1)}(i+j-k-d_r, t)\right)^{d_{c_2}} \right)^{d_{v_3}-1} \left(1 - \right. \\ &\left. \left(1 - \sum_{k=1}^{\infty} P_{r_3}(k-1) x_4^{(\ell-1)}(i-k, t)\right)^{d_{c_3}} \left(1 - \sum_{k=0}^{\infty} P_{r_2}(k) x_2^{(\ell-1)}(i-k-d_r, t)\right)^{d_{c_2}} \right). \end{aligned} \quad (5.10)$$

In the above density evolution, we consider $x_1^{(\ell)}(i, t) = x_2^{(\ell)}(i, t) = 0, \forall \ell$ when $i \leq 0$ and $x_3^{(\ell)}(i, t) = x_4^{(\ell)}(i, t) = 0, \forall \ell$ when $i \leq d_r$. Finally, we initialize the density evolution with $x_1^{(0)}(i, t) = x_2^{(0)}(i, t) = \epsilon_{sd}$ when $0 < i \leq t$ and $x_3^{(0)}(i, t) = x_4^{(0)}(i, t) = \xi_{srd}(i)$ when $d_r < i \leq t$. We now present the results obtained from the above density evolution analysis. In Fig. 5.9, we present the performance comparison between the coded and uncoded relaying schemes with sufficiently large relay's delay ($d_r = 50$). For the coded relaying scheme, we consider a $(d_{v_1}, d_{c_1}, \lambda_1, d_{v_2}, d_{c_2}, \lambda_2) = (3, 6, 0.1, 2, 6, 0.1)$ -bilayer anytime code with the assumption that a rate half code is used in the rd link which can protect the extra parity bits even for $\epsilon_{rd} = 0.5$. For a fair comparison, we consider a $(d_{v_1}, d_{c_1}, \lambda_1, d_{v_2}, d_{c_2}, \lambda_2) = (3, 6, 0.1, 2, 3, 0.1)$ -bilayer anytime code for the uncoded relaying schemes with $\epsilon_{rd} = 0.5$. With these code parameters, both coded and uncoded relaying schemes transmit same number of bits from the relay, and on average the destination receives the same number of non-erased extra parity bits. From Fig. 5.9, we observe that the coded relaying scheme outperforms the uncoded relaying schemes provided that a reliable transmission in the rd link. On the other hand, the modified uncoded scheme performs better than the earlier discussed uncoded scheme. The rationale behind this improvement is that the erasure probability of extra parity bits decays at the destination due to the modification in the design. Numerically we observe that the modified scheme results in an improved delay exponent $\alpha = \lambda_1 d_{v_1} + \lambda_2 d_{v_2}$, which is similar to the delay exponent of the coded relaying scheme. More importantly, unlike the coded relaying scheme, we do not make assumptions (like reliable transmission over the rd link) for the modified uncoded scheme. However, it is important to mention that due to the

four edge-type density evolution analysis of the modified scheme, we were unable to prove the delay-exponent analytically.

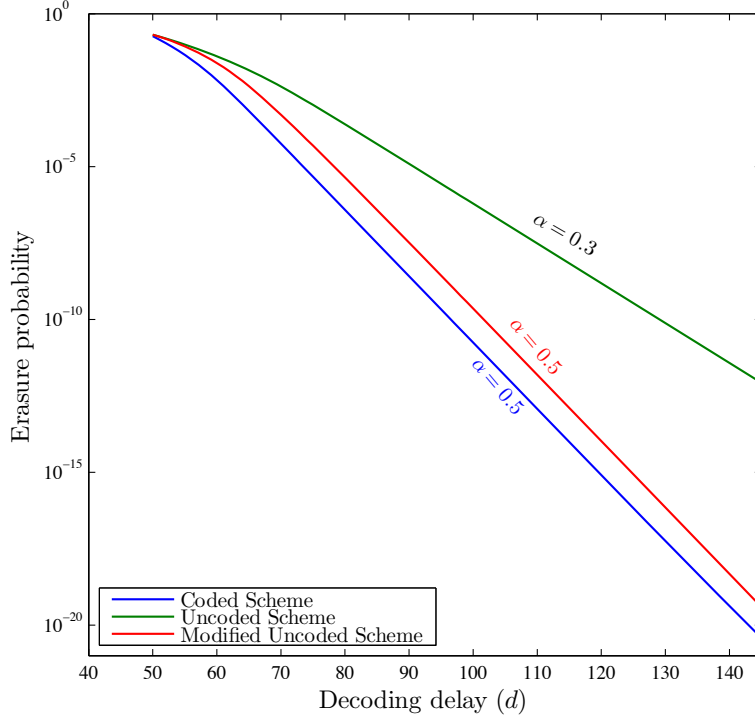


FIGURE 5.9: Asymptotic performance comparison between the proposed coded and uncoded relaying schemes. For the simulation, we set $\epsilon_{sd} = 0.7$, $\epsilon_{sr} = 0.4$ and $\epsilon_{rd} = 0.5$. For the modified scheme we choose $d_{v_3} = 3$.

For the above mentioned code parameters, we also present the finite-length performance comparison between the coded and uncoded relaying schemes. In Fig. 5.10, we show the erasure probability of message⁵ $\mathbf{m}_1$ with $M_s = 30$ and $d_r = 21$. Similar to the density evolution results shown in Fig. 5.9, we observe that the coded scheme results in better performance than the uncoded relaying schemes. However, for such a short message length, the coded relaying is not practical to implement, since it assumes reliable transmission over the rd link. On the other hand, the modified scheme leads to a better erasure probability decaying rate than the uncoded scheme. We observe that both coded and modified schemes exhibit high erasure floors at large decoding delay. This is expected since the both schemes introduce more cycles compare to the uncoded scheme.

⁵Note that, although we have presented the results for message $\mathbf{m}_1$ only, we observed similar performance for other messages. We have observed that the performances of the first few messages are slightly better than the later ones due to the boundary effect.

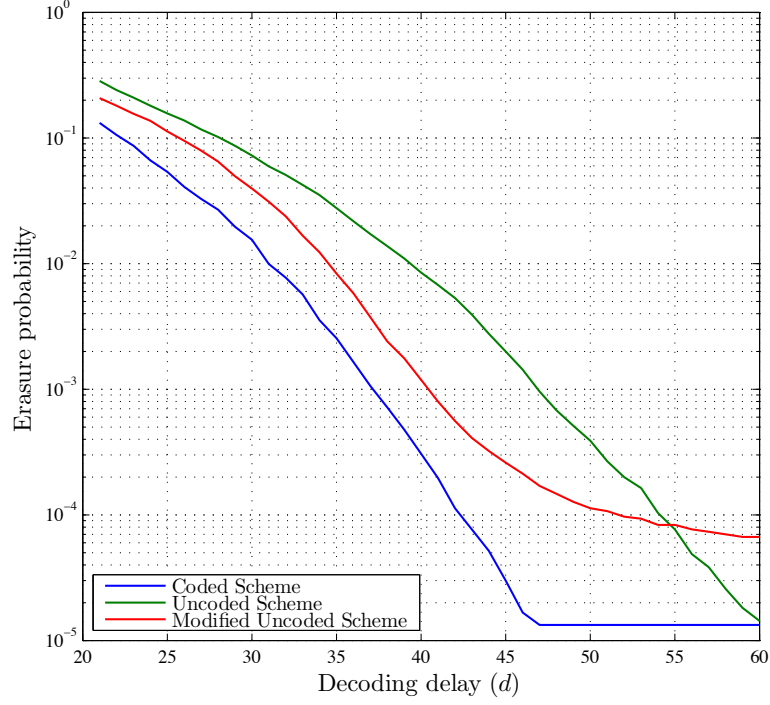


FIGURE 5.10: Finite-length ($M_s = 30$) performance comparison between the proposed coded and uncoded relaying schemes. In the simulation, we set the channel parameters as $\epsilon_{sd} = 0.65$, $\epsilon_{sr} = 0.4$ and $\epsilon_{rd} = 0.5$.

5.5 Conclusion

In this chapter, we have proposed a practical bilayer anytime coding scheme for the decode-and-forward relay channel. For two different relay-destination transmission schemes (namely coded and uncoded relaying schemes), we have investigated the anytime characteristics of the proposed code through density evolution analysis. We have shown the effect of the relay's delay on the overall performance of the proposed code. We have analytically found the delay-exponents for both schemes and observed a close match with the numerical delay-exponents. We have observed that the coded relaying scheme leads to the highest delay-exponent for a given bilayer anytime code while assuming reliable transmissions in the source-relay and relay-destination links. However, coded relaying scheme does not fit with practical implementation due to the assumptions of reliable transmissions. Although the uncoded relaying scheme is suitable for practical implementation, it exhibits a lower delay-exponent than the coded relaying scheme. To improve the delay-exponent of the uncoded relaying scheme, we have modified the code design which results in the highest delay-exponent without considering any unrealistic assumption.

Chapter 6

Anytime Codes for Multiple Access Channels

In this chapter, we propose anytime coding scheme for the 2-user multiple access channel. We first show the asymptotic analysis and analytical delay-exponent of the proposed code over the erasure multiple access channel. Then we extend the asymptotic analysis of the proposed code over Gaussian multiple access channel.

This chapter is organized as follows. In Section 6.1, we describe the system model when the transmission is taking place over multiple access channel. The proposed anytime coding scheme for multiple access channel is shown in Section 6.2. For erasure MAC, the density evolution analysis and delay-exponent analysis are described in Section 6.3. Asymptotic analysis and performance of the proposed code over Gaussian MAC are shown in Section 6.4. Before conclusion, we show the finite-length performance of the proposed code in Section 6.5.

