An Algebraic Approach to Physical-Layer Network Coding

by

Chen Feng

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

Abstract

An Algebraic Approach to Physical-Layer Network Coding

Chen Feng

Doctor of Philosophy

Graduate Department of Electrical and Computer Engineering

University of Toronto

2014

Physical-layer network coding (PNC) is a new approach to harnessing interference in wireless networks. Rather than avoiding interference or treating it as noise, PNC enables wireless relays to compute linear functions of the transmitted messages directly from the interfering signals. This allows for concurrent transmissions, thereby increasing network throughput.

This dissertation studies a family of the state-of-the-art PNC schemes called compute-and-forward (C&F). C&F was originally proposed and studied from an information-theoretic perspective. As such, it typically relies on several strong assumptions: very long block length, almost unbounded complexity, perfect channel state information, and no decoding errors at the relays; its benefit is often analyzed for simple network configurations. The aim of this dissertation is two-fold: first, to relax the above assumptions while preserving the performance of C&F, and second, to understand the benefit of C&F in more realistic network scenarios, such as random-access wireless networks.

There are four main results in this dissertation. First, an algebraic framework is developed, which establishes a direct connection between C&F and module theory. This connection allows us to systematically design lattice codes for C&F with controlled block length and complexity. In particular, explicit design criteria are derived, concrete design examples are provided, and it is shown that nominal coding gains from 3 to 7.5 dB can be obtained with relatively short block length and reasonable decoding complexity. Second, a new C&F scheme is proposed, which, unlike conventional C&F schemes, does not require any channel state information (CSI). It is shown that this CSI-free scheme achieves, for a certain class of lattice codes, almost the same throughput as its CSI-enabled counterpart. Third, an end-to-end error control mechanism is designed, which effectively mitigates decoding errors introduced at wireless relays. In particular, the end-to-end error control problem is modeled as a finite-ring matrix channel problem, for which tight capacity bounds and capacity-approaching schemes are provided. The final part of this dissertation studies the benefit of C&F in random-access wireless networks. In particular, it is shown that C&F significantly improves the network throughput and delay performance of slotted-ALOHA-based random-access protocols.

Acknowledgements

This thesis is the result of the support, help, and advice that I have received from an amazing group of scholars and students at the University of Toronto. Without them this dissertation wouldn't have been possible.

First, I feel very fortunate to have two great supervisors: Prof. Frank R. Kschischang and Prof. Baochun Li. As an advisor, Frank has consistently exceeded my expectation. He is not only a brilliant researcher and excellent teacher, but also truly a kind and caring person. He has given me a tremendous amount in the past six years, I can only thank him for a subset. In particular, I have benefited greatly from his focus on fundamental problems, his taste for beautiful mathematics, his emphasis on clarity and grace, and his dedication in developing his students to their full potential. I am very grateful to Frank for making my PhD journey so exciting and rewarding! Frank also has a great sense of humor. I will always remember how much fun we had together during our endless discussions on research and many other things.

I am likewise deeply indebted to Prof. Baochun Li. After eight years, I still do not understand how someone can accomplish so much and manage so many things, yet remain so relaxed (and so popular on Sina Weibo). I truly admire his skill of time management, his unique style of presentation, his vision of doing great research, and his pursuit of simple, elegant, and beautiful solutions. Over the years, Baochun has been a constant source of inspiration to me. This dissertation reflects some of his research philosophies, especially his emphasis on bringing theoretical results to practical implementations.

I had a wonderful experience collaborating with Danilo Silva and Roberto W. Nóbrega. Danilo's quick insights and sharp thinking got us through a number of papers and into all sorts of interesting discussions. He is not only a wonderful collaborator, but also a lifetime friend to me. I cannot thank him enough for his encouragement and support during my PhD journey. Roberto's patient listening and careful thinking made for wonderful collaborations on matrices over finite chain rings.

I would also like to thank the members of my examination committee, Prof. Ashish Khisti, Prof. Ben Liang, Prof. Wei Yu, and Prof. Krishna Narayanan from the Texas A&M University, for their useful suggestions to improve this thesis.

I had the great pleasure of sharing my journey with the past and present members of the FRK group and the iQua group, including Ben Smith, Mansoor Yousefi, Siyu Liu, Chunpo Pan, Chu Pang, Siddarth Hari, Christopher Blake, Lei Zhang, Da Wang, Weifei Zeng, Christian Senger, Chuan Wu, Hassan Shojania, Yunfeng Lin, Xinyu Zhang, Di Niu, Jin Jin, Henry Xu, Zimu Liu, Jiahua Wu, Elias Kehdi, Yuan Feng, Wei Wang, Yiwei Pu, Yuefei Zhu, Jun Li, Li Chen, Liyao Xiang, Younan Wang, and

Heng Xu.

I would also like to thank my friends from the communications group: Lei Zhou, Hayssam Dahrouj, Gokul Sridharan, Yuhan Zhou, Yicheng Lin, Pratik Patil, Binbin Dai, Lei Hua, Guang Ji, Qiang Xiao, Sun Sun, Weiwei Li, Lilin Zhang, and Yongbo Tang. My life at the University of Toronto would definitely not have been the same without all of you.

Last but not least, I would like to express my earnest gratitude to my parents for their tremendous encouragement, love, and support, and especially to my loving wife Junqi Yu for being so understanding, caring, and supportive.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Wireless networks play an increasingly important role in our lives. A recent study shows that wireless data traffic will increase by multiple orders of magnitude over the next few years [1]. This rapidly growing demand mainly comes from the proliferation of mobile devices (e.g., smart-phones, tablets, and laptops) and from users' desire for real-time data services (e.g., social networking, video streaming, and online gaming).

However, wireless radio spectrum is a limited natural resource. Hence, it is a big challenge for wireless industry to keep up with the fast-growing demand. This problem, referred to as the "wireless data crunch," urges the development of new communication techniques to improve resource utilization. One such technique is called physical-layer network coding (PNC).

1.1 Physical-Layer Network Coding

PNC is inspired by the principle of network coding [2], a theoretical breakthrough that fundamentally changes how network resources are utilized. The key idea behind network coding is to let relay nodes in a (wired) network forward *coded packets* instead of routing. Surprisingly, this idea not only achieves better network throughput in general, but also provides the best possible throughput for certain network scenarios [3, 4].

PNC pushes this idea even further to wireless networks. In particular, it exploits the broadcast and superposition nature of the wireless medium. The basic idea of PNC appears to have been independently proposed by several research groups in 2006: Zhang, Liew, and Lam [5], Popovski and Yomo [6], and Nazer and Gastpar [7]. To explain this idea, we introduce the so-called *two-way relay channel* in which two wireless nodes want to communicate via a relay as depicted in Fig. 1.1.

Figure 1.1: Two-way relay channel.

Consider the scenario where Alice and Bob want to exchange their packets. But they are far away from each other, and so they need a relay to help. The conventional method requires four time slots to accomplish this goal. In the first slot, Alice sends her packet $\mathbf{w}_1$ to the relay. In the second slot, Bob sends his packet $\mathbf{w}_2$ to the relay. In the third slot, the relay sends the packet $\mathbf{w}_1$ to Bob. In the last slot, the relay sends the packet $\mathbf{w}_2$ to Alice. Clearly, the throughput is 1/2 packet per slot, since four slots are needed to exchange two packets.

We can achieve better throughput by using the broadcast nature of the wireless medium. In the first slot, Alice sends her packet $\mathbf{w}_1$ to the relay. In the second slot, Bob sends his packet $\mathbf{w}_2$ to the relay. In the third slot, the relay computes the sum of the packets $\mathbf{w}_1 + \mathbf{w}_2$ (over some finite field) and broadcasts this coded packet to both Alice and Bob. With $\mathbf{w}_1 + \mathbf{w}_2$, Alice is able to recover Bob's packet $\mathbf{w}_2$, because she knows $\mathbf{w}_1$. Similarly, Bob can recover $\mathbf{w}_1$. This scheme requires three time slots to exchange two packets, achieving a throughput of 2/3 packet per slot.

We can do even better by exploiting the superposition nature of the wireless medium as well. In the first slot, Alice and Bob transmit their packets *simultaneously*, and the relay tries to infer $\mathbf{w}_1 + \mathbf{w}_2$ directly from the superposition of the transmitted signals. In the second slot, the relay broadcasts $\mathbf{w}_1 + \mathbf{w}_2$ to both Alice and Bob. As discussed above, Alice and Bob can obtain each other's packet with $\mathbf{w}_1 + \mathbf{w}_2$. Compared to the conventional approach, this new scheme reduces the required time slots from four to two, thereby doubling the throughput.

Due to its simplicity and potential to improve network throughput, PNC has received much research attention since 2006. A large number of strategies for PNC have been proposed, with a particular focus on two-way relay channels. A survey of PNC for two-way relay channels can be found in [8]. More recent surveys are in [9, 10].

Despite its popularity, PNC suffers from *fading*—another nature of the wireless medium. Roughly speaking, fading means that the transmitted signals will get distorted over the wireless channel. The

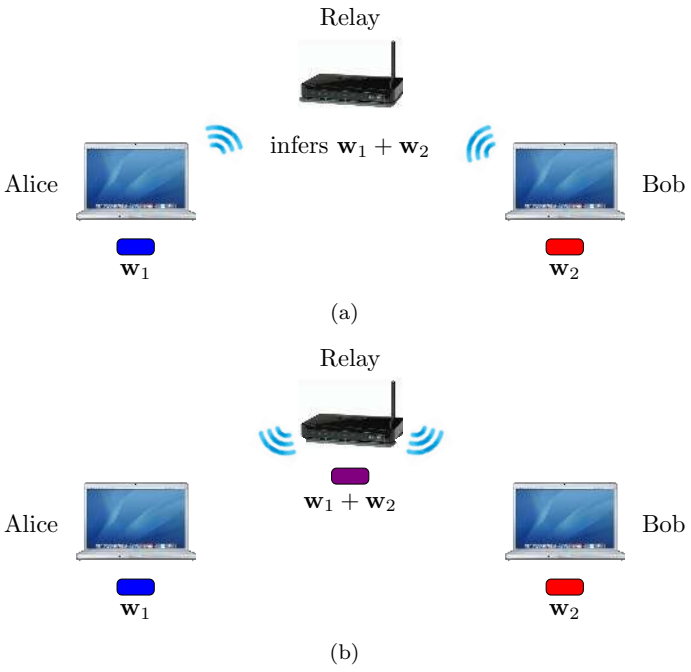


Figure 1.2: A PNC strategy for the two-way relay channel that requires two time slots.

channel distortion is often modeled as multiplication by a complex number[1], as illustrated in Fig. 1.3. For example, the received signal at the relay can be expressed as

$$\mathbf{y} = h_1\mathbf{x}_1 + h_2\mathbf{x}_2 + \mathbf{z}.$$

Here, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the transmitted signals from Alice and Bob, respectively (where $\mathbf{x}_i$ is a complex-valued vector depending on $\mathbf{w}_i$[2]), $\mathbf{z}$ is the channel noise, and $h_1, h_2 \in \mathbb{C}$ are called *channel gains*. Intuitively, when the channel gains h_1 and h_2 are quite different, it will be very difficult, if not impossible, for the relay to infer $\mathbf{w}_1 + \mathbf{w}_2$ correctly. This poses a unique challenge to PNC. One possible solution is called compute-and-forward (C&F).

1.2 Compute-and-Forward

C&F [11] mitigates the effect of fading by smartly introducing coefficients in front of $\mathbf{w}_1$ and $\mathbf{w}_2$. The relay no longer just infers $\mathbf{w}_1 + \mathbf{w}_2$. It has more choices: it can infer a linear combination $a_1\mathbf{w}_1 + a_2\mathbf{w}_2$. Intuitively, if the coefficients a_1 and a_2 are decided based on the channel gains h_1 and h_2, it would be much easier for the relay to infer this linear combination correctly. This will be made clear in Chapter 3.

[1] Such a model assumes narrow-band communication so that fading is frequency non-selective.

[2] In fact, there is a one-to-one correspondence between the vector $\mathbf{x}_i$ and the actual transmitted waveform, which can be found in standard textbooks on communications.

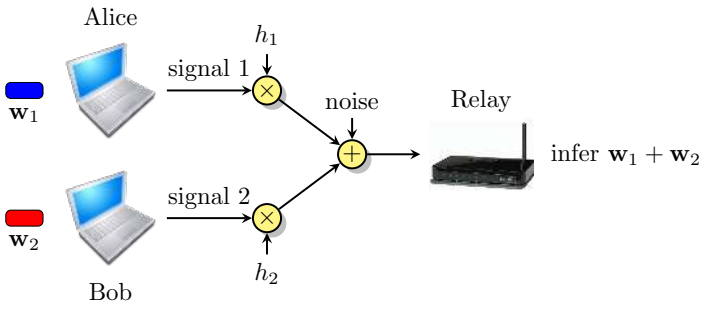


Figure 1.3: Illustration of the effect of fading on PNC.