6.1 System Model

The anytime transmission scheme considered for 2-user MAC is given as follows. At time t , let $\mathbf{m}_{A_t} \in \{0, 1\}^{K_A}$ and $\mathbf{m}_{B_t} \in \{0, 1\}^{K_B}$ be the information blocks produced by user- A and user- B , respectively. Then user- A generates an encoded message $\mathbf{u}_{A_t} \in \{0, 1\}^{M_A}$ in an anytime manner such that $\mathbf{u}_{A_t} = \mathcal{E}_t(\mathbf{m}_{A_1}^t)$, where $\mathbf{m}_{A_1}^t = [\mathbf{m}_{A_1}, \mathbf{m}_{A_2}, \dots, \mathbf{m}_{A_t}]$. Similarly, user- B generates an encoded message $\mathbf{u}_{B_t} \in \{0, 1\}^{M_B}$ in anytime manner such that $\mathbf{u}_{B_t} = \mathcal{E}_t(\mathbf{m}_{B_1}^t)$, where $\mathbf{m}_{B_1}^t =$

$[\mathbf{m}_{B_1}, \mathbf{m}_{B_2}, \dots, \mathbf{m}_{B_t}]$. Thus, the rate of the channel encoders of user- A and user- B become $R_A = \frac{K_A}{M_A}$ and $R_B = \frac{K_B}{M_B}$, respectively. Both users simultaneously transmit $\mathbf{u}_{A_t}$ and $\mathbf{u}_{B_t}$ over a MAC. Let the receiver observes $\mathbf{v}_t$ at time t due to the MAC. Then the receiver attempts to estimate all the transmitted messages of both users using the decoding function $(\hat{\mathbf{m}}_{A_1}^t(t), \hat{\mathbf{m}}_{B_1}^t(t)) = \mathcal{D}_t(\mathbf{v}_1^t)$, where $\hat{\mathbf{m}}_{A_1}^t(t)$ and $\hat{\mathbf{m}}_{B_1}^t(t)$ are the estimation of $\mathbf{m}_{A_1}^t$ and $\mathbf{m}_{B_1}^t$, respectively.

Let $P_{e_j}(i, t) = \Pr(\hat{\mathbf{m}}_{j_i}(t) \neq \mathbf{m}_{j_i})$ be the error probability of the message $\mathbf{m}_{j_i}$ at time t for user- j ($j \in \{A, B\}$). For the anytime transmission, a code must satisfy

$$P_{e_j}(i, t) \leq \beta e^{-\alpha_j(t-i)}, \quad \forall (t-i) \geq 0, \quad (6.1)$$

where $\beta > 0$ is a constant and $\alpha_j > 0$ is the delay-exponent of user- j .

6.2 Proposed Anytime Code Over MAC

Based on the P2P-anytime code shown in Chapter 4, we design the following anytime code for 2-user MAC. User- A takes a $(d_{v_A}, d_{c_A}, \lambda_A)$ -anytime SC-LDPC code, denoted as code $\mathcal{C}_A$, to generate an encoded message $\mathbf{u}_{A_t}$. Similarly, user- B takes another $(d_{v_B}, d_{c_B}, \lambda_B)$ -anytime SC-LDPC code denoted as $\mathcal{C}_B$ to generate an encoded message $\mathbf{u}_{B_t}$. We refer to the combination of code $\mathcal{C}_A$ and code $\mathcal{C}_B$ as $(d_{v_A}, d_{v_B}, d_{c_A}, d_{c_B}, \lambda_A, \lambda_B)$ -anytime MAC code. At time t , both encoded messages are simultaneously transmitted to the receiver over a MAC. The receiver performs iterative joint decoding to recover both messages simultaneously. The factor graph used in the decoding process is shown in Fig. 6.1, where the code $\mathcal{C}_A$ and code $\mathcal{C}_B$ are connected through MAC state or factor nodes. Note that we consider a parallel scheduling decoding as described in Section 3.3 of Chapter 3.

6.3 Over Erasure Multiple Access Channel

In this section, we consider the proposed anytime MAC code over the adder followed by an erasure channel (see Subsection 2.1.3 of Chapter 2 for the details of the channel model). The anytime characteristics of the proposed coding scheme are investigated through density evolution analysis. We also derive the analytical delay-exponent for the proposed code by using mathematical induction technique.

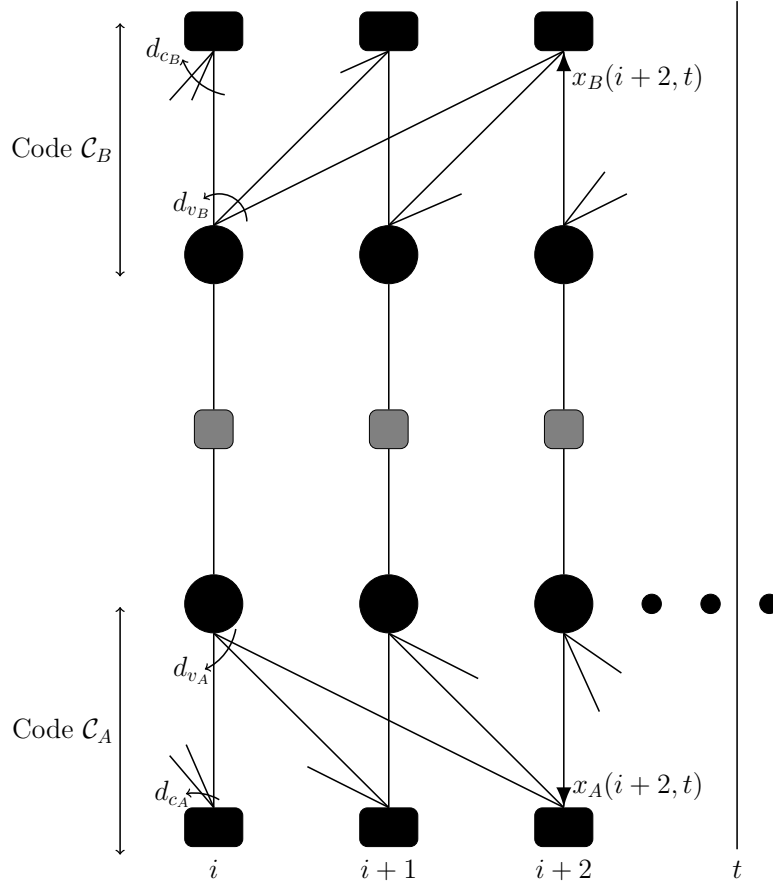


FIGURE 6.1: Proposed anytime code for 2-user multiple access channel. The state nodes are depicted as light gray nodes. The degrees of the nodes are depicted in the left side, while the variable node update messages used in the density evolution are depicted in the right side of the figure.

6.3.1 Density Evolution

In [59], it was shown that the adder followed by an erasure channel is equivalent to a binary erasure channel with erasure probability $\epsilon + \frac{1-\epsilon}{2}$. Let $x_j^{(\ell)}(i, t)$, $j \in \{A, B\}$ be the erasure probability of a message outgoing from variable nodes at position i to the check nodes of code $\mathcal{C}_j$ in the ℓ^{th} iteration, at time t . The density evolution equations for the proposed code can be derived by following the techniques used in [59] for spatially coupled code over the erasure MAC channel. The update equations for the proposed anytime MAC are presented in (6.2) and (6.3), where $f_A(i, t)$ represents

- $x_A^{(\ell)}(i, t)$ when $q_A = d_{v_A} - 1$,
- $P_{e_A}(i, t)$ when $q_A = d_{v_A}$ and $\ell = \infty$,

and $f_B(i, t)$ represents

$$f_A(i, t) = \left(\epsilon + \frac{1-\epsilon}{2} \left(1 - \sum_{j=0}^{t-i} P_{r_B}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_B}(k) x_B^{(\ell-1)}(i+j-k, t) \right)^{d_{c_B}-1} \right)^{d_{v_B}} \right. \\ \left. \cdot \left(1 - \sum_{j=0}^{t-i} P_{r_A}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_A}(k) x_A^{(\ell-1)}(i+j-k, t) \right)^{d_{c_A}-1} \right)^{q_A} \right), \quad (6.2)$$

$$f_B(i, t) = \left(\epsilon + \frac{1-\epsilon}{2} \left(1 - \sum_{j=0}^{t-i} P_{r_A}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_A}(k) x_A^{(\ell-1)}(i+j-k, t) \right)^{d_{c_A}-1} \right)^{d_{v_A}} \right. \\ \left. \cdot \left(1 - \sum_{j=0}^{t-i} P_{r_B}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_B}(k) x_B^{(\ell-1)}(i+j-k, t) \right)^{d_{c_B}-1} \right)^{q_B} \right), \quad (6.3)$$

where $P_{r_A}(k) = e^{-k\lambda_A}(1 - e^{-\lambda_A})$ and $P_{r_B}(k) = e^{-k\lambda_B}(1 - e^{-\lambda_B})$.

- $x_B^{(\ell)}(i, t)$ when $q_B = d_{v_B} - 1$,
- $P_{e_B}(i, t)$ when $q_B = d_{v_B}$ and $\ell = \infty$.

In (6.2) and (6.3), we consider $x_A^{(\ell)}(i, t) = x_B^{(\ell)}(i, t) = 0, \forall \ell$ when $i \leq 0$. Finally, we initialize the density evolution with the following messages.

$$x_A^{(0)}(i, t) = x_B^{(0)}(i, t) = \epsilon + \frac{1-\epsilon}{2} \quad \text{if } 0 < i \leq t. \quad (6.4)$$

We present the results from the above density evolution analysis. We consider a $(d_{v_A}, d_{v_B}, d_{c_A}, d_{c_B}, \lambda_A, \lambda_B) = (3, 6, 3, 6, 0.1, 0.1)$ -anytime MAC code. The erasure probability performances of both users are identical, since both users encode the messages with the same anytime SC-LDPC code. Fig. 6.2 shows the erasure probabilities of three messages versus the decoding time instant. We observe similar anytime performance as shown for the P2P communication scenario. For each of the messages, we observe an exponential decay of the erasure probability. We also observe that the performance of a message is independent of its index/position.

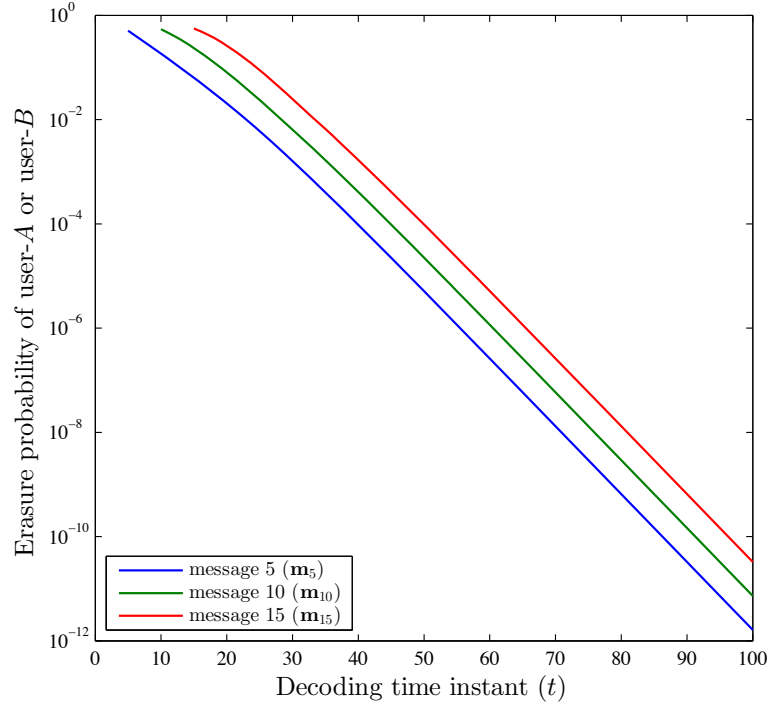


FIGURE 6.2: Asymptotic performance of the proposed code over the adder followed by an erasure channel ($\epsilon = 0.25$).