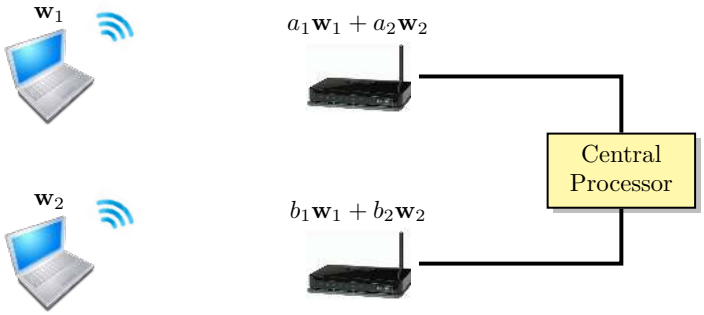


Figure 1.4: A wireless scenario where C&F is beneficial.

It turns out that the shift from $\mathbf{w}_1 + \mathbf{w}_2$ to $a_1\mathbf{w}_1 + a_2\mathbf{w}_2$ has many other advantages. For instance, it allows us to handle more wireless scenarios than the two-way relay channel. A particular example is illustrated in Fig. 1.4, where two relays are connected to a central processor through digital links (such as DSL or fiber connections). With C&F, each relay infers a linear combination and forwards it to the central processor. Since channel gains are often different for different relays, these two linear combinations are very likely to be linearly independent. Thus, the central processor can recover $\mathbf{w}_1$ and $\mathbf{w}_2$ by solving a system of linear equations. The benefit of C&F in such a scenario will be discussed in Chapter 6.

Existing work on C&F mainly focuses on analyzing its asymptotic performance using information-theoretic tools. For example, Narayanan, Wilson, and Sprintson proved that C&F is able to approach the cut-set bound for the two-way relay channel [12, 13]. Subsequently, Nam, Chung, and Lee extended their analysis to the case of unequal (but known) channel gains [14, 15]. Further work along this line can be found in [16, 17].

In their seminal work [11], Nazer and Gastpar showed that C&F outperforms other strategies (such as compress-and-forward, amplify-and-forward, and decode-and-forward) for a class of wireless relay networks in moderate signal-to-noise ratio (SNR) regimes, even if the transmitters lack channel state

information (CSI). If CSI is available at the transmitters, the performance of C&F can be greatly improved, achieving the full degrees of freedom for single-hop relay networks [18] and providing constant-gap-to-capacity result for multi-hop relay networks (without noise accumulation) [19]. C&F can be further enhanced when wireless relays are equipped with multiple antennas [20].

In addition to wireless relay networks, C&F achieves competitive performance in many other practically-relevant wireless networks (such as distributed antenna systems and small-cell networks) compared to other information-theoretic schemes [21]. Also, C&F gives new rate regions for symmetric interference channels [22] and for many-to-one interference channels [23]. The benefits of using C&F in network information theory have been summarized in Table 1.1. A recent survey of C&F can be found in [9].

Very recently, there are some other promising information-theoretic schemes proposed for wireless relay networks, including quantize-map-and-forward [24] and noisy network coding [25]. However, both of them require joint decoding at the final destination, which greatly complicates the implementation especially for large networks. In sharp contrast, the final destination in C&F only needs to solve a system of linear equations, making C&F scale well for large networks.

Table 1.1: Advantages of C&F in network information theory.

scenario	advantage	reference
two-way relay channel	better rates at high SNR	[13]
many-to-one interference channel	new rate regions	[23]
symmetric interference channel	new rate regions	[22]
single-hop relay network	full degrees of freedom	[18]
multi-hop relay network	constant gap to capacity	[19]

1.3 Motivation of This Work

Despite its great potential, prior work on C&F tends to ignore some practical constraints, as it mostly focuses on the asymptotic performance. For example, in C&F theory, the block length usually needs to go to infinity. But in reality, the block length is constrained by several factors, such as the coherence time and the delay requirement. In C&F theory, the complexity is often unbounded. In practice, the complexity should be well controlled. In C&F theory, each relay is assumed to know the perfect CSI. In reality, each relay only has imperfect CSI (due to channel estimation errors) or even no CSI. In C&F theory, the relays are always reliable. In practice, they might make decoding errors. All of these gaps between C&F theory and wireless practice have been summarized in Table 1.2, which motivate our work.

Table 1.2: Summary of four important gaps between C&F theory and wireless practice.

compute-and-forward	theory	practice
block length	very long	constrained
complexity	unbounded	bounded
channel state information	perfect	imperfect
relays	reliable	unreliable

1.4 Our Contributions

In this dissertation, we develop several new theories for C&F that take into account the practical constraints listed in Table 1.2. The specific contributions of this dissertation are described below.

1.4.1 An Algebraic Framework

The original analysis of C&F [11] is based on the existence of an (infinite) sequence of "asymptotically-good" nested lattice codes described in [26]. This sequence of lattice codes—originally constructed by Erez and Zamir to approach the capacity of additive white Gaussian noise (AWGN) channels—requires very long block length and almost unbounded complexity. To control the block length and complexity, perhaps an easy way is to focus on a special class of practical lattice codes and study the performance of C&F schemes built from these codes.

This dissertation takes a different approach. Rather than focusing on a particular class of nested lattice codes, it addresses two fundamental questions. First, which nested lattice codes are compatible with C&F? Second, what are the design criteria for C&F? The answers to these questions would provide a systematic understanding of C&F schemes with controlled block length and complexity. For the first question, we develop an algebraic framework that establishes a direct connection between C&F and module theory. This connection allows us to construct nested lattice codes for C&F in a systematic way. In particular, a generic C&F scheme is presented that makes no assumptions on the underlying nested lattice code. Based on this generic scheme, several generalized constructions of C&F schemes are given, which enable us to systematically control the block length and complexity. For the second question, we derive the design criteria for hypercube-shaped C&F schemes. Following our design criteria, several exemplary C&F schemes are provided, showing that nominal coding gains of 3 to 7.5 dB can be obtained with relatively short block length and reasonable decoding complexity.

1.4.2 Blind Compute-and-Forward

Conventional C&F schemes require CSI at the receivers so that an "optimal" scaling factor can be computed for the purposes of decoding. In this dissertation, we aim to eliminate the need for CSI in C&F. This is motivated by the fact that C&F is sensitive to channel estimation error [27] and the fact that the requirement of accurate CSI is quite demanding when the number of concurrent transmissions is large [28].

This dissertation proposes a blind C&F scheme that does not require the knowledge of CSI. Rather than attempting to compute the optimal scaling factor as conventional C&F does, our new scheme seeks one or more "good" scalars, i.e., scalars which allow correct decoding despite possibly being sub-optimal. To find a good scalar, a computationally efficient scheme is proposed, which involves three key components: error-detection, a hierarchically organized list, as well as a use of the Smoothing Lemma from lattice theory. This scheme is able to achieve almost the same throughput as coherent C&F (its CSI-enabled counterpart) with a modest increase in computational complexity. Furthermore, our simulation results show that, for a certain class of nested lattice codes, this new scheme has roughly twice the complexity of coherent C&F in the high-throughput region.

1.4.3 End-to-End Error Control

The asymptotic analysis of C&F assumes *vanishing* probability of decoding errors at relays. In reality, such error probability is *non-vanishing*, which means that relays could be unreliable. This gives rise to the issue of error propagation. To see this, let us consider a two-hop wireless relay network as depicted in Fig. 1.5. With C&F, in the first time slot, two transmitters on the left are sending their packets $\mathbf{w}_1$ and $\mathbf{w}_2$ simultaneously, and each relay on the middle attempts to decode a linear combination. Similarly, in the second slot, two relays on the middle are sending their packets, and each relay on the right tries to decode a linear combination and then forwards it to the central processor. Now, suppose that in the first slot, the relay on the bottom makes a decoding error. Clearly, this error would quickly propagate to the relays on the right, making the central processor incapable of recovering the original packets $\mathbf{w}_1$ and $\mathbf{w}_2$.

This dissertation models such error propagation as a matrix channel $Y = AX + BE$, where X is the channel input, Y is the channel output observed by the central processor, E is random error introduced by unreliable relays, and A and B are random transfer matrices. For instance, the matrix channel

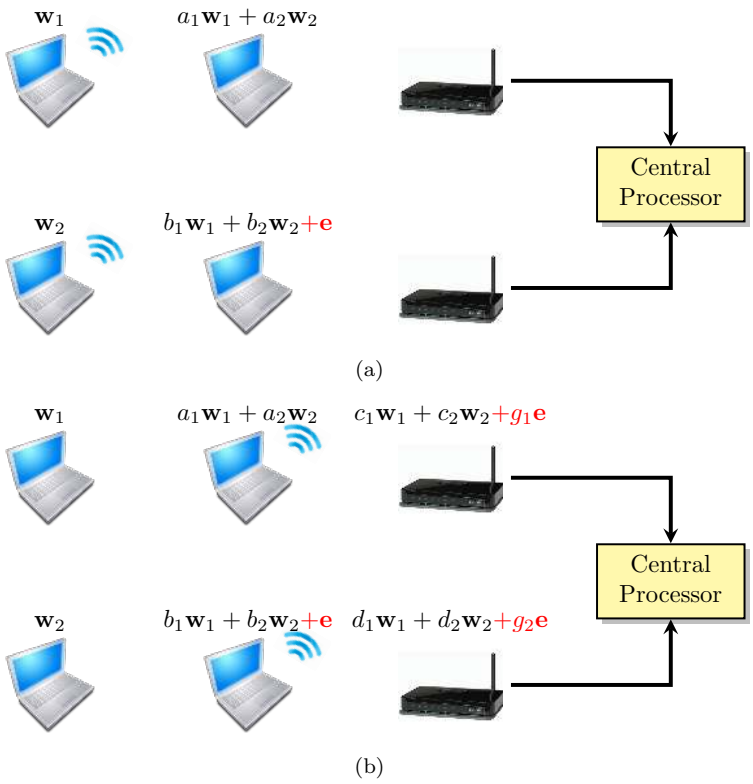


Figure 1.5: Illustration of error propagation in C&F.

associated with Fig. 1.5 can be expressed as

$$Y = \begin{bmatrix} c_1 & c_2 \\ d_1 & d_2 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \mathbf{e}.$$

The main contribution of this part includes tight capacity bounds and polynomial-complexity capacity-achieving coding schemes under certain distributions of A, B, and E.

1.4.4 Application to Random-Access Wireless Networks

Prior work on C&F often focuses on simple network configurations for the sake of clean theoretical results. In particular, a number of key aspects of real-world networks have been ignored. For example, prior work on C&F assumed that data is always available at the wireless nodes. In practice, data traffic is often bursty. Prior work on C&F assumed that the network operation is centralized. In reality, many wireless networks are decentralized. In addition, in many networking applications, delay is a primary concern, which has not been analyzed in prior work.

As a starting point to capture these aspects, this dissertation studies the use of C&F in slotted-

ALOHA protocols (a family of random-access protocols that embrace bursty traffic and decentralized operations), with a particular focus on its throughput and delay performance. It turns out that this problem is closely related to the stability condition of slotted ALOHA with multi-packet reception, which is, however, largely open except for two special cases. To address this difficulty, we first propose an *approximate* stability condition, which not only recovers existing special cases, but also is provably *exact* when the number of users grows large. Furthermore, we show that this stability condition is very accurate even for small systems. Finally, using this stability condition, we characterize the throughput and delay performance of slotted-ALOHA with C&F, demonstrating its clear advantages over standard slotted-ALOHA systems.

Chapter 2

Mathematical Preliminaries

In this chapter, we present some mathematical preliminaries that will be used throughout the thesis.

2.1 Rings, Modules, and Matrices

In this section, we present some basic results for finite chain rings, and modules and matrices over finite chain rings. This section establishes notation and the results that will be used later; nevertheless, this material is standard; see e.g., [29–35] for more details.

2.1.1 Rings and Ideals

We begin with some common definitions and notations for rings. All rings in this thesis will be commutative with identity $1 \neq 0$. Let R be a ring. We will let R^* denote the nonzero elements of R, i.e., $R^* = R \setminus \{0\}$. An element a in R is called a *unit* if $ab = 1$ for some $b \in R$. We will let $U(R)$ denote the units in R. Two elements $a, b \in R$ are said to be *associates* if $a = ub$ for some $u \in U(R)$. Associatedness is an equivalence relation on R.