6.3.2 Delay-Exponent Analysis

We now present delay-exponent analysis over erasure MAC which is similar to the delay-exponent analysis of P2P and relay communications as shown in Chapter 4 and Chapter 5, respectively. The following theorem provides analytical formulas for the delay exponents of user-A and user-B.

Theorem 6.1. *Consider the $(d_{v_A}, d_{v_B}, d_{c_A}, d_{c_B}, \lambda_A, \lambda_B)$ -anytime MAC code as described in Section 6.2. Over the adder channel followed by an erasure channel, the delay-exponents of user-A and user-B are given by*

$$\alpha_A = \lambda_A d_{v_A} \quad \text{and} \quad \alpha_B = \lambda_B d_{v_B}, \quad (6.5)$$

when $\epsilon \leq \bar{\epsilon}_b$, where

$$\bar{\epsilon}_b \triangleq \min \left\{ \max_{\beta \geq \epsilon + \frac{1-\epsilon}{2}} \min_{d \geq 0} \frac{\beta (e^{-\lambda_A} + \mathcal{G}_A)^{-d_{v_A}} - \frac{\bar{\mathcal{G}}_B}{2}}{\left(1 - \frac{\bar{\mathcal{G}}_B}{2}\right)}, \right.$$

$$\max_{\beta \geq \epsilon + \frac{1-\epsilon}{2}} \min_{d \geq 0} \left\{ \frac{\beta \left(e^{-\lambda_B} + \mathcal{G}_B \right)^{-d_{v_B}} - \frac{\bar{\mathcal{G}}_A}{2}}{\left(1 - \frac{\bar{\mathcal{G}}_A}{2} \right)} \right\}.$$

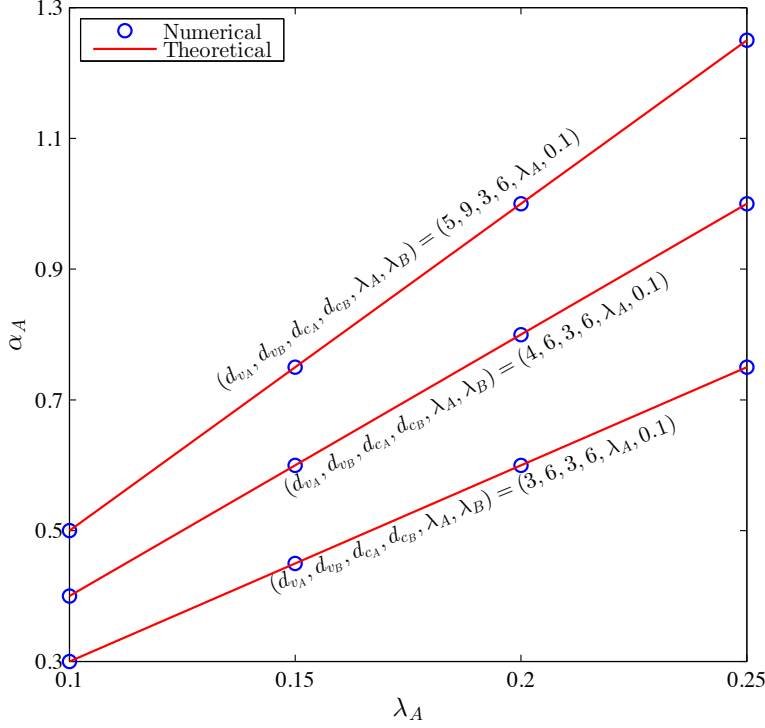


FIGURE 6.3: Comparison between theoretical and numerical delay-exponents of user-A.

The proof of the theorem and definitions of $\mathcal{G}_A$, $\bar{\mathcal{G}}_A$, $\mathcal{G}_B$, and $\bar{\mathcal{G}}_B$ are given in Appendix A.4. It can be shown numerically that user-A and user-B achieve the delay-exponents $\alpha_A = \lambda_A d_{v_A}$ and $\alpha_B = \lambda_B d_{v_B}$, respectively, if ϵ is below a threshold $\bar{\epsilon}$. We can consider $\bar{\epsilon}_b$ as a bound of the threshold $\bar{\epsilon}$. It is worth mentioning that there exists a large gap between $\bar{\epsilon}$ and $\bar{\epsilon}_b$ due to the limited tightness of the bounds used in the proof. For example, with a (3, 6, 0.1, 3, 6, 0.2)-anytime MAC code, the delay-exponents of user-A and user-B are $\alpha_A = 0.3$ and $\alpha_B = 0.6$, respectively while $\bar{\epsilon}$ (numerically obtained) and $\bar{\epsilon}_b$ are 0.331 and 0.092, respectively. In Fig. 6.3, we compare the theoretical exponent of user-A with the exponent obtained from the numerical results of density evolution. From Fig. 6.3, it is clear that the theoretically determined delay-exponent for each user exactly match the corresponding delay-exponent obtained from numerical results. For user-B, we also observed exact match between the theoretical delay-exponent $\alpha_B = \lambda_B d_{v_B}$ and numerical delay-exponent.

6.4 Over Gaussian Multiple Access Channel

We now extend the analysis of proposed anytime MAC code over 2-user Gaussian MAC (see Subsection 2.1.3 of Chapter 2 for the details of the channel model). For the asymptotic analysis, we follow the density evolution of SC-LDPC code over Gaussian MAC as shown in Section 3.3 of Chapter 3. In the density evolution, we consider the following notations at time t and at iteration ℓ .

- $x_j^{(\ell)}(i, t)$ denotes the mean of LLR message outgoing from the variable node at position i of user $j \in \{A, B\}$.
- $y_j^{(\ell)}(i, t)$ denotes the mean of LLR message outgoing from the check node at position i of user $j \in \{A, B\}$.
- $sv_j^{(\ell)}(i, t)$ denotes the mean of LLR message outgoing from the state node at position i to variable node at position i of user $j \in \{A, B\}$.
- $vs_j^{(\ell)}(i, t)$ denotes the mean of LLR message outgoing from the variable node at position i to state node at position i of user $j \in \{A, B\}$.

With the above notation, the Gaussian approximated density evolution update equations for the proposed anytime code over Gaussian MAC are given by:

- **Check node to variable node:** For user $j \in \{A, B\}$

$$y_j^{(\ell)}(i, t) = \phi^{-1} \left[1 - \left\{ 1 - \phi \left(\mathcal{J}^{-1} \left(\sum_{k=0}^{\infty} P_{r_j}(k) \mathcal{J} \left(x_j^{(\ell)}(i - k, t) \right) \right) \right) \right\}^{d_{c_j}-1} \right],$$

where $x_j^{(\ell)}(i, t) = \infty$ if $i \leq 0$. We initialize above update equations with the following messages:

$$x_j^{(0)}(i, t) = \mathcal{J}^{-1} \left(\frac{1}{3} C_T(\sigma_n) \right) \quad \text{if } 0 < i \leq t.$$

Recall that $C_T(\sigma_n)$ is the capacity of ternary input AWGN channel with input distribution $\frac{1}{4}\Delta_{-2} + \frac{1}{2}\Delta_0 + \frac{1}{4}\Delta_{+2}$ and variance σ_n^2 .

- **Variable node to state node:** For user $j \in \{A, B\}$

$$vs_j^{(\ell)}(i, t) = d_{v_j} \mathcal{J}^{-1} \left(\sum_{k=0}^{\infty} P_{r_j}(k) \mathcal{J} \left(y_j^{(\ell)}(i + k, t) \right) \right).$$

- **State node to variable node:** Following [58],

$$\begin{aligned} sv_A^{(\ell)}(i, t) &= \mathcal{J}^{-1} \left(\frac{1}{3} \left(1 + \mathcal{J} \left(vs_B^{(\ell)}(i, t) \right) \right) C_T(\sigma_n) \right), \\ sv_B^{(\ell)}(i, t) &= \mathcal{J}^{-1} \left(\frac{1}{3} \left(1 + \mathcal{J} \left(vs_A^{(\ell)}(i, t) \right) \right) C_T(\sigma_n) \right). \end{aligned}$$

- **Variable node to check node:** For user $j \in \{A, B\}$

$$x_j^{(\ell+1)}(i, t) = sv_j^{(\ell)}(i) + (d_{v_j} - 1) \mathcal{J}^{-1} \left(\sum_{k=0}^{t-i} P_{r_j}(k) \mathcal{J} \left(y_j^{(\ell)}(i + k, t) \right) \right).$$

For user- j ($j \in \{A, B\}$), the error probability of the message i at time t is i.e., $P_e^{[j]}(i, t)$ is given by

$$P_{e_j}(i, t) = Q \left(\sqrt{\frac{1}{2} \left(sv_j^{(\infty)}(i) + d_{v_j} \mathcal{J}^{-1} \left(\sum_{k=0}^{t-i} P_{r_j}(k) \mathcal{J} \left(y_j^{(\infty)}(i + k, t) \right) \right) \right)} \right).$$

Using the above density evolution analysis, we plot the bit error rate performance of user- A in Fig. 6.4. We consider a $(d_{v_A}, d_{v_B}, d_{c_A}, d_{c_B}, \lambda_A, \lambda_B) = (3, 6, 3, 6, 0.1, 0.1)$ -anytime MAC code with $\sigma_n = 0.7$. Similar to the performance over the erasure MAC, we observe the exponential decay of error probability over the Gaussian MAC for both users.