Suppose $a, b \in R$. The element a *divides* b, written $a \mid b$, if $ac = b$ for some $c \in R$. Let $d \in R^*$ be a nonzero element in R. Two elements a, b are said to be *congruent modulo* d if d divides $a - b$. Congruence modulo d is an equivalence relation on R. A set containing exactly one element from each equivalence class is called a *complete set of residues* with respect to d, and is denoted by $\mathcal{R}(R, d)$. Note that the difference $a - b$ between distinct elements $a, b \in \mathcal{R}(R, d)$, $a \neq b$, can never be a multiple of d.

A non-unit element $p \in R$ is called a *prime* of R if whenever $p \mid ab$ for some elements a and b in R, then either $p \mid a$ or $p \mid b$. An element a of R^* is a called a *zero-divisor* if $ab = 0$ for some $b \in R^*$. If R contains no zero-divisors, then R is an *integral domain*. If R is finite and an integral domain, then R

is, in fact, a finite field. This latter case is not of central interest in this thesis; almost all of the rings considered here will have zero divisors.

Example 2.1. *Let $R = \mathbb{Z}_8 \triangleq \{0, \ldots, 7\}$, under integer addition and multiplication modulo 8. Then $U(\mathbb{Z}_8) = \{1, 3, 5, 7\}$. There are four equivalent classes induced by congruence modulo 4, namely, $\{0, 4\}$, $\{1, 5\}$, $\{2, 6\}$, and $\{3, 7\}$. An example of a complete set of residues with respect to the element 4 in $\mathcal{R}(\mathbb{Z}_8)$ is $\mathcal{R}(\mathbb{Z}_8, 4) = \{0, 1, 2, 3\}$. The zero-divisors of $\mathbb{Z}_8$ form the set $\{2, 4, 6\}$. There is no prime of $\mathbb{Z}_8$.*

A nonempty subset I of R that is closed under subtraction, i.e., $a, b \in I$ implies $a - b \in I$, and closed under inside-outside multiplication, i.e., $a \in I$ and $r \in R$ implies $ar \in I$, is called an *ideal* of R. If $A = \{a_1, \ldots, a_m\}$ is a finite nonempty subset of R, we will use $\langle a_1, \ldots, a_m \rangle$ to denote the ideal generated by A, i.e.,

$$\langle a_1, \ldots, a_m \rangle = \{a_1 c_1 + \cdots + a_m c_m \colon c_1, \ldots, c_m \in R\}.$$

An ideal I of R is said to be *principal* if I is generated by a single element in I, i.e., $I = \langle d \rangle$ for some $d \in I$. A ring R is called a *principal ideal ring* (PIR) if every ideal I of R is principal.

Let I be an ideal of R. Two elements a and b are said to be *congruent modulo I* if $a - b \in I$. In particular, if $I = \langle d \rangle$ is principal, then congruence modulo I is the same as congruence modulo d. Congruence modulo I is an equivalence relation whose equivalence classes are (additive) cosets $a + I$ of I in R. The *quotient ring* of R by I, denoted R/I, is the ring obtained by defining addition and multiplication operations on the cosets of I in R in the usual way, as

$$(a + I) + (b + I) = (a + b) + I \text{ and } (a + I) \times (b + I) = (ab) + I.$$

An ideal N is said to be *maximal* if $N \neq R$ and the only ideals containing N are N and R (in other words, N is "maximal" with respect to set inclusion among all proper ideals). If N is a maximal ideal, then the quotient ring R/N is a field, called a *residue field*. A ring with a unique maximal ideal is said to be *local*.

Example 2.2. *The ideals of $\mathbb{Z}_8$ are $\{0\} = \langle 0 \rangle$, $\{0, 4\} = \langle 4 \rangle$, $\{0, 2, 4, 6\} = \langle 2 \rangle$, and $R = \langle 1 \rangle$. Thus, $\mathbb{Z}_8$ is a PIR, and has a unique maximal ideal $\langle 2 \rangle$. The residue field $\mathbb{Z}_8/\langle 2 \rangle$ is isomorphic to the finite field $\mathbb{F}_2$ of two elements.*

2.1.2 Principal Ideal Domains

An integral domain in which every ideal is principal is called a *principal ideal domain* (PID). Clearly, a PID is a special case of a PIR. Typical examples of a PID include the integers $\mathbb{Z}$, the Gaussian

integers $\mathbb{Z}[i]$ and the Eisenstein integers $\mathbb{Z}[\omega]$, where $\omega = e^{2\pi i/3}$. Formally, Gaussian integers are the set $\mathbb{Z}[i] \triangleq \{a + bi : a, b \in \mathbb{Z}\}$, and Eisenstein integers are the set $\mathbb{Z}[\omega] \triangleq \{a + b\omega : a, b \in \mathbb{Z}\}$.

The Gaussian integers $\mathbb{Z}[i]$ have four units $(\pm 1, \pm i)$. A Gaussian integer is called a *Gaussian prime* if it is a prime in $\mathbb{Z}[i]$. A Gaussian integer $a + bi$ is a Gaussian prime if and only if it satisfies exactly one of the following:

1. $|a| = |b| = 1$;
2. one of $|a|, |b|$ is zero and the other is a prime number in $\mathbb{Z}$ of the form $4j + 3$ (with j a nonnegative integer);
3. both of $|a|, |b|$ are nonzero and $a^2 + b^2$ is a prime number in $\mathbb{Z}$ of the form $4j + 1$.

Note that these properties are symmetric with respect to $|a|$ and $|b|$. Thus, if $a + bi$ is a Gaussian prime, so are $\{\pm a \pm bi\}$ and $\{\pm b \pm ai\}$.

The Eisenstein integers $\mathbb{Z}[\omega]$ have six units $(\pm 1, \pm\omega, \pm\omega^2)$. An Eisenstein integer is called an *Eisenstein prime* if it is a prime in $\mathbb{Z}[\omega]$. An Eisenstein integer $a + b\omega$ is an Eisenstein prime if and only if it satisfies exactly one of the following:

1. $a + b\omega$ is a product of a unit in $\mathbb{Z}[\omega]$ and a prime number in $\mathbb{Z}$ of the form $3j + 2$;
2. $|a + b\omega|^2 = a^2 - ab + b^2$ is a prime number in $\mathbb{Z}$.

Let T be a PID and let $d \in T$. Then it is known that the quotient ring $T/\langle d \rangle$ is a PIR [31].

2.1.3 Finite Chain Rings

A ring R is called a *chain ring* if the ideals of R satisfy a containment condition: for any two ideals I, J of R, either $I \subseteq J$ or $J \subseteq I$. If R is a chain ring with finitely many elements, then R is called a *finite chain ring*. Clearly, a finite chain ring has a unique maximal ideal, and hence is local. It is known [29] that a finite ring is a chain ring if and only if it is a local PIR; thus, in a finite chain ring, all ideals are principal. Examples of finite chain rings include $\mathbb{Z}_{p^n}$ (the ring of integers modulo p^n where p is a prime) and Galois rings.

Let R be a finite chain ring, and let $\pi \in R$ be any generator of the maximal ideal of R. Then $R/\langle \pi \rangle$ is the residue field of R. It can be shown (see, e.g., [29]) that every ideal I of R, including the zero ideal $\langle 0 \rangle$, is generated by a power of π, i.e., $I = \langle \pi^l \rangle$ for some $l \geq 0$. It follows that π is nilpotent; we denote by s the *nilpotency index* of π, i.e., the smallest positive integer such that $\pi^s = 0$. There are,

then, exactly $s+1$ distinct ideals of R, namely, $R = \langle \pi^0 \rangle, \langle \pi^1 \rangle, \ldots, \langle \pi^s \rangle = \{0\}$ which form a chain (with respect to set inclusion):

$$R = \langle \pi^0 \rangle \supset \langle \pi^1 \rangle \supset \cdots \supset \langle \pi^{s-1} \rangle \supset \langle \pi^s \rangle = \{0\}.$$

Thus, s is often called the *chain length* of R. We refer to R as a (q, s) chain ring, if R has a residue field of size q and a chain length of s.

Example 2.3. *The ideals of $\mathbb{Z}_8$ form a chain with respect to set inclusion:*

$$R = \langle 1 \rangle \supset \langle 2 \rangle \supset \langle 4 \rangle \supset \langle 0 \rangle = \{0\}.$$

Thus, $\mathbb{Z}_8$ is a finite chain ring with chain length $s = 3$. Since the residue field $\mathbb{Z}_8/\langle 2 \rangle$ is isomorphic to $\mathbb{F}_2$, $\mathbb{Z}_8$ is a $(2, 3)$ chain ring.

Now let $\mathcal{R}(R, \pi) \subseteq R$ be a complete set of residues with respect to π and, without loss of generality, assume that $0 \in \mathcal{R}(R, \pi)$. Every element $a \in R$ then has a unique representation, called the *π-adic decomposition* of a (with respect to $\mathcal{R}(R, \pi)$), in the form

$$a = a_0 + a_1\pi + \cdots + a_{s-1}\pi^{s-1}, \tag{2.1}$$

where $a_0, \ldots, a_{s-1} \in \mathcal{R}(R, \pi)$. It follows from the uniqueness of (2.1) that the size of R is q^s, i.e., the number of elements in a (q, s) chain ring is q^s. Thus, like a finite field, a finite chain ring has a cardinality that is an integer power of a prime number.

The *degree* of a nonzero element $a_0 + a_1\pi + \cdots + a_{s-1}\pi^{s-1} \in R^*$, denoted by $\deg(a)$, is defined as the *least* index j for which $a_j \neq 0$. By convention, the degree of 0 is defined as s. All elements of the same degree are associates in R. Further, a divides b if and only if $\deg(a) \leq \deg(b)$. Finally, $\deg(a + b) \geq \min\{\deg(a), \deg(b)\}$, i.e., adding two elements cannot result in an element of lower degree.

Example 2.4. *Let $\mathcal{R}(\mathbb{Z}_8, 2) = \{0, 1\}$. The 2-adic decomposition of $5 \in \mathbb{Z}_8$ is $5 = 1 + 0 \cdot 2 + 1 \cdot 2^2$. The elements in $\mathbb{Z}_8$ of degree 0 (respectively, 1, 2, and 3) are $\{1, 3, 5, 7\}$ (respectively, $\{2, 6\}$, $\{4\}$, and $\{0\}$).*

Finally, we present two methods for constructing finite chain rings.

If R is itself a (q, s) chain ring with maximal ideal $\langle \pi \rangle$, then the quotient ring $R/\langle \pi^l \rangle$ $(0 < l < s)$ is a (q, l) chain ring. This method constructs new finite chain rings from existing ones.

If T is a PID, and p is a prime in T, then $T/\langle p \rangle$ is a field, since $\langle p \rangle$ is a maximal ideal of T. Let q

be the size of $T/\langle p\rangle$ and suppose that q is finite. Then the quotient ring $T/\langle p^l\rangle$ is a (q,l) $(l>0)$ chain ring. This method constructs finite chain rings from PIDs.

2.1.4 Modules

Modules are to rings as vector spaces are to fields. Formally, let R be a commutative ring with identity $1 \neq 0$. An R-module is a set M together with 1) a binary operation $+$ on M under which M is an abelian group, and 2) an action of R on M which satisfies the same axioms as those for vector spaces.

An R-submodule of M is a subset of M which itself forms an R-module. Let N be a submodule of M. The quotient group M/N can be made into an R-module by defining an action of R satisfying, for all $r \in R$, and all $x+N \in M/N$, $r(x+N) = (rx)+N$. Hence, M/N is often referred to as a *quotient R-module*.

Let M and N be R-modules. A map $\varphi : M \to N$ is called an *R-module homomorphism* if the map φ satisfies

1. $\varphi(x+y) = \varphi(x)+\varphi(y)$, for all $x, y \in M$ and
2. $\varphi(rx) = r\varphi(x)$, for all $r \in R, x \in M$.

The *kernel of* φ is defined as $\ker\varphi \triangleq \{m \in M : \varphi(m) = 0\}$. Clearly, $\ker\varphi$ is a submodule of M.

An R-module homomorphism $\varphi : M \to N$ is called an *R-module isomorphism* if it is both injective and surjective. In this case, the modules M and N are said to be *isomorphic*, denoted by $M \cong N$. An R-module M is called a *free* module of *rank* t if $M \cong R^t$ for some nonnegative integer t.

There are several isomorphism theorems for modules. The so-called "first isomorphism theorem" is useful for this thesis.

Theorem 2.1 (First Isomorphism Theorem for Modules [34, p. 349])**.** *Let M, N be R-modules and let $\varphi : M \to N$ be an R-module homomorphism. Then $\ker\varphi$ is a submodule of M and $M/\ker\varphi \cong \varphi(M)$.*

2.1.5 Modules over PIDs

Finitely-generated modules over PIDs play an important role in this thesis, and are defined as follows.