6.5 Finite-Length Performance

We now present the finite-length erasure/error performance of the proposed anytime MAC code. In Fig. 6.5, we show the performance of the proposed anytime MAC code for aforementioned MAC channels. In the simulation, we consider a $(d_{v_A}, d_{v_B}, d_{c_A}, d_{c_B}, \lambda_A, \lambda_B) = (3, 6, 3, 6, 0.1, 0.1)$ -anytime MAC code with

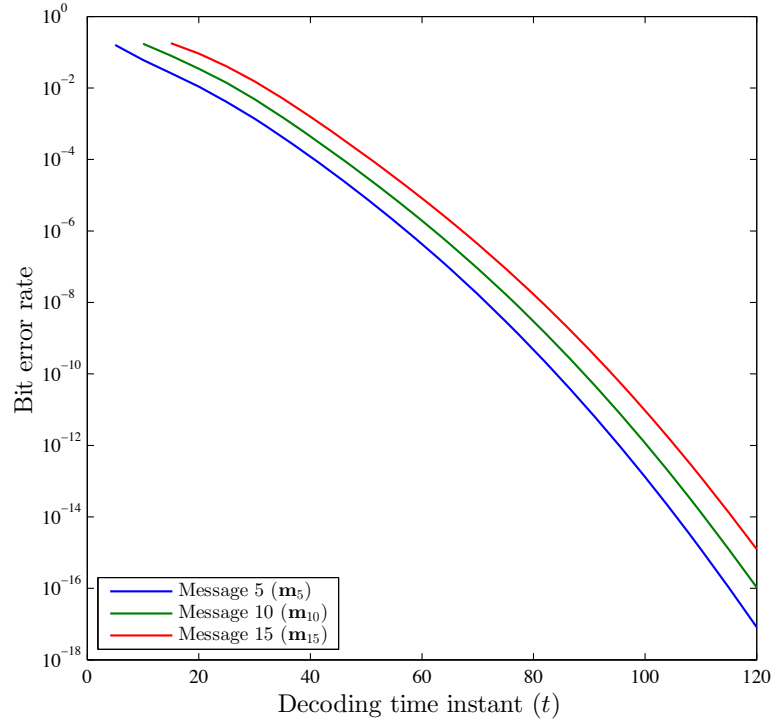


FIGURE 6.4: Asymptotic performance of the proposed anytime MAC code over 2-user Gaussian multiple access channel with $\sigma_n = 0.7$.

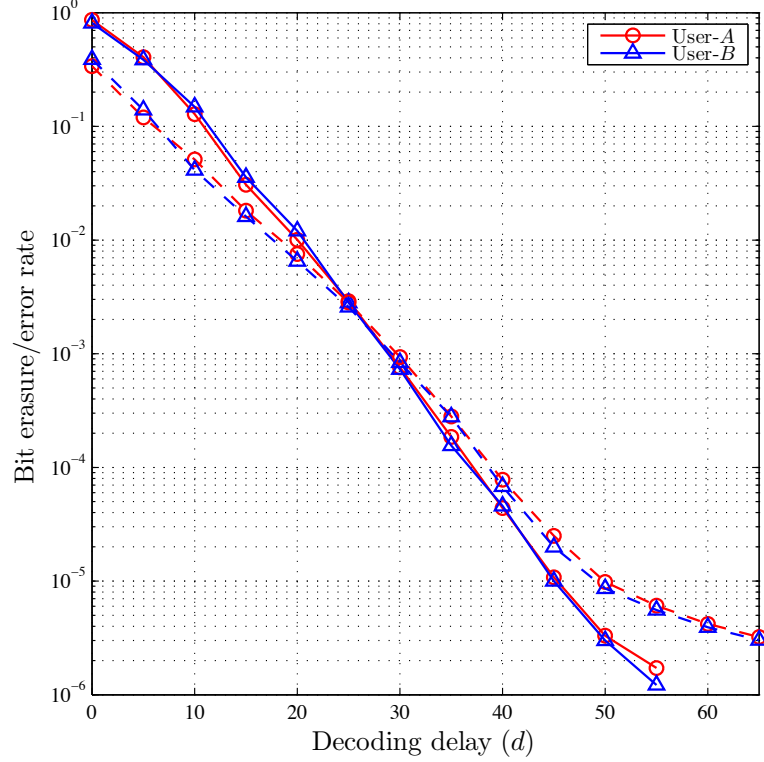


FIGURE 6.5: Finite-length performance of the proposed anytime MAC code with $M_A = M_B = 30$. Solid line and dashed line are for the erasure MAC ($\epsilon = 0.25$) and Gaussian MAC ($\sigma_n = 0.7$), respectively.

$M_A = M_B = 30$. We show the average erasure/error performance of message $\mathbf{m}_1$ for both users, where each curve is obtained by averaging over 10^6 simulation runs. We observe that for both users exhibit the exponential decay of the erasure/error probability with the decoding delay. However, error floor appears at the large decoding delay.

6.6 Conclusion

In this Chapter, we have proposed a practical anytime coding scheme for the 2-user multiple access channel. For two different MAC models, we have investigated the anytime characteristics of the proposed code through density evolution analysis. Moreover, we have found analytical expressions for the delay exponent for each user for the erasure multiple access channel.

Chapter 7

Conclusions

This thesis has focused on designing and analysing anytime channel coding scheme for point-to-point, relay, and multiple-access communications scenarios. Since limited tools are available to analyse the anytime code, we also focused to develop analytical tools, which will allow tractable analysis of the proposed anytime coding schemes.

In Chapter 3, we have developed analytical tools, which are applicable for both block and anytime codes. For the block codes, these tools give higher accuracy and/or less computational complexity than the corresponding existing tools. Additionally, these tools give a tractable analysis for the anytime codes. We have shown an algorithm which can predict the finite-length performance of LDPC codes with reasonable accuracy. A low complexity density evolution analysis is proposed for SC-LDPC codes over BIAWGN channel, which can predict the actual decoding threshold quite accurately. Design and analysis of SC-LDPC codes for Gaussian multiple access channel have also been studied.

A practical and high performance anytime coding scheme is proposed in Chapter 4 for point-to-point communication. The proposed anytime code is designed on spatially coupled codes. Anytime performance of the proposed is investigated over BEC and BIAWGN channels through density evolution analysis. For the BEC, the corresponding delay-exponent is derived, and the finite-length performance of the proposed anytime code is numerically predicted. Additionally, adaptive window decoding techniques is proposed to ensure low complexity anytime receivers.

Next, anytime coding scheme is proposed for three-terminal relay network. Considering two different transmission schemes in the relay-destination link, we have

investigated the anytime characteristics of the proposed code and derived the delay-exponents. For both schemes, the effect of the relay's delay on the overall performance is shown numerically. We have observed that the coded relaying scheme gives a better delay-exponent compare to the uncoded relaying scheme. To improve the performance of the uncoded relaying scheme, a modified scheme is proposed, which gives the same delay-exponent as that of the coded relaying scheme.

Finally, We have proposed a practical anytime coding scheme for the 2-user multiple access channel. Through asymptotic analysis, we have investigated the anytime characteristics of the proposed code for two different multiple access channels. We have shown analytical expressions for the delay exponent for each user and observed an exact match to the numerically obtained delay-exponent.

7.1 Future Research

In this thesis, we have presented design and analysis of practical anytime coding schemes for different communication scenarios. However, there exist many open topics regarding our anytime coding schemes that can be investigated in future. Some of these topics are given as follows:

- We have proposed the anytime codes for different communication scenarios. It will also be interesting to investigate the performance of our proposed coding schemes while applying in the practical controlling/tracking scenarios. For example our proposed anytime code can be applied to a simple control scenario like cart-stick balancer.
- An absolute performance assessment of the proposed coding scheme can be done for the point-to-point communication scenario by invoking the classical random coding error-exponent theory.
- We have analytically derived the delay-exponent for asymptotic case i.e., for infinite message length. The theoretical analysis of the delay-exponent for finite-length anytime codes needs to be investigated. The effect of cycles on the delay-exponent requires to be addressed, since cycles cannot be avoided in the finite-length code structure.

- We have proposed the bilayer anytime code for the relay network while considering binary erasure channels. It would be useful to investigate the application of our proposed bilayer code structure to AWGN channels. Unlike the BEC, a decoded bit can not be declared as recovered with full reliability for AWGN channels. Thus, crossover of the transmitted bits may occur at the relay and destination terminals, which may degrade the overall performance drastically. An appropriate method needs to be developed for the utilization of the unreliable decoded bits at the relay.
- Although achievable delay-exponent of CF relaying was studied in [15], the design of practical anytime code for CF relaying remains an open problem. The main challenge for this extension is the compression of the source's message at the relay.

Appendices

Appendix A

Proof of Theorems

A.1 Proof of Theorem 4.1

At iteration $\ell = \infty$, (4.3) and (4.4) can be written as follows:

$$f(i, t) = \epsilon \left(1 - \sum_{j=0}^{t-i} P_r(j) \left(1 - \sum_{k=0}^{\infty} P_r(k) x(i+j-k, t) \right)^{d_c-1} \right)^q \quad (\text{A.1})$$

Note that we drop the iteration index ℓ for the sake of simplicity. When $q = d_v - 1$, (A.1) represents the outgoing erasure probability from variable node i at decoding instant t . Similarly, when $q = d_v$, (A.1) represents the erasure probability of message $\mathbf{m}_i$ at decoding instant t . By using the following induction method, we find the delay-exponent for the proposed anytime code.

- **The basis:** For $0 \leq i \leq d$ and $\beta \geq \epsilon$, assume that

$$f(t-i, t) \leq \beta e^{-\alpha_1 i} \text{ where } \alpha_1 = \lambda q \quad (\text{A.2})$$

The basis is trivially correct for $d = 0$.