Definition 2.1 (Finitely-Generated Modules)**.** *Let R be a commutative ring with identity $1 \neq 0$ and let M be an R-module. For any subset A of M, let $\langle A\rangle$ be the smallest submodule of M containing A, called the* submodule generated by A. *If $M = \langle A\rangle$ for some finite subset A, then M is said to be* finitely generated.

A finite module (i.e., a module that contains finitely many elements) is always finitely generated, but a finitely-generated module is not necessarily finite. For example, the even integers $2\mathbb{Z}$ form a $\mathbb{Z}$-module generated by $\{2\}$.

The following structure theorem says that, if T is a PID, then a finitely-generated T-module is isomorphic to a finite direct product of T-modules of the form T or $T/\langle d\rangle$.

Theorem 2.2 (Structure Theorem for Finitely-Generated Modules over a PID—Invariant Factor Form [34, p. 462])**.** *Let T be a PID and let M be a finitely-generated T-module. Then for some integer $t \geq 0$ and nonzero non-unit elements $d_1, \ldots, d_k$ of T satisfying the divisibility relations $d_1 \mid d_2 \mid \cdots \mid d_k$,*

$$M \cong T^t \times T/\langle d_1\rangle \times T/\langle d_2\rangle \times \cdots \times T/\langle d_k\rangle.$$

The elements $d_1, \ldots, d_k$, called the invariant factors *of M, are unique up to multiplication by units in T. The integer t is called the* free rank *of M.*

2.1.6 Modules over Finite Chain Rings

When R is a finite chain ring, an R-module is always isomorphic to a direct product of various ideals of R; this structure can be described by a "shape." An *s-shape* $\mu = (\mu_1, \mu_2, \ldots, \mu_s)$ is simply a sequence of non-decreasing non-negative integers, i.e., $0 \leq \mu_1 \leq \mu_2 \leq \cdots \leq \mu_s$. We denote by $|\mu|$ the sum of its components, i.e., $|\mu| = \sum_{i=1}^{s} \mu_i$. For later notational convenience, we define the "zeroth component" of a shape as $\mu_0 = 0$.

An s-shape $\kappa = (\kappa_1, \ldots, \kappa_s)$ is said to be a *subshape* of $\mu = (\mu_1, \ldots, \mu_s)$, written $\kappa \preceq \mu$, if $\kappa_i \leq \mu_i$ for all $i = 1, \ldots, s$. Thus, for example, $(1, 1, 3) \preceq (2, 4, 4)$. The number of subshapes of the s-shape $(m, \ldots, m)$ is given by $\binom{m+s}{s}$, which implies that the number of subshapes of $\mu = (\mu_1, \ldots, \mu_s)$ is upper-bounded by $\binom{\mu_s+s}{s}$.

Two s-shapes can be added together to form a new s-shape simply by adding componentwise. Thus, for example, $(1, 1, 3) + (2, 4, 4) = (3, 5, 7)$. Also, for a shape $\mu = (\mu_1, \ldots, \mu_s)$ and a positive integer m we define $\mu/m = (\mu_1/m, \ldots, \mu_s/m)$ (which is an s-tuple, but not necessarily a shape). For convenience, we will sometimes identify the integer t with the s-shape $(t, \ldots, t)$. Thus, for example, $\mu \preceq t$ means $\mu_i \leq t$ for all i, $\kappa = t$ means $\kappa_i = t$ for all i, and $\mu - t = (\mu_1 - t, \ldots, \mu_s - t)$, assuming $t \preceq \mu$.

Let R be a (q, s) chain ring with maximal ideal $\langle\pi\rangle$. For any s-shape μ, we define the R-module R^μ as

$$R^\mu \triangleq \underbrace{\langle 1\rangle \times \cdots \times \langle 1\rangle}_{\mu_1} \times \underbrace{\langle\pi\rangle \times \cdots \times \langle\pi\rangle}_{\mu_2-\mu_1} \times \cdots \times \underbrace{\langle\pi^{s-1}\rangle \times \cdots \times \langle\pi^{s-1}\rangle}_{\mu_s-\mu_{s-1}}. \tag{2.2}$$

Since a positive integer t is identified with the shape $(t, \ldots, t)$, it is indeed true that R^t denotes the t-fold Cartesian product of R with itself.

The module R^μ can be viewed as a collection of μ_s-tuples whose components are drawn from R subject to certain constraints imposed by μ. Specifically, while the first μ_1 components can be any element of R, the next $\mu_2 - \mu_1$ components must be multiples of π, and so on. Since each ideal $\langle \pi^i \rangle$ in (2.2) contains q^{s-i} elements ($0 \leq i < s$), it follows that the size of R^μ is $|R^\mu| = q^{|\mu|}$.

Example 2.5. *Let $R = \mathbb{Z}_8$, and let $\mu = (2, 4, 4)$. Then*

$$R^\mu = \underbrace{\langle 1 \rangle \times \langle 1 \rangle}_{2} \times \underbrace{\langle 2 \rangle \times \langle 2 \rangle}_{4-2}.$$

Note that the first two components of R^μ can each be chosen in 2^3 ways, while the last two components can each be chosen in only 2^2 ways. Hence, the size of R^μ is 2^{10}.

For every s-shape μ, R^μ is a finite R-module. Conversely, the following theorem establishes that every finite R-module is isomorphic to R^μ for some unique s-shape μ.

Theorem 2.3. *[33, Theorem 2.2] For any finite R-module M over a (q, s) chain ring R, there is a unique s-shape μ such that $M \cong R^\mu$.*

We call the unique shape μ given in Theorem 2.3 *the shape of* M, and write $\mu = \operatorname{shape} M$.[1] It is known [33] that if M' is a submodule of M, then $\operatorname{shape} M' \preceq \operatorname{shape} M$, i.e., the shape of a submodule is a subshape of the module. It is also known [33] that the number of submodules of R^μ whose shape is κ is given by

$$\left[\!\!\begin{matrix}\mu \\ \kappa\end{matrix}\!\!\right]_q = \prod_{i=1}^{s} q^{(\mu_i - \kappa_i)\kappa_{i-1}} \begin{bmatrix}\mu_i - \kappa_{i-1} \\ \kappa_i - \kappa_{i-1}\end{bmatrix}_q, \tag{2.3}$$

where

$$\begin{bmatrix} m \\ k \end{bmatrix}_q \triangleq \prod_{i=0}^{k-1} \frac{q^m - q^i}{q^k - q^i}$$

is the Gaussian coefficient. In particular, when the chain length $s = 1$, R becomes the finite field $\mathbb{F}_q$ of q elements, and $\left[\!\!\begin{smallmatrix}\mu \\ \kappa\end{smallmatrix}\!\!\right]_q$ becomes $\begin{bmatrix}\mu_1 \\ \kappa_1\end{bmatrix}_q$, which is the number of κ_1-dimensional subspaces of $\mathbb{F}_q^{\mu_1}$.

2.1.7 Matrices over Rings

We turn now to matrices over rings. Let $R^{n \times m}$ denote the set of all $n \times m$ matrices over R. For any matrix $A \in R^{n \times m}$, we denote by $A[i, j]$ the entry of A in the ith row and jth column, where $1 \leq i \leq n$

[1] Some authors (like Honold *et al.* [33]) use a different convention and define the shape of an R-module to be the conjugate (in the integer-partition-theoretic sense) of the shape as defined in this chapter.

and $1 \leq j \leq m$. We will let $A[i_1{:}i_2, j_1{:}j_2]$ denote the submatrix of A formed by rows i_1 to i_2 and by columns j_1 to j_2, where $1 \leq i_1 \leq i_2 \leq n$ and $1 \leq j_1 \leq j_2 \leq m$. Finally, we will let $A[i,:]$ denote the ith row of A and $A[:,j]$ denote the jth column A.

A square matrix $U \in R^{n\times n}$ is *invertible* if $UV = VU = I_n$ for some $V \in R^{n\times n}$, where I_n denotes the $n \times n$ identity matrix. The set of invertible matrices in $R^{n\times n}$, denoted as $\mathrm{GL}_n(R)$, forms a group—the so-called *general linear group*—under matrix multiplication. Two matrices $A, B \in R^{n\times m}$ are said to be *left-equivalent* if there exists a matrix $U \in \mathrm{GL}_n(R)$ such that $UA = B$. Two matrices $A, B \in R^{n\times m}$ are said to be *equivalent* if there exist matrices $U \in \mathrm{GL}_n(R)$ and $V \in \mathrm{GL}_m(R)$ such that $UAV = B$. We will write $A \approx B$ if A and B are equivalent.

A matrix $D \in R^{n\times m}$ is called a *diagonal matrix* if $D[i,j] = 0$ whenever $i \neq j$. Note that a diagonal matrix needs not be square. A diagonal matrix D can be written as $D = \mathrm{diag}(d_1, \ldots, d_r)$, where $r = \min\{n, m\}$, and $d_i = D[i,i]$ for $i = 1, \ldots, r$. Let $A \in R^{n\times m}$. A diagonal matrix $D = \mathrm{diag}(d_1, \ldots, d_r) \in R^{n\times m}$ ($r = \min\{n, m\}$) is called a *Smith normal form* of A, if $D \approx A$ and $d_1 \mid d_2 \mid \cdots \mid d_r$ in R.

Note that $d_1 \mid d_2 \mid \cdots \mid d_r$ in R if and only if $\langle d_1\rangle \supseteq \langle d_2\rangle \supseteq \cdots \supseteq \langle d_r\rangle$. In particular, if d_i is a unit in R, then $d_1, \ldots, d_i$ are all units in R. Similarly, if $d_i = 0$, then $d_i, \ldots, d_r$ are all 0. Thus, if $D = \mathrm{diag}(d_1, \ldots, d_r)$ is a Smith normal form of $\mathbf{A}$, then the diagonal entries $d_1, \ldots, d_r$ of D can be expressed as

$$d_1, \ldots, d_r = \underbrace{u_1, \ldots, u_i}_{i}, \underbrace{d_{i+1}, \ldots, d_{i+j}}_{j}, \underbrace{0, \ldots, 0}_{k}$$

where $u_1, \ldots, u_i$ are units in R, $d_{i+1}, \ldots, d_{i+j}$ are nonzero, non-unit elements in R, and $i, j, k \geq 0$ with $i + j + k = r$. The nonzero entries $\{u_1, \ldots, u_i, d_{i+1}, \ldots, d_{i+j}\}$ are called a *sequence of invariant factors* of A.

The Smith normal form theorem says that every matrix over a PIR has a Smith normal form whose sequence of invariant factors is unique up to equivalence of associates.

Theorem 2.4 (Smith Normal Form Theorem [31, p. 194])**.** *Let R be a PIR. Then any $A \in R^{m\times n}$ has a Smith normal form. Furthermore, if $D_1 = \mathrm{diag}(d_1, \ldots, d_r)$ and $D_2 = \mathrm{diag}(s_1, \ldots, s_r)$ are two Smith normal forms of A, then $\langle d_i\rangle = \langle s_i\rangle$ for all $i = 1, \ldots, r$.*

2.1.8 Matrices over Finite Chain Rings

For convenience, we shall require the diagonal entries $d_1, \ldots, d_r$ in the Smith normal form D to be powers of π, i.e.,

$$(d_1, \ldots, d_r) = (\pi^{l_1}, \ldots, \pi^{l_r}),$$

where $0 \le l_1 \le \ldots \le l_r \le s$. With this constraint, once π is fixed, every matrix over a finite chain ring has a unique Smith normal form.

Example 2.6. *Consider the two matrices*

$$A = \begin{bmatrix} 4 & 6 & 2 & 1 \\ 0 & 0 & 0 & 2 \\ 2 & 4 & 6 & 1 \\ 2 & 0 & 2 & 1 \end{bmatrix}, \; S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

over $\mathbb{Z}_8$. *It is easy to check that*

$$A = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 2 & 2 & 1 \\ 1 & 1 & 2 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = USV.$$

Since U *and* V *are invertible,* S *is equivalent to* A. *Since the diagonal entries of* S *satisfy* $1 \mid 2 \mid 4 \mid 0$ *in* $\mathbb{Z}_8$, S *is the Smith normal form of* A.

For any $A \in R^{n\times m}$, we denote by $\operatorname{row} A$ and $\operatorname{col} A$ the row span and column span of A, respectively. By using the Smith normal form, it is easy to see that the row span $\operatorname{row} A$ is isomorphic, as an R-module, to the column span $\operatorname{col} A$.

Note that two matrices $A, B \in R^{n\times m}$ are left-equivalent if and only if $\operatorname{row} A = \operatorname{row} B$, i.e., left-equivalent matrices have identical row spans. On the other hand, two matrices $A, B \in R^{n\times m}$ are equivalent if and only if $\operatorname{row} A \cong \operatorname{row} B$, i.e., equivalent matrices have isomorphic row spans.