- **The inductive step:** The proof is complete if we can prove

$$f(t-d-1, t) \leq \beta e^{-\alpha_1(d+1)} \quad (\text{A.3})$$

In the following proof, we consider $P_r(k) = e^{-k\lambda}(1 - e^{-\lambda}) = Ke^{-k\lambda}$. From (A.1) we get,

$$\begin{aligned}
f(t-d-1, t) &= \epsilon \left(1 - \sum_{j=0}^{d+1} P_r(j) \left(1 - \sum_{k=0}^{\infty} P_r(k) x(t-d-1+j-k, t) \right)^{d_c-1} \right)^q \\
&\stackrel{(a)}{\leq} \epsilon \left(1 - \sum_{j=0}^{d+1} P_r(j) \left(1 - \beta \sum_{k=0}^{j-1} P_r(k) e^{-\alpha_1(d+1-j+k)} - \beta \sum_{k=j}^{\infty} P_r(k) e^{-\alpha_1 d} \right)^{d_c-1} \right)^q \\
&= \epsilon \left(1 - \sum_{j=0}^{d+1} P_r(j) \left(1 - \beta K \sum_{k=0}^{j-1} e^{-\lambda k - \alpha_1(d+1-j+k)} - \beta e^{-\lambda j - \alpha_1 d} \right)^{d_c-1} \right)^q \\
&= \epsilon \left(1 - \sum_{j=0}^{d+1} P_r(j) \left(1 - \beta K e^{-\alpha_1(d+1-j)} \frac{1 - e^{(-\lambda - \alpha_1)j}}{1 - e^{-\lambda - \alpha_1}} - \beta e^{-\lambda j - \alpha_1 d} \right)^{d_c-1} \right)^q \\
&\stackrel{(b)}{\leq} \epsilon \left(1 - \sum_{j=0}^{d+1} P_r(j) \left(1 - (d_c - 1) \left(\beta K e^{-\alpha_1(d+1-j)} \frac{1 - e^{(-\lambda - \alpha_1)j}}{1 - e^{-\lambda - \alpha_1}} + \beta e^{-\lambda j - \alpha_1 d} \right) \right) \right)^q \\
&= \epsilon \left(\sum_{j=d+2}^{\infty} P_r(j) + \sum_{j=0}^{d+1} P_r(j) (d_c - 1) \left(\beta K e^{-\alpha_1(d+1-j)} \frac{1 - e^{(-\lambda - \alpha_1)j}}{1 - e^{-\lambda - \alpha_1}} + \beta e^{-\lambda j - \alpha_1 d} \right) \right)^q \\
&= \epsilon \left(e^{-\lambda(d+2)} + \sum_{j=0}^{d+1} K e^{-\lambda j} (d_c - 1) \left(\beta K e^{-\alpha_1(d+1-j)} \frac{1 - e^{(-\lambda - \alpha_1)j}}{1 - e^{-\lambda - \alpha_1}} + \beta e^{-\lambda j - \alpha_1 d} \right) \right)^q \\
&= \epsilon \left(e^{-\lambda(d+2)} + e^{-\alpha_1(d+1)} (d_c - 1) \beta K \sum_{j=0}^{d+1} \left(K e^{(-\lambda + \alpha_1)j} \frac{1 - e^{(-\lambda - \alpha_1)j}}{1 - e^{-\lambda - \alpha_1}} + e^{-2\lambda j + \alpha_1} \right) \right)^q \\
&= \epsilon \left(e^{-\lambda(d+2)} + e^{-\alpha_1(d+1)} (d_c - 1) \beta K \right. \\
&\quad \cdot \left. \left(\frac{K}{1 - e^{-\lambda - \alpha_1}} \sum_{j=0}^{d+1} (e^{(-\lambda + \alpha_1)j} - e^{-2\lambda j}) + e^{\alpha_1} \sum_{j=0}^{d+1} e^{-2\lambda j} \right) \right)^q \\
&= \epsilon \left(e^{-\lambda(d+2)} + e^{-\alpha_1(d+1)} (d_c - 1) \beta K \right. \\
&\quad \cdot \left. \left(\frac{K}{1 - e^{-\lambda - \alpha_1}} \frac{1 - e^{(-\lambda + \alpha_1)(d+2)}}{1 - e^{-\lambda + \alpha_1}} + \left(e^{\alpha_1} - \frac{K}{1 - e^{-\lambda - \alpha_1}} \right) \frac{1 - e^{-2\lambda(d+2)}}{1 - e^{-2\lambda}} \right) \right)^q \\
&= \epsilon e^{-\lambda q(d+1)} (e^{-\lambda} + \mathcal{G})^q, \tag{A.4}
\end{aligned}$$

where

$$\mathcal{G} = e^{(\lambda - \alpha_1)(d+1)} (d_c - 1) \beta K$$

$$\cdot \left(\frac{K}{1 - e^{-\lambda - \alpha_1}} \frac{1 - e^{(-\lambda + \alpha_1)(d+2)}}{1 - e^{-\lambda + \alpha_1}} + \left(e^{\alpha_1} - \frac{K}{1 - e^{-\lambda - \alpha_1}} \right) \frac{1 - e^{-2\lambda(d+2)}}{1 - e^{-2\lambda}} \right) \quad (\text{A.5})$$

In step (a), we apply induction assumption and the fact that $x(i+k, t) \geq x(i, t)$ for all $k \geq 0$. In step (b), we apply $(1-z)^n \geq 1-nz$ for $|z| \leq 1$. With $\beta > 0$ and $\epsilon \leq \min_{d \geq 0} \beta (e^{-\lambda} + \mathcal{G})^{-q}$, from (A.4) we can prove (A.3). Substituting $q = d_v$ in (A.4) gives the following erasure probability,

$$P_e(t-d-1, t) \leq e^{-\lambda d_v(d+1)} \epsilon (e^{-\lambda} + \mathcal{G})^{d_v}.$$

Replacing $P_e(t-d-1, t)$ in (4.1) gives the delay-exponent as mentioned in (4.5) provided that $\epsilon \leq \bar{\epsilon}_b \triangleq \max_{\beta \geq \epsilon} \min_{d \geq 0} \beta (e^{-\lambda} + \mathcal{G})^{-d_v}$.

A.2 Proof of Theorem 5.1

Assuming full cooperation in *sr* link, at iteration $\ell = \infty$, (5.3) can be written as:

$$f_1(i, t) = \epsilon_{sd} \left(1 - \sum_{j=0}^{t-i} P_{r_1}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_1}(k) x_1(i+j-k, t) \right)^{d_{c_1}-1} \right)^{q_1} \cdot \left(1 - \sum_{j=0}^{t-i} P_{r_2}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_2}(k) x_2(i+j-k, t) \right)^{d_{c_2}-1} \right)^{q_2}. \quad (\text{A.6})$$

Note that we drop the iteration index ℓ for the sake of simplicity. We will prove the theorem by mathematical induction technique.

- **The basis:** For $0 \leq i \leq d$ and $\beta \geq \epsilon_{sd}$, assume that¹

$$x_1(t-i, t) \leq \beta e^{-\alpha_1 i}, \text{ where } \alpha_1 = \lambda_1(d_{v_1} - 1) + \lambda_2 d_{v_2}$$

$$x_2(t-i, t) \leq \beta e^{-\alpha_2 i}, \text{ where } \alpha_2 = \lambda_1 d_{v_1} + \lambda_2(d_{v_2} - 1)$$

$$P_e(t-i, t) \leq \beta e^{-\alpha i}, \text{ where } \alpha = \lambda_1 d_{v_1} + \lambda_2 d_{v_2}$$

The basis is trivially correct for $d = 0$.

¹These assumptions can be verified from numerical results.

- **The inductive step:** The proof is complete if we can prove

$$x_1(t-d-1, t) \leq \beta e^{-\alpha_1(d+1)} \quad (\text{A.8a})$$

$$x_2(t-d-1, t) \leq \beta e^{-\alpha_2(d+1)} \quad (\text{A.8b})$$

$$P_e(t-d-1, t) \leq \beta e^{-\alpha(d+1)} \quad (\text{A.8c})$$

For notational convenience, we denote $K_1 = 1 - e^{-\lambda_1}$, $K_2 = 1 - e^{-\lambda_2}$ and thus $P_{r_1}(k) = K_1 e^{-k\lambda_1}$ and $P_{r_2}(k) = K_2 e^{-k\lambda_2}$. Following (A.6)

$$\begin{aligned} f_1(t-d-1, t) &= \epsilon_{sd} \left(1 - \sum_{j=0}^{d+1} P_{r_1}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_1}(k) x_1(t-d-1+j-k, t) \right)^{d_{c_1}-1} \right)^{q_1} \\ &\quad \cdot \left(1 - \sum_{j=0}^{d+1} P_{r_2}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_2}(k) x_2(t-d-1+j-k, t) \right)^{d_{c_2}-1} \right)^{q_2} \\ &= \epsilon_{sd} A_1 A_2, \end{aligned} \quad (\text{A.9})$$

where

$$A_m = \left(1 - \sum_{j=0}^{d+1} P_{r_m}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_m}(k) x_m(t-d-1+j-k, t) \right)^{d_{c_m}-1} \right)^{q_m}, \quad (\text{A.10})$$

for $m \in \{1, 2\}$. Now from (A.10)

$$\begin{aligned} A_m &\stackrel{(a)}{\leq} \left(1 - \sum_{j=0}^{d+1} P_{r_m}(j) \left(1 - \beta \sum_{k=0}^{j-1} P_{r_m}(k) e^{-\alpha_m(d+1-j+k)} - \beta \sum_{k=j}^{\infty} P_{r_m}(k) e^{-\alpha_m d} \right)^{d_{c_m}-1} \right)^{q_m} \\ &= \left(1 - \sum_{j=0}^{d+1} P_{r_m}(j) \left(1 - \beta K_m \sum_{k=0}^{j-1} e^{-\lambda_m k - \alpha_m(d+1-j+k)} - \beta e^{-\lambda_m j - \alpha_m d} \right)^{d_{c_m}-1} \right)^{q_m} \\ &= \left(1 - \sum_{j=0}^{d+1} P_{r_m}(j) \left(1 - \beta K_m e^{-\alpha_m(d+1-j)} \frac{1 - e^{(-\lambda_m - \alpha_m)j}}{1 - e^{-\lambda_m - \alpha_m}} - \beta e^{-\lambda_m j - \alpha_m d} \right)^{d_{c_m}-1} \right)^{q_m} \\ &\stackrel{(b)}{\leq} \left(1 - \sum_{j=0}^{d+1} P_{r_m}(j) \left(1 - (d_{c_m} - 1) \left(\beta K_m e^{-\alpha_m(d+1-j)} \frac{1 - e^{(-\lambda_m - \alpha_m)j}}{1 - e^{-\lambda_m - \alpha_m}} + \beta e^{-\lambda_m j - \alpha_m d} \right) \right) \right)^{q_m} \\ &= \left(\sum_{j=d+2}^{\infty} P_{r_m}(j) + \sum_{j=0}^{d+1} P_{r_m}(j) (d_{c_m} - 1) \left(\beta K_m e^{-\alpha_m(d+1-j)} \frac{1 - e^{(-\lambda_m - \alpha_m)j}}{1 - e^{-\lambda_m - \alpha_m}} + \beta e^{-\lambda_m j - \alpha_m d} \right) \right)^{q_m} \end{aligned}$$

$$\begin{aligned}
 &= \left(e^{-\lambda_m(d+2)} + \sum_{j=0}^{d+1} K_m e^{-\lambda_m j} (d_{c_m} - 1) \right. \\
 &\quad \cdot \left. \left(\beta K_m e^{-\alpha_m(d+1-j)} \frac{1 - e^{(-\lambda_m - \alpha_m)j}}{1 - e^{-\lambda_m - \alpha_m}} + \beta e^{-\lambda_m j - \alpha_m d} \right) \right)^{q_m} \\
 &= \left(e^{-\lambda_m(d+2)} + e^{-\alpha_m(d+1)} (d_{c_m} - 1) \beta K_m \left(\frac{K_m}{1 - e^{-\lambda_m - \alpha_m}} \right. \right. \\
 &\quad \cdot \left. \left. \frac{1 - e^{(-\lambda_m + \alpha_m)(d+2)}}{1 - e^{-\lambda_m + \alpha_m}} + \left(e^{\alpha_m} - \frac{K_m}{1 - e^{-\lambda_m - \alpha_m}} \right) \frac{1 - e^{-2\lambda_m(d+2)}}{1 - e^{-2\lambda_m}} \right) \right)^{q_m}. \quad (\text{A.11})
 \end{aligned}$$

In step (a), we apply the induction assumption and the fact that $x_m(i+k, t) \geq x_m(i, t)$ for all $k \geq 0$. In step (b), we apply $(1-z)^n \geq 1-nz$ for $|z| \leq 1$. By substituting (A.11) in (A.9), we get

$$f_1(t-d-1, t) \leq \epsilon_{sd} e^{-(\lambda_1 q_1 + \lambda_2 q_2)(d+1)} (e^{-\lambda_1} + \mathcal{G}_1)^{q_1} (e^{-\lambda_2} + \mathcal{G}_2)^{q_2}, \quad (\text{A.12})$$

where

$$\begin{aligned}
 \mathcal{G}_m = & e^{(\lambda_m - \alpha_m)(d+1)} (d_{c_m} - 1) \beta K_m \left(\frac{K_m}{1 - e^{-\lambda_m - \alpha_m}} \right. \\
 & \cdot \left. \frac{1 - e^{(-\lambda_m + \alpha_m)(d+2)}}{1 - e^{-\lambda_m + \alpha_m}} + \left(e^{\alpha_m} - \frac{K_m}{1 - e^{-\lambda_m - \alpha_m}} \right) \frac{1 - e^{-2\lambda_m(d+2)}}{1 - e^{-2\lambda_m}} \right), \quad (\text{A.13})
 \end{aligned}$$

for $m \in \{1, 2\}$. With $\epsilon_{sd} \leq \min_d \beta (e^{-\lambda_1} + \mathcal{G}_1)^{-q_1} (e^{-\lambda_2} + \mathcal{G}_2)^{-q_2}$ and from (A.12), we can prove

- (A.8a), if $q_1 = d_{v_1} - 1$ and $q_2 = d_{v_2}$.
- (A.8b), if $q_1 = d_{v_1}$ and $q_2 = d_{v_2} - 1$.
- (A.8c), if $q_1 = d_{v_1}$ and $q_2 = d_{v_2}$.

From the proof of (A.8c), we get the delay-exponent mentioned in (5.4).

A.3 Proof of Theorem 5.2

Due to the anytime characteristics, we consider $\xi_{srd}(i) = \xi_{srd}(i+1) = \dots = \xi_{srd}$, while we ignore the boundary effect². With this assumption, at iteration $\ell = \infty$, (5.5) can be written as follows:

$$f_2(i, t) = \epsilon_{sd} \left(1 - \sum_{j=0}^{t-i} P_{r_1}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_1}(k) x_1(i+j-k, t) \right)^{d_{c_1}-1} \right)^{q_1} \cdot \left(1 - (1 - \xi_{srd}) \sum_{j=0}^{t-i-d_r} P_{r_2}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_2}(k) x_2(i+j-k, t) \right)^{d_{c_2}-1} \right)^{q_2}. \quad (\text{A.14})$$

We drop the iteration index ℓ for the sake of simplicity. By using the following mathematical induction, we prove the theorem 5.2.

- **The basis:** For $0 \leq i \leq d$ and $\beta \geq \epsilon_{sd}$,

$$x_1(t-i, t) \leq \beta e^{-\alpha_1 i}, \text{ where } \alpha_1 = \lambda_1(d_{v_1} - 1)$$

$$x_2(t-i, t) \leq \beta e^{-\alpha_2 i}, \text{ where } \alpha_2 = \lambda_1 d_{v_1}$$

$$P_e(t-i, t) \leq \beta e^{-\alpha i}, \text{ where } \alpha = \lambda_1 d_{v_1}$$

The basis is trivially correct for $d = 0$.

- **The inductive step:** The proof is complete if we can prove

$$x_1(t-d-1, t) \leq \beta e^{-\alpha_1(d+1)} \quad (\text{A.16a})$$

$$x_2(t-d-1, t) \leq \beta e^{-\alpha_2(d+1)} \quad (\text{A.16b})$$

$$P_e(t-d-1, t) \leq \beta e^{-\alpha(d+1)} \quad (\text{A.16c})$$

Note that several steps that are similar to that in the proof of Theorem 5.1 are skipped in this proof. Following (A.14)

$$f_2(t-d-1, t) = \epsilon_{sd} \left(1 - \sum_{j=0}^{d+1} P_{r_1}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_1}(k) x_1(t-d-1+j-k, t) \right)^{d_{c_1}-1} \right)^{q_1}$$

²Due to the code structure, the erasure probability of extra parity bits at the boundary of the chain (for example $\xi_{srd}(1)$ or $\xi_{srd}(2)$) will be slightly smaller than the erasure probability of other extra parity bits.

$$\begin{aligned}
 & \cdot \left(1 - (1 - \xi_{srd}) \sum_{j=0}^{d+1-d_r} P_{r_2}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_2}(k) x_2(t - d - 1 + j - k, t) \right)^{d_{c_2}-1} \right)^{q_2} \\
 & = \epsilon_{sd} A_1 A_3,
 \end{aligned} \tag{A.17}$$

where A_1 is defined in (A.10) with $m = 1$ and

$$\begin{aligned}
 A_3 & = \left(1 - (1 - \xi_{srd}) \sum_{j=0}^{d+1-d_r} P_{r_2}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_2}(k) x_2(t - d - 1 + j - k, t) \right)^{d_{c_2}-1} \right)^{q_2} \\
 & \stackrel{(a)}{\leq} \left(1 - (1 - \xi_{srd}) \sum_{j=0}^{d+1-d_r} P_{r_2}(j) \right. \\
 & \quad \cdot \left. \left(1 - \beta \sum_{k=0}^{j-1} P_{r_2}(k) e^{-\alpha_2(d+1-j+k)} - \beta \sum_{k=j}^{\infty} P_{r_2}(k) e^{-\alpha_2 d} \right)^{d_{c_2}-1} \right)^{q_2} \\
 & = \left(1 - (1 - \xi_{srd}) \sum_{j=0}^{d+1-d_r} P_{r_2}(j) \left(1 - \beta K_2 \sum_{k=0}^{j-1} e^{-\lambda_2 k - \alpha_2(d+1-j+k)} - \beta e^{-\lambda_2 j - \alpha_2 d} \right)^{d_{c_2}-1} \right)^{q_2} \\
 & \stackrel{(b)}{\leq} \left(1 - (1 - \xi_{srd}) \sum_{j=0}^{d+1-d_r} P_{r_2}(j) \right. \\
 & \quad \cdot \left. \left(1 - (d_{c_2} - 1) \beta \left(K_2 e^{-\alpha_2(d+1-j)} \frac{1 - e^{(-\lambda_2 - \alpha_2)j}}{1 - e^{-\lambda_2 - \alpha_2}} + e^{-\lambda_2 j - \alpha_2 d} \right) \right) \right)^{q_2} \\
 & = \left(e^{-\lambda_2(d+2-d_r)} + \xi_{srd} (1 - e^{-\lambda_2(d+2-d_r)}) + e^{-\alpha_2(d+1)} (d_{c_2} - 1) (1 - \xi_{srd}) \beta K_2 \right. \\
 & \quad \cdot \left. \left(\frac{K_2}{1 - e^{-\lambda_2 - \alpha_2}} \frac{1 - e^{(-\lambda_2 + \alpha_2)(d+2-d_r)}}{1 - e^{-\lambda_2 + \alpha_2}} + \left(e^{\alpha_2} - \frac{K_2}{1 - e^{-\lambda_2 - \alpha_2}} \right) \frac{1 - e^{-2\lambda_2(d+2-d_r)}}{1 - e^{-2\lambda_2}} \right) \right)^{q_2}.
 \end{aligned}$$

In step (a), we apply induction assumption and the fact that $x_2(i + k, t) \geq x_2(i, t)$ for all $k \geq 0$. In step (b), we apply $(1 - z)^n \geq 1 - nz$ for $|z| \leq 1$. By replacing the bounds of A_1 and A_3 in (A.17), we obtain

$$f_2(t - d - 1, t) \leq \epsilon_{sd} e^{-\lambda_1 q_1(d+1)} (e^{-\lambda_1} + \mathcal{G}_1)^{q_1} (\xi_{srd} (1 - e^{-\lambda_2(d+2-d_r)}) + \mathcal{G}_3)^{q_2}, \tag{A.18}$$

where

$$\mathcal{G}_3 = e^{-\lambda_2(d+2-d_r)} + e^{-\alpha_2(d+1)}(d_{c_2} - 1)(1 - \xi_{srd})\beta K_2$$

$$\cdot \left(\frac{K_2}{1 - e^{-\lambda_2 - \alpha_2}} \frac{1 - e^{(-\lambda_2 + \alpha_2)(d+2-d_r)}}{1 - e^{-\lambda_2 + \alpha_2}} + \left(e^{\alpha_2} - \frac{K_2}{1 - e^{-\lambda_2 - \alpha_2}} \right) \frac{1 - e^{-2\lambda_2(d+2-d_r)}}{1 - e^{-2\lambda_2}} \right).$$
(A.19)

With $\epsilon_{sd} \leq \min_{d \geq 0} \beta (e^{-\lambda_1} + \mathcal{G}_1)^{-q_1} (\xi_{srd} (1 - e^{-\lambda_2(d+2-d_r)}) + \mathcal{G}_3)^{-q_2}$ and from (A.18), we can prove

- (A.16a), if $q_1 = d_{v_1} - 1$ and $q_2 = d_{v_2}$.
- (A.16b), if $q_1 = d_{v_1}$ and $q_2 = d_{v_2} - 1$.
- (A.16c), if $q_1 = d_{v_1}$ and $q_2 = d_{v_2}$.

From the proof of (A.16c), we obtain the delay-exponent (5.6).

A.4 Proof of Theorem 6.1

By using the following mathematical induction technique, we find the delay-exponents α_A and α_B as mentioned in Theorem 6.1.