The *shape* of a matrix A is defined as the shape of the row span of A, i.e.,

$$\operatorname{shape} A = \operatorname{shape}(\operatorname{row} A).$$

Clearly, $\operatorname{shape} A = \operatorname{shape}(\operatorname{col} A)$. Moreover, $\operatorname{shape} A = \mu$ if and only if the Smith normal form of A is given by

$$\operatorname{diag}(\underbrace{1,\ldots,1}_{\mu_1}, \underbrace{\pi,\ldots,\pi}_{\mu_2-\mu_1}, \ldots, \underbrace{\pi^{s-1},\ldots,\pi^{s-1}}_{\mu_s-\mu_{s-1}}, \underbrace{0,\ldots,0}_{r-\mu_s}),$$

where $r = \min\{n, m\}$. In particular, a matrix $U \in R^{n\times n}$ is invertible if and only if $\operatorname{shape} U = (n, \ldots, n)$.

Example 2.7. *Since* $D = \operatorname{diag}(1, 2, 4, 0)$ *is the Smith normal form of* A *in Example 2.6,* $\operatorname{shape} A =$

$(1, 2, 3)$.

As one might expect, matrix shape has a number of properties similar to matrix rank.

Proposition 2.1. *Let $A \in R^{n\times m}$ and $B \in R^{m\times k}$. Then*

1. $\operatorname{shape} A = \operatorname{shape} A^T$, *where A^T is the transpose of A.*
2. *For any $P \in \mathrm{GL}_n(R), Q \in \mathrm{GL}_m(R)$,* $\operatorname{shape} A = \operatorname{shape} PAQ$.
3. $\operatorname{shape} AB \preceq \operatorname{shape} A$, $\operatorname{shape} AB \preceq \operatorname{shape} B$.
4. *For any submatrix C of A,* $\operatorname{shape} C \preceq \operatorname{shape} A$.

Proof. 1) Since $\operatorname{row} A \cong \operatorname{col} A$, we have $\operatorname{row} A \cong \operatorname{row} A^T$. Hence, $\operatorname{shape} A = \operatorname{shape} A^T$. 2) Since A is equivalent to PAQ for any invertible P and Q, $\operatorname{shape} A = \operatorname{shape} PAQ$. 3) Since $\operatorname{row} AB$ is a submodule of $\operatorname{row} B$, we have $\operatorname{shape} AB \preceq \operatorname{shape} B$. Similarly, since $\operatorname{col} AB$ is a submodule of $\operatorname{col} A$, we have $\operatorname{shape} AB \preceq \operatorname{shape} A$. 4) Finally, note that any submatrix C of A is equal to $E_1 A E_2$ for some $E_1 \in R^{k\times n}$ (selecting k rows) and $E_2 \in R^{m\times l}$ (selecting l columns). Hence, $\operatorname{shape} C = \operatorname{shape} E_1 A E_2 \preceq \operatorname{shape} A$. □

For convenience, we say a matrix $A \in R^{n\times m}$ have *rank* t, if $\operatorname{shape} A = t$. Note that the rank of a matrix is not always defined. A matrix $A \in R^{n\times m}$ is called *full rank* if $\operatorname{rank} A = \min\{n, m\}$. A matrix $A \in R^{n\times m}$ is called *full row rank* if $\operatorname{rank} A = n$ (which requires $n \leq m$). The number of full-row-rank matrices in $R^{n\times m}$ is $q^{snm} \prod_{i=0}^{n-1}(1 - q^{i-m})$. A matrix is *full column rank* if its transpose is full row rank. Full-column-rank matrices have the following property.

Lemma 2.1. *Let A be a full-column-rank matrix. Then AB is a zero matrix if and only if B is a zero matrix.*

Proof. The "if" part is trivial, so we turn to the "only if" part. Let $A \in R^{n\times m}$. Suppose that $AB = 0$ for some matrix $B \in R^{m\times k}$. We will show that B is a zero matrix. Since A is full column rank, its Smith normal form S must have the form

$$S = \begin{bmatrix} I_m \\ 0_{(n-m)\times m} \end{bmatrix}$$

and $A = USV$ for some invertible matrices U and V. Thus, we have

$$AB = U \begin{bmatrix} I \\ 0 \end{bmatrix} VB = 0,$$

which implies $B = 0$. □

2.2 Lattices and Nested Lattice Codes

Here, we introduce basic concepts and notation about lattices and nested lattice codes, mainly based on [36, 37].

2.2.1 Lattices

Recall that a real lattice $\Lambda \in \mathbb{R}^n$ is a regular array of points in $\mathbb{R}^n$. Algebraically, a real lattice is defined as a discrete $\mathbb{Z}$-submodule of $\mathbb{R}^n$. A lattice $\Lambda \in \mathbb{R}^n$ may be specified by a set of m basis (row) vectors $\mathbf{g}_1, \ldots, \mathbf{g}_m \in \mathbb{R}^n$, consisting of all $\mathbb{Z}$-linear combinations of the basis vectors, i.e.,

$$\Lambda = \{\mathbf{r}\mathbf{G}_\Lambda : \mathbf{r} \in \mathbb{Z}^m\},$$

where $\mathbf{G}_\Lambda \triangleq \left[\mathbf{g}_1^T | \cdots | \mathbf{g}_m^T\right]^T \in \mathbb{R}^{m\times n}$ is called a *generator matrix* for Λ. Note that $\mathbf{G}_\Lambda$ is not unique for a given Λ. We call m the *rank* of Λ, and n the *dimension* of Λ. Clearly, $m \leq n$, because otherwise the basis vectors cannot be linearly independent. When $m = n$, Λ is called a *full-rank* real lattice.

Complex lattices are natural generalizations of real lattices. Let T be a discrete subring of $\mathbb{C}$ forming a PID. Typical examples of T include the Gaussian integers $\mathbb{Z}[i]$ and the Eisenstein integers $\mathbb{Z}[\omega]$. A *T-lattice* Λ in $\mathbb{C}^n$ is a discrete T-submodule of $\mathbb{C}^n$, consisting of all T-linear combinations of a set of basis vectors. Throughout this thesis, we will focus on full-rank T-lattices for simplicity, but all the results can be easily extended to the case of non-full-rank T-lattices.

A few important notions are associated with a T-lattice. An n-dimensional T-lattice Λ partitions the space $\mathbb{C}^n$ into *congruent cells*. Such a partition is not unique. The most important example is based on the *nearest neighbor quantizer* $\mathcal{Q}_\Lambda^{\mathrm{NN}}$ that sends a point $\mathbf{x} \in \mathbb{C}^n$ to a nearest lattice point in Euclidean distance, i.e.,

$$\mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{x}) = \boldsymbol{\lambda} \in \Lambda, \quad \text{if } \forall \boldsymbol{\lambda}' \in \Lambda \left(\|\mathbf{x} - \boldsymbol{\lambda}\| \leq \|\mathbf{x} - \boldsymbol{\lambda}'\|\right),$$

where ties are broken in a systematic manner. The *Voronoi cell* $\mathcal{V}_\Lambda(\boldsymbol{\lambda})$ associated with each $\boldsymbol{\lambda} \in \Lambda$ is defined as the set of all points in $\mathbb{C}^n$ that are closest to $\boldsymbol{\lambda}$, i.e., $\mathcal{V}_\Lambda(\boldsymbol{\lambda}) \triangleq \{\mathbf{x} \in \mathbb{C}^n : \mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{x}) = \boldsymbol{\lambda}\}$. The cell $\mathcal{V}_\Lambda(\mathbf{0})$ associated with the origin is often referred to as the *Voronoi region* of Λ. Clearly, the Voronoi cells $\{\mathcal{V}_\Lambda(\boldsymbol{\lambda})\}$ have the following three properties:

1. Each cell $\mathcal{V}_\Lambda(\boldsymbol{\lambda})$ is a shift of the cell $\mathcal{V}_\Lambda(\mathbf{0})$ by $\boldsymbol{\lambda} \in \Lambda$, i.e., $\mathcal{V}_\Lambda(\boldsymbol{\lambda}) = \boldsymbol{\lambda} + \mathcal{V}_\Lambda(\mathbf{0})$.

2. The cells do not intersect, i.e., $\mathcal{V}_\Lambda(\boldsymbol{\lambda}) \cap \mathcal{V}_\Lambda(\boldsymbol{\lambda}') = \emptyset$ for all $\boldsymbol{\lambda} \neq \boldsymbol{\lambda}'$.

3. The union of the cells covers the whole space, i.e., $\bigcup_{\boldsymbol{\lambda}\in\Lambda} \mathcal{V}_\Lambda(\boldsymbol{\lambda}) = \mathbb{C}^n$.

In general, any collection of cells $\{\mathcal{R}_\Lambda(\boldsymbol{\lambda})\}$ that satisfies the above three conditions is called a set of *fundamental cells*. The cell $\mathcal{R}_\Lambda(\mathbf{0})$ associated with the origin is called a *fundamental region* and will also be denoted simply by $\mathcal{R}_\Lambda$. Note that every fundamental region of a lattice Λ has exactly the same volume, which is denoted by $V(\Lambda)$.

A *lattice quantizer* $\mathcal{Q}_\Lambda : \mathbb{C}^n \to \Lambda$ corresponding to $\mathcal{R}_\Lambda$ sends every point $\mathbf{x} \in \mathbb{C}^n$ to the lattice point $\boldsymbol{\lambda}$ that is associated with the fundamental cell $\mathcal{R}_\Lambda(\boldsymbol{\lambda})$ containing $\mathbf{x}$, i.e.,

$$\mathcal{Q}_\Lambda(\mathbf{x}) = \boldsymbol{\lambda} \in \Lambda, \text{ if } \mathbf{x} \in \mathcal{R}_\Lambda(\boldsymbol{\lambda}).$$

Hence, any point $\mathbf{x}$ in $\mathbb{C}^n$ can be uniquely expressed as the sum of a lattice point and a point in the fundamental region $\mathcal{R}_\Lambda$, i.e., $\mathbf{x} = \mathcal{Q}_\Lambda(\mathbf{x}) + (\mathbf{x} - \mathcal{Q}_\Lambda(\mathbf{x}))$, where $\mathbf{x} - \mathcal{Q}_\Lambda(\mathbf{x})$ is a point in $\mathcal{R}_\Lambda$. This implies that, for all lattice points $\boldsymbol{\lambda} \in \Lambda$ and all vectors $\mathbf{z} \in \mathbb{C}^n$,

$$\mathcal{Q}_\Lambda(\boldsymbol{\lambda} + \mathbf{z}) = \boldsymbol{\lambda} + \mathcal{Q}_\Lambda(\mathbf{z}). \tag{2.4}$$

The modulo-Λ operation is defined, for a fixed $\mathcal{Q}_\Lambda$, as

$$\mathbf{x} \bmod \Lambda = \mathbf{x} - \mathcal{Q}_\Lambda(\mathbf{x}).$$

Clearly, the modulo-Λ operation always outputs a point in the fundamental region $\mathcal{R}_\Lambda$. The modulo-Λ operation has a geometrical interpretation:

$$\mathbf{x} \bmod \Lambda = (\mathbf{x} + \Lambda) \cap \mathcal{R}_\Lambda,$$

where the *lattice shift* $\mathbf{x} + \Lambda$ is defined as $\mathbf{x} + \Lambda = \{\mathbf{x} + \boldsymbol{\lambda} : \boldsymbol{\lambda} \in \Lambda\}$.

A *T-sublattice* Λ' of Λ is a subset of Λ which is itself a T-lattice. Two lattices Λ' and Λ are said to be *nested* if Λ' is a sublattice of Λ, i.e., $\Lambda' \subseteq \Lambda$.

For each $\boldsymbol{\lambda} \in \Lambda$, the lattice shift $\boldsymbol{\lambda} + \Lambda'$ is a coset of Λ' in Λ, and the point $\boldsymbol{\lambda} \bmod \Lambda'$ is called the *coset leader* of $\boldsymbol{\lambda} + \Lambda'$. Two cosets $\boldsymbol{\lambda}_1 + \Lambda'$ and $\boldsymbol{\lambda}_2 + \Lambda'$ are either identical (when $\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 \in \Lambda'$) or disjoint (when $\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 \notin \Lambda'$). Thus, the set of all distinct cosets of Λ' in Λ, denoted by Λ/Λ', forms a partition of Λ. Algebraically, Λ/Λ' is a quotient T-module, hereafter called a *T-lattice quotient*.

2.2.2 Nested Lattice Codes

A *nested lattice code* $\mathcal{L}(\Lambda, \Lambda')$ is defined as the set of all coset leaders in Λ/Λ', i.e.,

$$\mathcal{L}(\Lambda, \Lambda') = \Lambda \bmod \Lambda' = \{\boldsymbol{\lambda} \bmod \Lambda' : \boldsymbol{\lambda} \in \Lambda\}.$$

Geometrically, $\mathcal{L}(\Lambda, \Lambda')$ is the intersection of the lattice Λ with the fundamental region $\mathcal{R}_{\Lambda'}$, i.e.,

$$\mathcal{L}(\Lambda, \Lambda') = \Lambda \cap \mathcal{R}_{\Lambda'}.$$

For this reason, the fundamental region $\mathcal{R}_{\Lambda'}$ is often interpreted as the *shaping region*. Note that there is a bijection between Λ/Λ' and $\mathcal{L}(\Lambda, \Lambda')$; in particular,

$$|\Lambda/\Lambda'| = |\mathcal{L}(\Lambda, \Lambda')| = V(\Lambda')/V(\Lambda).$$

Finally, we mention that, for reasons of energy-efficiency, it is often useful to consider a translated version of nested lattice codes. For any fixed translation vector $\mathbf{d} \in \mathbb{C}^n$, a *translated nested lattice code* $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ is defined as

$$\mathcal{L}(\Lambda, \Lambda', \mathbf{d}) = (\mathbf{d} + \Lambda) \bmod \Lambda' = (\mathbf{d} + \Lambda) \cap \mathcal{R}_{\Lambda'}.$$

Chapter 3

An Algebraic Framework

This chapter studies the design of C&F schemes via nested lattice codes. Building on the work of Nazer and Gastpar, who demonstrated the asymptotic gain of C&F using information-theoretic tools, this chapter takes an algebraic approach to show the potential of C&F in practical, non-asymptotic, settings, with a particular focus on C&F schemes with controlled block length and complexity. First, a general framework is developed for studying such nested-lattice-based C&F schemes—called lattice network coding (LNC) schemes for short—by making a direct connection between C&F and module theory. In particular, a generic LNC scheme is presented that makes no assumptions on the underlying nested lattice code. This opens up the opportunity for systematic code construction. C&F is re-interpreted in this framework, and several generalized constructions of LNC schemes are given. Second, performance/complexity tradeoffs of LNC schemes are studied, with a particular focus on hypercube-shaped LNC schemes. The error probability of this class of LNC schemes is largely determined by the minimum inter-coset distances of the underlying nested lattices. Several exemplary hypercube-shaped LNC schemes are designed based on Construction A and D, showing that nominal coding gains of 3 to 7.5 dB can be obtained with relatively short block length and reasonable decoding complexity. Finally, the possibility of decoding multiple linear combinations is considered and related to the shortest independent vectors problem. A notion of dominant solutions is developed together with a suitable lattice-reduction-based algorithm.

3.1 Introduction

C&F has emerged as a compelling information transmission scheme in wireless relay networks. C&F exploits the property that integer linear combinations of lattice points are again lattice points. Based on

this property, relays in a network attempt to decode their received signals into integer linear combinations of codewords, which they then forward to the destination. With enough such linear combinations, the destination is able to recover all the transmitted messages simply by solving a system of linear equations.

A key feature of C&F is that no CSI is required at the transmitters and the destination. This is in sharp contrast to alternative advanced strategies, such as noisy network coding [25] and quantize-map-and-forward strategy [24,38], which generally require global channel-gain information at the destinations. This makes C&F an appealing candidate for practical implementation.

Prior work on C&F mainly focuses on its asymptotic performance (e.g., [7, 12]), whose analysis is based on the existence of an (infinite) sequence of "asymptotically-good" nested lattice codes. This sequence of lattice codes requires very long block length and almost unbounded complexity.

In this chapter, we develop a generic LNC scheme that makes no particular assumption on the structure of the underlying nested lattice code, thereby enabling a variety of code-design techniques. A key aspect of this approach is a so-called "linear labeling" of the points in a nested lattice code that gives rise to a beneficial compatibility between the $\mathbb{C}$-linear arithmetic operations performed by the wireless channel and the linear operations in the message space that are required for linear network coding. Similar to vector-space-based noncoherent network coding (e.g., [39]), the linear labelings of this chapter induce a noncoherent end-to-end network coding channel with a message space having, in general, a module-theoretic algebraic structure, thereby providing a foundation for achieving noncoherent network coding over general wireless relay networks.

We study the error performance of a class of hypercube-shaped LNC schemes, and show that the error performance is largely determined by the minimum inter-coset distance of the underlying nested lattice code. By way of illustration, we adapt several known lattice constructions to give three exemplary LNC schemes that provide nominal coding gains of 3 to 7.5 dB while admitting relatively short block length and reasonable decoding complexity.

We also study the possibility that a relay may attempt to decode more than one linearly independent combination of messages, and we relate this problem to the "shortest independent vectors problem" in lattices [40]. For this problem, a notion of dominant solutions is introduced together with a lattice-reduction-based algorithm, which may be of independent interest.

Our generic LNC scheme can be seen as generalization of several previous PNC schemes [5–7]. The earliest PNC schemes were applied to a two-way relay channel in which the relay attempts to decode the modulo-two sum (XOR) of the transmitted messages, as explained in Chapter 1. Subsequently, it was observed in [41, 42] that the XOR can be replaced by a family of functions satisfying the so-called "exclusive law of network coding." Furthermore, the choice of function can potentially be adapted to

the instantaneous channel realizations, although a complicated computer search may be needed [42] to choose the function optimally, even in the case of low-dimensional constellations such as 16-QAM. Unlike these work, our LNC scheme considers only linear combinations, and so it provides an efficient method, even in high-dimensional spaces, to perform channel-adaptive decoding.

There are some other practical code constructions for C&F in the literature (see, e.g., [43–47]). Compared to these work, our algebraic framework provides a systematic approach, which clarifies the practical implementation of C&F, enriches the design space of C&F, and enables new code constructions.

After the conference publication of an earlier version of this work [48] (see also [49, 50]), several papers have appeared following our algebraic framework. For example, the work of [46] presents several design examples based on Eisenstein lattices, which can achieve a shaping gain of 0.167 dB compared to our examples based on Gaussian lattices. The work of [47] studies the existence of asymptotically-good nested lattices over Eisenstein integers, which can offer higher computation rates for certain channel realizations compared to the computation rates in [11] (which are based on Gaussian integers).

3.2 Motivating Examples

In this section, we illustrate the role of algebra in PNC with a particular focus on the two-way relay channel, where two wireless terminals attempt to exchange their messages W_1, W_2 through a relay node. For this channel model, a PNC scheme consists of two rounds of communication. In the first round, the terminals simultaneously transmit their signals X_1, X_2 to the relay, and the relay tries to decode a function $f(W_1, W_2)$ of the messages from the received signal Y. In the second round, the relay broadcasts the decoded function $f(W_1, W_2)$ to the terminals, based on which each terminal recovers the other message with its own message held as side information.

To illustrate how a PNC scheme works, we assume that the channels between terminals and the relay are complex-valued flat-fading channels with additive white Gaussian noise, that the messages W_1, W_2 take values in the set $\{00, 01, 10, 11\}$, and that (uncoded) Gray-labeled quaternary phase-shift-keying (QPSK) modulation is used, with the signal constellation given in Fig. 3.1. The channel gains between the terminals and the relay are denoted as h_1 and h_2. Furthermore, we assume that the relay aims to decode the XOR of the messages.

We first consider the ideal special case in which the channel gains are precisely unity, i.e., $h_1 = h_2 = 1$. The received constellation is depicted in Fig. 3.2(a), together with the decision region for XOR decoding. Although some received points are overlapping, say point $(W_1, W_2) = (01, 11)$ and point $(11, 01)$, the overlapping points have the same XOR value, resulting in no ambiguity.

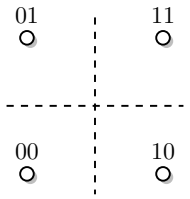


Figure 3.1: Transmitted QPSK constellation.

Next, suppose that the channel gains are $h_1 = 1, h_2 = i$. In this scenario, unfortunately, overlapping points have different XOR values; see Fig. 3.2(b). For instance, point $(01, 10)$ has XOR value $01 \oplus 10 = 11$; whereas point $(11, 11)$ has XOR value 00.

To solve this ambiguity, one natural attempt is to let the relay decode some linear function instead of the XOR. For example, if the relay interprets each message $W_\ell = [w_{\ell 1}\ w_{\ell 2}]$ $(\ell = 1, 2)$ as an element in $\mathbb{F}_4$ by mapping it to $w_{\ell 1}\alpha + w_{\ell 2}$ (where α is a primitive element of $\mathbb{F}_4$) and tries to decode the function $f_1(W_1, W_2) = W_1 + \alpha W_2$, then both point $(01, 10)$ and point $(11, 11)$ give rise to the same value 10. However, there are still some ambiguities that cannot be resolved by this function (the shaded dots in Fig. 3.2(b)).

In fact, no linear functions over $\mathbb{F}_4$ can resolve all the ambiguities in the received constellation, and the relay has to make use of the structure of a finite ring rather than that of a finite field. Specifically, let the relay interpret each message $W_\ell = [w_{\ell 1}\ w_{\ell 2}]$ as $w_{\ell 1} + w_{\ell 2} i \in \mathbb{Z}_2[i]$ with addition and multiplication defined as

$$a + bi + c + di = [a + c]_2 + [b + d]_2 i,$$
$$(a + bi)(c + di) = [ac - bd]_2 + [ad + bc]_2 i,$$

where $[\cdot]_2$ denotes the mod 2 operation. Then the function $f_2(W_1, W_2) = W_1 + iW_2$ is able to resolve all the ambiguities in Fig. 3.2(b). Moreover, the function f_2 works well even under other channel gains. In other words, the finite ring $\mathbb{Z}_2[i]$ seems to be a "good match" for QPSK constellation. This is not a coincidence. As we will see later, every nested-lattice-based constellation has such a good match.

3.3 Problem Statement

This section gives a general definition of a *linear* physical-layer network coding (or compute-and-forward) scheme, and also describes the assumptions on the system model made in this chapter. We focus on the problem faced by a receiver node of decoding one or more linear combinations of simultaneously

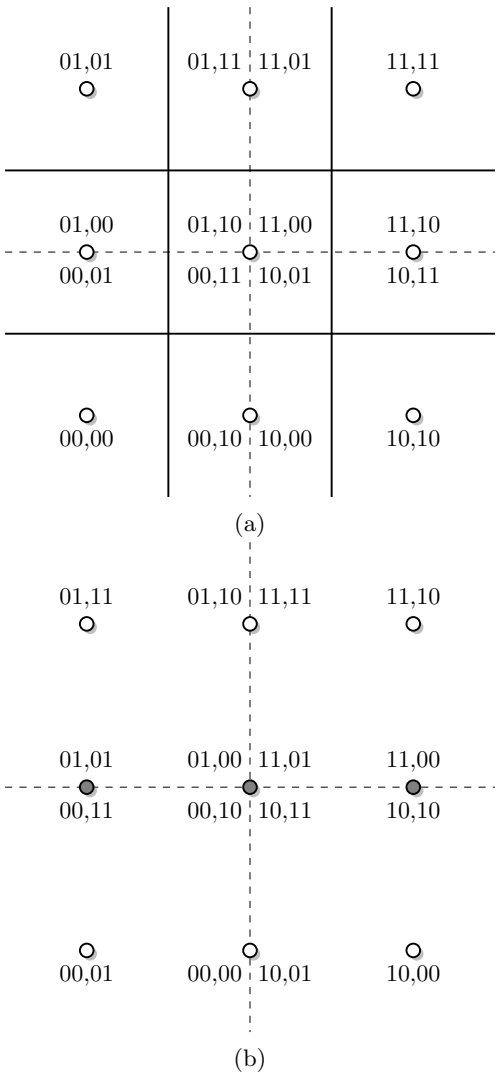


Figure 3.2: Received constellations with QPSK when (a) $h_1 = h_2 = 1$, and (b) $h_1 = 1$, $h_2 = i$.

transmitted messages, as it is at the heart of any system employing physical-layer network coding (see [9] for such a discussion). We conclude the section by briefly describing some achievability results obtained by Nazer and Gastpar in [11].

While linear network coding is traditionally defined over a finite field [3, 4], our description considers a more general notion of linear network coding over a finite commutative ring R. In this context, the message space, i.e., the set from where message packets are drawn, is no longer a vector space, but an R-module [51]. As hinted at in Sec. 5.2 and as will become clear in Sec. 3.4, ring-linear network coding is required if we wish to ensure compatibility with a *general* lattice network coding scheme.

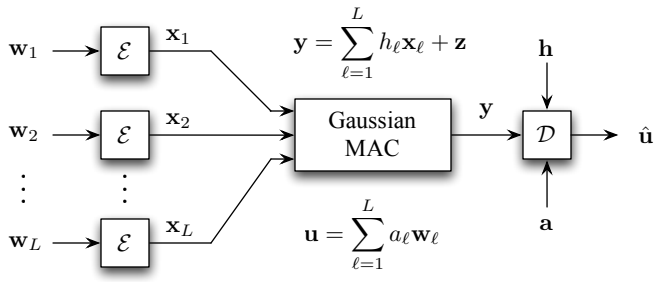


Figure 3.3: Computing a linear function over a Gaussian multiple-access channel.