- **The basis:** For $0 \leq i \leq d$ and $\beta \geq \epsilon + \frac{1-\epsilon}{2}$,

$$x_A(t - i, t) \leq \beta e^{-i\alpha_{x_A}}, \text{ where } \alpha_{x_A} = \lambda_A(d_{v_A} - 1)$$

$$x_B(t - i, t) \leq \beta e^{-i\alpha_{x_B}}, \text{ where } \alpha_{x_B} = \lambda_B(d_{v_B} - 1)$$

$$P_e^A(t - i, t) \leq \beta e^{-i\alpha_A}, \text{ where } \alpha_A = \lambda_A d_{v_A}$$

$$P_e^B(t - i, t) \leq \beta e^{-i\alpha_B}, \text{ where } \alpha_B = \lambda_B d_{v_B}$$

The basis is trivially correct for $d = 0$.

- **The inductive step:** We have to prove:

$$x_A(t - d - 1, t) \leq \beta e^{-\alpha_{x_A}(d+1)} \tag{A.21a}$$

$$x_B(t - d - 1, t) \leq \beta e^{-\alpha_{x_B}(d+1)} \tag{A.21b}$$

$$P_e^A(t - d - 1, t) \leq \beta e^{-\alpha_A(d+1)} \tag{A.21c}$$

$$P_e^B(t - d - 1, t) \leq \beta e^{-\alpha_B(d+1)} \tag{A.21d}$$

For notational convenience, we denote $K_A = 1 - e^{-\lambda_A}$, $K_B = 1 - e^{-\lambda_B}$. For $m \in \{A, B\}$, let us define

$$A_m = 1 - \sum_{j=0}^{d+1} P_{r_m}(j) \left(1 - \sum_{k=0}^{\infty} P_{r_m}(k) x_m(t - d - 1 + j - k, t) \right)^{d_{cm}-1} \quad (\text{A.22})$$

From (6.2) and (6.3), we can write

$$f_A(t - d - 1, t) = \left(\epsilon + \frac{1 - \epsilon}{2} (A_B)^{d_{v_B}} \right) (A_A)^{q_A}, \quad (\text{A.23})$$

$$f_B(t - d - 1, t) = \left(\epsilon + \frac{1 - \epsilon}{2} (A_A)^{d_{v_A}} \right) (A_B)^{q_B}. \quad (\text{A.24})$$

Following (A.22)

$$\begin{aligned} A_m &\stackrel{(a)}{\leq} 1 - \sum_{j=0}^{d+1} P_{r_m}(j) \left(1 - \beta \sum_{k=0}^{j-1} P_{r_m}(k) e^{-\alpha_{x_m}(d+1-j+k)} - \beta \sum_{k=j}^{\infty} P_{r_m}(k) e^{-\alpha_{x_m}d} \right)^{d_{cm}-1} \\ &\stackrel{(b)}{\leq} 1 - \sum_{j=0}^{d+1} P_{r_m}(j) \left(1 - (d_{cm} - 1) \left(\beta K_m e^{-\alpha_{x_m}(d+1-j)} \frac{1 - e^{(-\lambda_m - \alpha_{x_m})j}}{1 - e^{-\lambda_m - \alpha_{x_m}}} + \beta e^{-\lambda_m j - \alpha_{x_m}d} \right) \right) \\ &= e^{-\lambda_m(d+2)} + e^{-\alpha_{x_m}(d+1)} (d_{cm} - 1) \beta K_m \\ &\quad \cdot \left(\frac{K_m}{1 - e^{-\lambda_m - \alpha_{x_m}}} \frac{1 - e^{(-\lambda_m + \alpha_{x_m})(d+2)}}{1 - e^{-\lambda_m + \alpha_{x_m}}} + \left(e^{\alpha_{x_m}} - \frac{K_m}{1 - e^{-\lambda_m - \alpha_{x_m}}} \right) \frac{1 - e^{-2\lambda_m(d+2)}}{1 - e^{-2\lambda_m}} \right) \end{aligned} \quad (\text{A.25})$$

In step (a), we apply induction assumption and the fact that $x_m(i+k, t) \geq x_m(i, t)$ for all $k \geq 0$. In step (b), we apply $(1 - z)^n \geq 1 - nz$ for $|z| \leq 1$. By substituting (A.25) in (A.23) and (A.24), we get

$$f_A(t - d - 1, t) \leq e^{-\lambda_A q_A} \left(\epsilon \left(1 - \frac{\bar{\mathcal{G}}_B}{2} \right) + \frac{\bar{\mathcal{G}}_B}{2} \right) (e^{-\lambda_A} + \mathcal{G}_A)^{q_A}, \quad (\text{A.26})$$

$$f_B(t - d - 1, t) \leq e^{-\lambda_B q_B} \left(\epsilon \left(1 - \frac{\bar{\mathcal{G}}_A}{2} \right) + \frac{\bar{\mathcal{G}}_A}{2} \right) (e^{-\lambda_B} + \mathcal{G}_B)^{q_B}, \quad (\text{A.27})$$

where for $m \in \{A, B\}$

$$\begin{aligned} \bar{\mathcal{G}}_m &= \left(e^{-\lambda_m(d+2)} + e^{-\alpha_{x_m}(d+1)} (d_{cm} - 1) \beta K_m \left(\frac{K_m}{1 - e^{-\lambda_m - \alpha_{x_m}}} \frac{1 - e^{(-\lambda_m + \alpha_{x_m})(d+2)}}{1 - e^{-\lambda_m + \alpha_{x_m}}} + \right. \right. \\ &\quad \left. \left. \left(e^{\alpha_{x_m}} - \frac{K_m}{1 - e^{-\lambda_m - \alpha_{x_m}}} \right) \frac{1 - e^{-2\lambda_m(d+2)}}{1 - e^{-2\lambda_m}} \right) \right)^{d_{vm}} \end{aligned}$$

and

$$\mathcal{G}_m = e^{(\lambda_m - \alpha_{x_m})(d+1)}(d_{c_m} - 1)\beta K_m \left(\frac{K_m}{1 - e^{-\lambda_m - \alpha_{x_m}}} \frac{1 - e^{(-\lambda_m + \alpha_{x_m})(d+2)}}{1 - e^{-\lambda_m + \alpha_{x_m}}} + \left(e^{\alpha_{x_m}} - \frac{K_m}{1 - e^{-\lambda_m - \alpha_{x_m}}} \right) \frac{1 - e^{-2\lambda_m(d+2)}}{1 - e^{-2\lambda_m}} \right).$$

From (A.26) we can prove

- (A.21a), if $q_A = d_{v_A} - 1$,
- (A.21b), if $q_A = d_{v_A}$,

and from (A.27), we can prove

- (A.21c), if $q_B = d_{v_B} - 1$,
- (A.21d), if $q_B = d_{v_B}$,

when

$$\epsilon \leq \min \left\{ \min_{d \geq 0} \frac{\beta (e^{-\lambda_A} + \mathcal{G}_A)^{-q_A} - \frac{\bar{\mathcal{G}}_B}{2}}{\left(1 - \frac{\bar{\mathcal{G}}_B}{2}\right)}, \min_{d \geq 0} \frac{\beta (e^{-\lambda_B} + \mathcal{G}_B)^{-q_B} - \frac{\bar{\mathcal{G}}_A}{2}}{\left(1 - \frac{\bar{\mathcal{G}}_A}{2}\right)} \right\}.$$

From the proof of (A.21b) and (A.21d), we get the delay-exponents mentioned in Theorem 6.1.

Bibliography

- [1] C. E. Shannon, “A mathematical theory of communication,” *Bell system technical journal*, vol. 27, 1948.
- [2] A. Sahai, “Anytime information theory,” Ph.D. dissertation, Department of Electrical Engineering and Computer Science, MIT, 2001.
- [3] L. Schulman, “Coding for interactive communication,” *IEEE Trans. on Inform. Theory*, vol. 42, no. 6, pp. 1745 –1756, Nov. 1996.
- [4] R. Ostrovsky, Y. Rabani, and L. Schulman, “Error-correcting codes for automatic control,” *IEEE Trans. on Inform. Theory*, vol. 55, no. 7, pp. 2931 –2941, July 2009.
- [5] D. Teneketzis, “On the structure of optimal real-time encoders and decoders in noisy communication,” *IEEE Trans. on Inform. Theory*, vol. 52, no. 9, pp. 4017 –4035, Sept. 2006.
- [6] L. Bao, A. Shirazinia, and M. Skoglund, “Iterative encoder-controller design based on approximate dynamic programming,” in *Proc. IEEE International Workshop on Signal Processing Advances in Wireless Comm.*, June 2010, pp. 1 –5.
- [7] A. Sahai and S. Mitter, “The necessity and sufficiency of anytime capacity for stabilization of a linear system over a noisy communication link ; part I : Scalar systems,” *IEEE Trans. on Inform. Theory*, vol. 52, no. 8, pp. 3369 –3395, Aug. 2006.
- [8] R. Sukhavasi and B. Hassibi, “Linear error correcting codes with anytime reliability,” in *Proc. IEEE International Symposium on Inform. Theory (ISIT)*, Aug. 2011, pp. 1748 –1752.
- [9] A. Shirazinia, L. Bao, and M. Skoglund, “Anytime source transmission using UEP-LT channel coding,” in *Proc. European Wireless Conf. - Sustainable Wireless Technologies*, Apr. 2011, pp. 1 –6.

- [10] L. Grosjean, L. Rasmussen, R. Thobaben, and M. Skoglund, "Systematic LDPC convolutional codes: Asymptotic and finite-length anytime properties," *IEEE Trans. on Comm.*, vol. 62, no. 12, pp. 4165–4183, Dec 2014.
- [11] A. Tarable, A. Nordin, F. Dabbene, and R. Tempo, "Anytime reliable LDPC convolutional codes for networked control over wireless channel," in *Proc. IEEE International Symposium on Inform. Theory (ISIT)*, July 2013, pp. 2064–2068.
- [12] H. Şimşek, "Anytime channel coding with feedback," Ph.D. dissertation, University of California, Berkeley, 2004.
- [13] K. D. Nguyen and L. Rasmussen, "Delay-universal decode-and-forward relaying," in *Proc. Australian Comm. Theory Workshop (AusCTW)*, 2011, pp. 170–175.
- [14] K. Nguyen and L. Rasmussen, "Delay-exponent of decode-forward streaming," in *Proc. IEEE International Symposium on Inform. Theory (ISIT)*, July 2013, pp. 2805–2809.
- [15] K. D. Nguyen, "Delay-exponent of delay-universal compress-and-forward relaying," in *Proc. IEEE International Symposium on Inform. Theory (ISIT)*, 2012, pp. 2851–2855.
- [16] C. Chang and A. Sahai, "Sequential random coding error exponents for multiple access channels," in *Proc. International Conference on Wireless Networks, Comm. and Mobile Computing*, vol. 2, June 2005, pp. 1581–1586 vol.2.
- [17] S. J. Johnson, *Iterative Error Correction: Turbo, Low-Density Parity-Check and Repeat-Accumulate Codes*. Cambridge University Press, Jan. 2010.
- [18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, USA: Wiley-Interscience, 1991.
- [19] E. C. V. D. Meulen, "Three-terminal communication channels," *Advances in Applied Probability*, vol. 3, no. 1, pp. 120–154, 1971. [Online]. Available: <http://www.jstor.org/stable/1426331>
- [20] T. Cover and A. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. on Inform. Theory*, vol. 25, no. 5, pp. 572–584, 1979.
- [21] T. K. Moon, *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley-Interscience, 2005.