3.3.1 System Model

Consider a multiple-access channel with L transmitters and a single receiver subject to block fading and additive white Gaussian noise, as illustrated in Fig. 3.3.

Channel inputs are denoted by $\mathbf{x}_1, \ldots, \mathbf{x}_L \in \mathbb{C}^n$ and the channel output is given by

$$\mathbf{y} = \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \mathbf{z}$$

where $h_1, \ldots, h_L \in \mathbb{C}$ are channel gains (fading coefficients) and $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}, N_0 \mathbf{I}_n)$ is a circularly-symmetric jointly-Gaussian complex random vector. We assume that the channel gains are perfectly known at the receiver but are *unknown* at the transmitters.

Transmitter ℓ is subject to a power constraint given by

$$\frac{1}{n} E\left[\|\mathbf{x}_\ell\|^2\right] \leq P_\ell$$

where the expectation is taken with respect to a uniform distribution over the corresponding message space. For simplicity (and without loss of generality), we assume that the power constraint is symmetric, $P_1 = \cdots = P_L \triangleq P$, and that any asymmetric power constraints are incorporated by appropriately scaling the channel gains h_ℓ.

For convenience, we define

$$\mathsf{SNR} \triangleq P/N_0.$$

Note that the received SNR corresponding to signal $\mathbf{x}_\ell$ is equal to $|h_\ell|^2 P/N_0$. Hence, the interpretation of SNR as the average received SNR is only valid when $E[|h_\ell|^2] = 1$.

3.3.2 Linear Physical-Layer Network Coding

Let R be a *finite* commutative ring with identity $1 \neq 0$ and let T be some (usually infinite) commutative ring such that there exists a surjective ring homomorphism $\sigma : T \to R$. Let the *ambient space* W be a finite R-module. Note that σ automatically makes W into a T-module by defining $a\mathbf{w} = \sigma(a)\mathbf{w}$, for all $a \in T$ and all $\mathbf{w} \in W$. As an example, we may have $T = \mathbb{Z}$, $R = \mathbb{Z}/\langle 2\rangle$, $W = \mathbb{Z}/\langle 2\rangle$, and $\sigma(a) = a + \langle 2\rangle$. In the following setup, "digital-layer" network coding operates on W over R, while physical-layer network coding operates on W over T, and the ring homomorphism σ guarantees the compatibility of such operations.

For each $\ell \in \{1, \ldots, L\}$, let the *message space* of transmitter ℓ be an R-submodule $W_\ell \subseteq W$. A T-linear PNC scheme with block length n consists of L encoders

$$\mathcal{E}_\ell : W_\ell \to \mathbb{C}^n$$

each taking a message vector $\mathbf{w}_\ell \in W_\ell$ to a signal vector $\mathbf{x}_\ell \in \mathbb{C}^n$, and a decoder

$$\mathcal{D} : \mathbb{C}^n \to W$$

that takes a received signal $\mathbf{y} \in \mathbb{C}^n$ and attempts to compute one (or more) T-linear combination(s) of the messages, such as

$$\mathbf{u} = \sum_{\ell=1}^{L} a_\ell \mathbf{w}_\ell \in W$$

whose coefficients $a_\ell \in T$ may or may not have been specified *a priori*. It is understood that any T-linear combinations computed by the decoder are subsequently delivered to the digital layer as R-linear combinations, such as

$$\mathbf{u} = \sum_{\ell=1}^{L} a_\ell \mathbf{w}_\ell = \sum_{\ell=1}^{L} \sigma(a_\ell)\mathbf{w}_\ell \in W$$

obtained by the application of σ on each coefficient.

The above generic description of the decoder may be specialized depending on the problem at hand. Specifically, any further information given to the decoder (such as side information about the channel gains) will be denoted as additional arguments to $\mathcal{D}$. Similarly, any further information provided by the decoder will be denoted as additional outputs of $\mathcal{D}$. Note that, in this chapter, we always assume that the channel-gain vector $\mathbf{h} \triangleq (h_1, \ldots, h_L) \in \mathbb{C}^L$ is perfectly known at the receiver.

For simplicity of notation, let $\mathbf{W} \in W^L$ be a matrix corresponding to the vertical stacking of $\mathbf{w}_1, \ldots, \mathbf{w}_L \in W$, taken as row vectors. If the coefficient vector $\mathbf{a} = (a_1, \ldots, a_L) \in T^L$ for the desired

linear combination is specified *a priori*, we will write

$$\mathcal{D} : \mathbb{C}^n \times \mathbb{C}^L \times T^L \to W, \quad \hat{\mathbf{u}} = \mathcal{D}(\mathbf{y}|\mathbf{h}, \mathbf{a}).$$

In this case, a decoding error is made if $\hat{\mathbf{u}} \neq \mathbf{aW}$. The corresponding probability of error is denoted by $P_e(\mathbf{h}, \mathbf{a})$. This decoder is illustrated in Fig. 3.3.

If no coefficient vectors are given *a priori*, but instead are required to be computed "on-the-fly" by the receiver, then we will write

$$\mathcal{D} : \mathbb{C}^n \times \mathbb{C}^L \to W^m \times T^{Lm}$$

$$(\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_m, \mathbf{a}_1, \ldots, \mathbf{a}_m) = \mathcal{D}(\mathbf{y}|\mathbf{h})$$

where m denotes the number of linear combinations computed. In this case, a decoding error is made if $\hat{\mathbf{u}}_i \neq \mathbf{a}_i\mathbf{W}$, for some $i \in \{1, \ldots, m\}$.

Since a message is transmitted over n (complex) channel uses, we define the *message rate* (spectral efficiency) for transmitter ℓ as $\mathsf{R}_{\mathsf{mes},\ell} \triangleq \frac{1}{n}\log_2|W_\ell|$, measured in bits per complex dimension. Throughout this dissertation we assume that all encoders are identical, $\mathcal{E}_1 = \ldots = \mathcal{E}_\ell \triangleq \mathcal{E}$, thus there is a single message space W with message rate[1]

$$\mathsf{R}_{\mathsf{mes}} \triangleq \frac{1}{n}\log_2|W|.$$

As the following examples illustrate, a number of existing PNC schemes can be described in this framework.

Example 3.1. *Let $L = 2$, $n = 1$, $T = \mathbb{Z}$ and $R = W = \mathbb{Z}/\langle 2\rangle$. Consider the encoder*

$$\mathcal{E}(w) = \gamma\left(\tilde{\sigma}(w) - \frac{1}{2}\right), \ w \in \mathbb{Z}/\langle 2\rangle$$

where $\gamma > 0$ is a scaling factor, and $\tilde{\sigma} : \mathbb{Z}/\langle 2\rangle \to \mathbb{Z}$ is defined as

$$\tilde{\sigma}(w) = \begin{cases} 1, & \text{when } w = 1 + \langle 2\rangle \\ 0, & \text{when } w = 0 + \langle 2\rangle. \end{cases}$$

Suppose $\mathbf{h} = [1\ 1] \in \mathbb{C}^2$. Let $\mathbf{a} = [1\ 1] \in \mathbb{Z}^2$ be a fixed coefficient vector. Then a decoder can be

[1]Note that this setup can be easily extended to the asymmetric case where the transmitters have different message rates or different power constraints. The key idea is to replace a pair of nested lattices with a nested lattice chain. See, e.g. [52], for details.

constructed as

$$\mathcal{D}(y|\mathbf{h},\mathbf{a}) = \begin{cases} 1+\langle 2\rangle, & \text{if } |Re\{y\}| < \gamma/2 \\ 0+\langle 2\rangle, & \text{otherwise.} \end{cases}$$

This is the simplest form of PNC [5, 6], which may be understood as XOR decoding under BPSK modulation, in the case of two users with equal channel gains.

Example 3.2. *Let $L = 2$, $n = 1$, $T = \mathbb{Z}[i]$ and $R = W = \mathbb{Z}[i]/\langle m\rangle$, where m is some positive integer. Consider the encoder*

$$\mathcal{E}(w) = \gamma\left(\tilde{\sigma}(w) - d\right),\ w \in \mathbb{Z}[i]/\langle m\rangle$$

where $d = \left(\frac{m-1}{2}\right)(1+i)$, $\gamma > 0$ is a scaling factor, and $\tilde{\sigma} : \mathbb{Z}[i]/\langle m\rangle \to \mathbb{Z}[i]$ is defined as

$$\tilde{\sigma}(a + bi + \langle m\rangle) = (a \bmod m) + (b \bmod m)i.$$

First, suppose $\mathbf{h} = [1\ 1] \in \mathbb{C}^2$. Let $\mathbf{a} = [1\ 1] \in \mathbb{Z}[i]^2$ be the fixed coefficient vector. Then a natural (although suboptimal) decoder is given by

$$\mathcal{D}(y|\mathbf{h},\mathbf{a}) = (\lfloor Re\{y'\}\rceil \bmod m) + (\lfloor Im\{y'\}\rceil \bmod m)\, i + \langle m\rangle,$$

where $y' = y/\gamma + (a_1 + a_2)d$ and $\lfloor\cdot\rceil$ denotes the rounding operation. This scheme is known as the m^2-QAM PNC scheme [5]. Next, suppose $\mathbf{h} = [1\ i] \in \mathbb{C}^2$. Let $\mathbf{a} = [1\ i] \in \mathbb{Z}[i]^2$ be the fixed coefficient vector. Then the above decoder generalizes the example discussed in Sec. 5.2.

3.3.3 Achievable Rates

We now mention some known achievable rates for the case of a single given coefficient vector, under the assumptions of Section 3.3.1. These results were obtained by Nazer and Gastpar [11].

Theorem 3.1 ([11])**.** *For all $\epsilon > 0$, all sufficiently large n, and some appropriately chosen prime integer p, there exists a $\mathbb{Z}[i]$-linear PNC scheme with block length n satisfying the following properties:*

1. *the message space is $W = (\mathbb{Z}[i]/\langle p\rangle)^k$ for some k;*

2. *for any channel-gain vector $\mathbf{h} \in \mathbb{C}^L$ and any non-zero coefficient vector $\mathbf{a} \in \mathbb{Z}[i]^L$, the probability of decoding error $P_e(\mathbf{h},\mathbf{a})$ is smaller than ϵ if k is such that the message rate R_{mes} is smaller than*

the computation rate

$$R_{\mathsf{comp}}(\mathbf{h},\mathbf{a}) \triangleq \max_{\alpha\in\mathbb{C}} \log_2\left(\frac{\mathit{SNR}}{\|\alpha\mathbf{h}-\mathbf{a}\|^2\,\mathit{SNR}+|\alpha|^2}\right).$$

Moreover, the optimal value of α in the above expression is given by

$$\alpha_{\mathit{opt}} = \frac{\mathbf{a}\mathbf{h}^{\mathsf{H}}\,\mathit{SNR}}{\|\mathbf{h}\|^2\,\mathit{SNR}+1} \tag{3.1}$$

which results in

$$R_{\mathsf{comp}}(\mathbf{h},\mathbf{a}) = \log_2\left(\frac{\mathit{SNR}}{\mathbf{a}\mathbf{M}\mathbf{a}^{H}}\right),$$

where

$$\mathbf{M} = \mathit{SNR}\,\mathbf{I}_L - \frac{\mathit{SNR}^2}{\mathit{SNR}\,\|\mathbf{h}\|^2+1}\mathbf{h}^{\mathsf{H}}\mathbf{h} \tag{3.2}$$

and $\mathbf{I}_L$ is the $L\times L$ identity matrix.

Remark: In the proof of the above result, p has to grow appropriately with n such that $n/p\to 0$ as $n\to\infty$ [11].

Theorem 3.1 is based on the existence of a "good" sequence of nested lattices of increasing dimension. Criteria to design low complexity, finite-dimensional PNC schemes are not immediately obvious from these results. In the remainder of this chapter, we will develop an algebraic framework for studying linear PNC schemes, which facilitates the construction and analysis of practical PNC schemes.