-
- [22] R. G. Gallager, *Low-Density Parity-Check Codes*. Massachusetts Institute of Technology, 1963.
 - [23] D. MacKay and R. Neal, “Near shannon limit performance of low density parity check codes,” *Electronics Letters*, vol. 33, no. 6, pp. 457–458, Mar. 1997.
 - [24] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. on Inform. Theory*, vol. 27, no. 5, pp. 533–547, Sep 1981.
 - [25] T. Richardson, M. Shokrollahi, and R. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. on Inform. Theory*, vol. 47, no. 2, pp. 619–637, Feb 2001.
 - [26] A. Roumy, S. Guemghar, G. Caire, and S. Verdu, “Design methods for irregular repeat-accumulate codes,” *IEEE Trans. on Inform. Theory*, vol. 50, no. 8, pp. 1711–1727, Aug 2004.
 - [27] A. Felstrom and K. Zigangirov, “Time-varying periodic convolutional codes with low-density parity-check matrix,” *IEEE Trans. on Inform. Theory*, vol. 45, no. 6, pp. 2181–2191, Sept. 1999.
 - [28] R. Tanner, “Convolutional codes from quasi-cyclic codes: a link between the theories of block and convolutional codes,” Tech. Rep., 1987.
 - [29] R. Tanner, D. Sridhara, A. Sridharan, T. Fuja, and J. Costello, D.J., “Ldpc block and convolutional codes based on circulant matrices,” *IEEE Trans. on Inform. Theory*, vol. 50, no. 12, pp. 2966 – 2984, Dec. 2004.
 - [30] G. Richter, M. Kaupper, and K. Zigangirov, “Irregular low-density parity-check convolutional codes based on protographs,” in *Proc. IEEE International Symposium on Inform. Theory (ISIT)*, July 2006, pp. 1633 –1637.
 - [31] J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” *JPL IPN Progress Report*, pp. 42–154, 2003.
 - [32] S. Kudekar, T. Richardson, and R. Urbanke, “Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC,” *IEEE Trans. on Inform. Theory*, vol. 57, no. 2, pp. 803 –834, Feb. 2011.
 - [33] S. Johnson and G. Lechner, “Spatially coupled repeat-accumulate codes,” *IEEE Comm. Letters*, vol. 17, no. 2, pp. 373–376, 2013.

- [34] A. Pusane, A. Feltstrom, A. Sridharan, M. Lentmaier, K. Zigangirov, and D. Costello, "Implementation aspects of LDPC convolutional codes," *IEEE Trans. on Comm.*, vol. 56, no. 7, pp. 1060–1069, July 2008.
- [35] A. Iyengar, M. Papaleo, P. Siegel, J. Wolf, A. Vanelli-Coralli, and G. Corazza, "Windowed decoding of protograph-based LDPC convolutional codes over erasure channels," *IEEE Trans. on Inform. Theory*, vol. 58, no. 4, pp. 2303–2320, Apr. 2012.
- [36] T. Richardson and R. Urbanke, *Modern Coding Theory*. New York, NY, USA: Cambridge University Press, 2008.
- [37] S. Kudekar, C. Measson, T. Richardson, and R. Urbanke, "Threshold saturation on BMS channels via spatial coupling," in *Proc. International Symposium on Turbo Codes and Iterative Inform. Processing*, 2010, pp. 309–313.
- [38] S. Kudekar, T. Richardson, and R. Urbanke, "Spatially coupled ensembles universally achieve capacity under belief propagation," *IEEE Trans. on Inform. Theory*, vol. 59, no. 12, pp. 7761–7813, Dec. 2013.
- [39] S. Kumar, A. Young, N. Macris, and H. Pfister, "Threshold saturation for spatially coupled LDPC and LDGM codes on BMS channels," *IEEE Trans. on Inform. Theory*, vol. 60, no. 12, pp. 7389–7415, Dec. 2014.
- [40] A. Yedla, Y.-Y. Jian, P. Nguyen, and H. Pfister, "A simple proof of maxwell saturation for coupled scalar recursions," *IEEE Trans. on Inform. Theory*, vol. 60, no. 11, pp. 6943–6965, Nov. 2014.
- [41] P. Razaghi and W. Yu, "Bilayer low-density parity-check codes for decode-and-forward in relay channels," *IEEE Trans. on Inform. Theory*, vol. 53, no. 10, pp. 3723–3739, 2007.
- [42] M. Azmi, J. Yuan, G. Lechner, and L. Rasmussen, "Design of multi-edge-type bilayer-expurgated LDPC codes for decode-and-forward in relay channels," *IEEE Trans. on Comm.*, vol. 59, no. 11, pp. 2993–3006, 2011.
- [43] J. Ha, J. Kim, and S. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Trans. on Inform. Theory*, vol. 50, no. 11, pp. 2824–2836, 2004.
- [44] J. Hu and T. Duman, "Low density parity check codes over wireless relay channels," *IEEE Trans. on Wireless Comm.*, vol. 6, no. 9, pp. 3384–3394, 2007.

-
- [45] A. Chakrabarti, A. de Baynast, A. Sabharwal, and B. Aazhang, “Low density parity check codes for the relay channel,” *IEEE Journal on Selected Areas in Comm.*, vol. 25, no. 2, pp. 280–291, 2007.
 - [46] M. Lentmaier, M. Prenda, and G. Fettweis, “Efficient message passing scheduling for terminated LDPC convolutional codes,” in *Proc. IEEE International Symposium on Inform. Theory (ISIT)*, July 2011, pp. 1826–1830.
 - [47] D. G. M. Mitchell, M. Lentmaier, and D. J. C. Jr., “Spatially coupled LDPC codes constructed from protographs,” *CoRR*, vol. abs/1407.5366, 2014. [Online]. Available: <http://arxiv.org/abs/1407.5366>
 - [48] S.-Y. Chung, T. Richardson, and R. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Trans. on Inform. Theory*, vol. 47, no. 2, pp. 657–670, 2001.
 - [49] S. Chung, “On the construction of some capacity-approaching coding schemes,” Ph.D. dissertation, Department of Electrical Engineering and Computer Science, MIT, 2000.
 - [50] D. Divsalar, S. Dolinar, C. Jones, and K. Andrews, “Capacity-approaching protograph codes,” *IEEE Journal on Selected Areas in Comm.*, vol. 27, no. 6, pp. 876–888, Aug. 2009.
 - [51] C. Di, D. Proietti, I. Telatar, T. Richardson, and R. Urbanke, “Finite-length analysis of low-density parity-check codes on the binary erasure channel,” *IEEE Trans. on Inform. Theory*, vol. 48, no. 6, pp. 1570–1579, Jun. 2002.
 - [52] J. Zhang and A. Orlitsky, “Finite-length analysis of LDPC codes with large left degrees,” in *Proc. IEEE International Symposium on Inform. Theory (ISIT)*, 2002, p. 3.
 - [53] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke, “Finite-length scaling for iteratively decoded LDPC ensembles,” *IEEE Trans. on Inform. Theory*, vol. 55, no. 2, pp. 473–498, Feb. 2009.
 - [54] R. Yazdani and M. Ardakani, “Waterfall performance analysis of finite-length LDPC codes on symmetric channels,” *IEEE Trans. on Comm.*, vol. 57, no. 11, pp. 3183–3187, Nov. 2009.
 - [55] M. Noor-A-Rahim, K. Nguyen, and G. Lechner, “Anytime reliability of spatially coupled codes,” *IEEE Trans. on Comm.*, vol. 63, no. 4, pp. 1069–1080, April 2015.

- [56] B. Rimoldi and R. Urbanke, "A rate-splitting approach to the gaussian multiple-access channel," *IEEE Trans. on Inform. Theory*, vol. 42, no. 2, pp. 364–375, Mar 1996.
- [57] A. Amraoui, S. Dusad, and R. Urbanke, "Achieving general points in the 2-user gaussian MAC without time-sharing or rate-splitting by means of iterative coding," in *Proc. IEEE International Symposium on Inform. Theory (ISIT)*, 2002, p. 334.
- [58] A. Roumy and D. Declercq, "Characterization and optimization of ldpc codes for the 2-user gaussian multiple access channel." *EURASIP Journal on Wireless Comm. and Networking*, vol. 2007, 2007. [Online]. Available: <http://dblp.uni-trier.de/db/journals/ejwcn/ejwcn2007.html#RoumyD07>
- [59] S. Kudekar and K. Kasai, "Spatially coupled codes over the multiple access channel," in *Proc. IEEE International Symposium on Inform. Theory (ISIT)*, July 2011, pp. 2816–2820.
- [60] A. Yedla, P. Nguyen, H. Pfister, and K. Narayanan, "Universal codes for the gaussian MAC via spatial coupling," in *Proc. Annual Allerton Conference on Comm., Control, and Computing*, Sept 2011, pp. 1801–1808.
- [61] A. Balatsoukas-Stimming, "Design of LDPC codes for the two-user gaussian multiple access channel," 2012. [Online]. Available: <http://www.telecom.tuc.gr/~alex/mscthesis.pdf>
- [62] M. Noor-A-Rahim, G. Lechner, and K. Nguyen, "Density evolution analysis of spatially coupled LDPC codes over BIAWGN channel," 2015. [Online]. Available: http://www.gottfriedlechner.com/homepage/Publications_files/DESCLDPCAWGN.pdf
- [63] M. Noor-A-Rahim, K. Nguyen, and G. Lechner, "Finite length analysis of LDPC codes," in *Proc. IEEE Wireless Comm. and Networking Conference (WCNC)*, April 2014, pp. 206–211.
- [64] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical loss-resilient codes," in *Proc. ACM Symposium on Theory of Computing*, 1997, pp. 150–159.
- [65] Z. Si, R. Thobaben, and M. Skoglund, "Bilayer LDPC convolutional codes for decode-and-forward relaying," *IEEE Trans. on Comm.*, vol. 61, no. 8, pp. 3086–3099, August 2013.