3.4 Lattice Network Coding

3.4.1 Linear Labelings

Let T be a discrete subring of $\mathbb{C}$ forming a PID, and let $\Lambda\subseteq\mathbb{C}^n$ and $\Lambda'\subseteq\Lambda$ be two full-rank T-lattices (called *fine* and *coarse*, respectively) so that the index $|\Lambda/\Lambda'|$ of Λ' in Λ is finite. Recall that Λ/Λ' is a quotient T-module, i.e., it is a set closed under addition and multiplication by elements of T. Specifically, addition of cosets is defined as $(\boldsymbol{\lambda}_1+\Lambda')+(\boldsymbol{\lambda}_2+\Lambda')\triangleq(\boldsymbol{\lambda}_1+\boldsymbol{\lambda}_2+\Lambda')$, for all $\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2\in\Lambda$, multiplication by $r\in T$ is defined as $r(\boldsymbol{\lambda}+\Lambda')\triangleq(r\boldsymbol{\lambda}+\Lambda')$, for all $\boldsymbol{\lambda}\in\Lambda$, and multiplication distributes over addition. An immediate consequence is that $\sum_{\ell=1}^{L} r_\ell(\boldsymbol{\lambda}_\ell+\Lambda') = (\sum_{\ell=1}^{L} r_\ell\boldsymbol{\lambda}_\ell)+\Lambda'$, i.e., a T-linear combination of cosets is determined by the linear combination of their coset representatives. This is the main property exploited in a lattice network coding (LNC) scheme.

Conceptually, an LNC scheme is a T-linear PNC scheme based on a finite lattice quotient Λ/Λ', in which each transmitter sends an information-embedding coset through a coset representative, and each receiver recovers one or more T-linear combinations of the transmitted coset representatives (which can potentially be forwarded to other nodes according to the same scheme). Upon receiving enough such combinations, the destination is able to decode all information-embedding cosets from the transmitters.

To facilitate practical implementation, we will specify a map $\varphi : \Lambda \to W$ from lattice points in Λ to messages in the message space W for use in the above architecture. The map φ must satisfy two conditions:

1. all points in the same coset are mapped to the same message, i.e., if for any two points $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \Lambda$ with $\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 \in \Lambda'$, $\varphi(\boldsymbol{\lambda}_1) = \varphi(\boldsymbol{\lambda}_2)$;

2. the map φ is T-linear, i.e., for all $r_1, r_2 \in T$ and all $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \Lambda$, we have $\varphi(r_1\boldsymbol{\lambda}_1 + r_2\boldsymbol{\lambda}_2) = r_1\varphi(\boldsymbol{\lambda}_1) + r_2\varphi(\boldsymbol{\lambda}_2)$.

We refer to the map φ as a *linear labeling* of Λ. As we shall see, it is this linear labeling that induces a natural compatibility between the $\mathbb{C}$-linear arithmetic of the multiple access channel observed by the receiver and the T-linear arithmetic desired in the message space.

The existence of the aforementioned linear labeling is guaranteed by the following theorem, which provides a *canonical decomposition* for any finite T-lattice quotient Λ/Λ'.

Theorem 3.2. *Let T be a PID and let Λ and $\Lambda' \subseteq \Lambda$ be T-lattices such that $|\Lambda/\Lambda'|$ is finite. Then, for some nonzero, non-unit elements $\pi_1, \pi_2, \ldots, \pi_k \in T$ satisfying the divisibility relations $\pi_1 \mid \pi_2 \mid \cdots \mid \pi_k$, we have*

$$\Lambda/\Lambda' \cong T/\langle\pi_1\rangle \times T/\langle\pi_2\rangle \times \cdots \times T/\langle\pi_k\rangle. \tag{3.3}$$

Moreover, there exists a surjective T-module homomorphism $\varphi : \Lambda \to T/\langle\pi_1\rangle \times \cdots \times T/\langle\pi_k\rangle$ whose kernel is Λ'.

Proof. The first statement follows from Theorem 2.2 since Λ/Λ' is a finite T-module. The second statement then follows from the First Isomorphism Theorem [34]. □

Evidently, the map φ is obtained as the composition of the natural projection from Λ to the quotient Λ/Λ' with the isomorphism of (3.3). According to Theorem 3.2, when the message space W is taken as the canonical decomposition in the right-hand side of (3.3), i.e.,

$$W = T/\langle\pi_1\rangle \times T/\langle\pi_2\rangle \times \cdots \times T/\langle\pi_k\rangle,$$

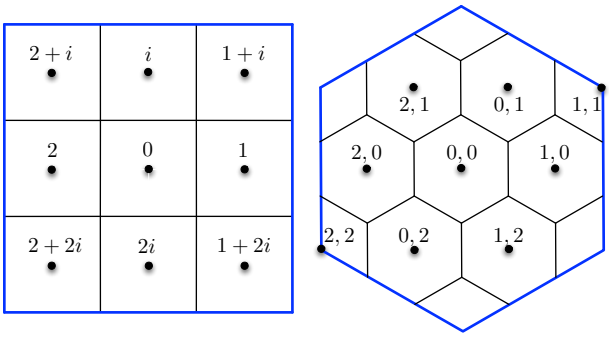


Figure 3.4: Linear labelings for Examples 3.3 and 3.4.

the map φ is indeed a linear labeling. The following examples provide two concrete linear labelings, which are depicted in Fig. 3.4.

Example 3.3. *Let $\Lambda = \mathbb{Z}[i]$ and $\Lambda' = 3\mathbb{Z}[i]$. Let $T = \mathbb{Z}[i]$ and $W = \mathbb{Z}[i]/\langle 3\rangle$. Consider the map $\varphi : \Lambda \to W$ given by*

$$\varphi(a + bi) = a + bi + \langle 3\rangle.$$

It is easy to check that the map φ is $\mathbb{Z}[i]$-linear and its kernel is $3\mathbb{Z}[i]$.

Example 3.4. *Let Λ be the (real) hexagonal lattice generated by $\mathbf{g}_1 = (1, 0)$ and $\mathbf{g}_2 = (1/2, \sqrt{3}/2)$. Let $\Lambda' = 3\Lambda$. Let $T = \mathbb{Z}$ and $W = \mathbb{Z}/\langle 3\rangle \times \mathbb{Z}/\langle 3\rangle$. Consider the map $\varphi : \Lambda \to W$ given by*

$$\varphi(a\mathbf{g}_1 + b\mathbf{g}_2) = (a \bmod 3, b \bmod 3).$$

It is easy to check that the map φ is $\mathbb{Z}$-linear and its kernel is 3Λ.

Linear labelings play a key role in LNC, as they directly map a T-linear combination of transmitted lattice points to a T-linear combination of transmitted messages, i.e., the latter can be immediately extracted from the former.

It is also convenient to define an inverse operation, mapping a message to a corresponding lattice point; this is done through an *embedding map* $\tilde{\varphi} : W \to \Lambda$. This map must be an injective function compatible with the linear labeling, so it must satisfy

$$\varphi(\tilde{\varphi}(\mathbf{w})) = \mathbf{w}, \quad \text{for all } \mathbf{w} \in W.$$

Equipped with a linear labeling φ and and embedding map $\tilde{\varphi}$, a high-level description of a generic LNC scheme can be given as follows. Each encoder ℓ maps a message $\mathbf{w}_\ell \in W$ to a lattice point $\mathbf{x}_\ell \in \Lambda$

labeled by $\mathbf{w}_\ell$, i.e., $\mathbf{x}_\ell = \tilde{\varphi}(\mathbf{w}_\ell)$. The decoder, upon the reception of $\mathbf{y}$, and given a coefficient vector $\mathbf{a} = (a_1, \ldots, a_L)$, attempts to compute the T-linear combination of transmitted lattice points

$$\boldsymbol{\lambda} = \sum_{\ell=1}^{L} a_\ell \mathbf{x}_\ell$$

from which it would be able to extract the corresponding linear combination of messages

$$\mathbf{u} = \varphi(\boldsymbol{\lambda}) = \sum_{\ell=1}^{L} a_\ell \varphi(\mathbf{x}_\ell) = \sum_{\ell=1}^{L} a_\ell \mathbf{w}_\ell.$$

In more detail, the decoder proceeds in three steps. First, it scales the received signal by a factor of α, obtaining

$$\alpha \mathbf{y} = \alpha \sum_{\ell=1}^{L} h_\ell \mathbf{x}_\ell + \alpha \mathbf{z} = \boldsymbol{\lambda} + \mathbf{n}_{\mathsf{eff}} \tag{3.4}$$

where

$$\mathbf{n}_{\mathsf{eff}} = \sum_{\ell=1}^{L} (\alpha h_\ell - a_\ell) \mathbf{x}_\ell + \alpha \mathbf{z} \tag{3.5}$$

is called the *effective noise*. Note that we can view (3.4) as an *equivalent point-to-point channel* under lattice coding: an effective message $\mathbf{u}$ is encoded as a lattice point $\boldsymbol{\lambda}$, which is then additively corrupted by the (signal-dependent and not necessarily Gaussian) effective noise $\mathbf{n}_{\mathsf{eff}}$.

Second, the decoder quantizes the scaled received signal with the fine lattice to obtain

$$\hat{\boldsymbol{\lambda}} = \mathcal{Q}_\Lambda(\alpha \mathbf{y}) = \mathcal{Q}_\Lambda(\boldsymbol{\lambda} + \mathbf{n}_{\mathsf{eff}}) = \boldsymbol{\lambda} + \mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}}) \tag{3.6}$$

where (3.6) follows from the property (2.4) of a lattice quantizer.

The last step is to apply the linear labeling, obtaining

$$\hat{\mathbf{u}} = \varphi(\hat{\boldsymbol{\lambda}}) = \varphi\left(\boldsymbol{\lambda} + \mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}})\right) = \mathbf{u} + \varphi\left(\mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}})\right).$$

The decoder makes an error if and only if $\varphi\left(\mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}})\right) = \mathbf{0}$ and therefore if and only if $\mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}}) \in \Lambda'$. This is intuitive: if $\mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}}) \in \Lambda'$, then the decoded lattice point $\hat{\boldsymbol{\lambda}}$ is in the same coset as $\boldsymbol{\lambda}$ and is thus labeled with $\mathbf{u}$. On the other hand, if the decoded lattice point $\hat{\boldsymbol{\lambda}}$ is labeled with $\mathbf{u}$, then we must have $\varphi(\mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}})) = \mathbf{0}$, which implies $\mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}}) \in \Lambda'$, since the kernel of φ is Λ'.

To sum up, the above encoding-decoding architecture is depicted in Fig. 3.5. The encoder $\mathcal{E} : W \to \mathbb{C}^n$ is given by

$$\mathbf{x}_\ell = \mathcal{E}(\mathbf{w}_\ell) = \tilde{\varphi}(\mathbf{w}_\ell)$$

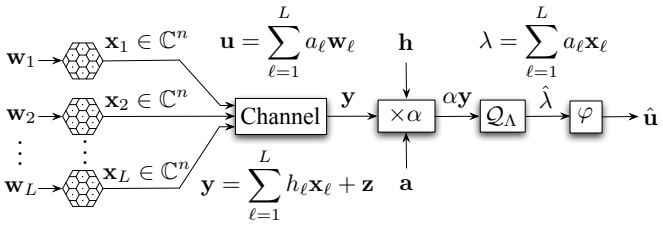


Figure 3.5: Encoding and decoding architecture for LNC.

and the decoder $\mathcal{D} : \mathbb{C}^n \times \mathbb{C}^L \times T^L$ is given by

$$\hat{\mathbf{u}} = \mathcal{D}(\mathbf{y}|\mathbf{h}, \mathbf{a}) = \varphi(\mathcal{Q}_\Lambda(\alpha \mathbf{y}))$$

where α is a scaling factor chosen by the decoder based on $\mathbf{h}$ and $\mathbf{a}$, which will be discussed fully in the next section. Intuitively, the purpose of α is to reduce the effective noise $\mathbf{n}_{\mathsf{eff}}$, by trading off between *self noise* (the first term in (3.5) due to non-integer channel gains) and Gaussian noise.

Clearly, the encoding-decoding complexity of an LNC scheme is not essentially different from that for a point-to-point channel using the same nested lattice code. Further, the error probability of the scheme can be characterized by Proposition 3.1, as explained before.

Proposition 3.1. *The message* $\mathbf{u} = \sum_{\ell=1}^{L} a_\ell \mathbf{w}_\ell$ *is computed incorrectly if and only if* $\mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}}) \notin \Lambda'$. *That is,* $\Pr[\hat{\mathbf{u}} \neq \mathbf{u}] = \Pr[\mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}}) \notin \Lambda']$.

In practice, the nearest-neighbor quantizer $\mathcal{Q}_\Lambda^{\mathrm{NN}}$ is often preferred in the implementation of the decoder. This is to reduce the error probability, as we will see in Sec. 3.5. Moreover, for reasons of energy-efficiency, a nested lattice code $\mathcal{L}(\Lambda, \Lambda')$ is usually preferred in the implementation of the encoder. In this case, the encoder takes the messages in W to their *minimum-energy* coset representatives, i.e., the embedding map is chosen to satisfy

$$\tilde{\varphi}(\mathbf{w}_\ell) = \tilde{\varphi}(\mathbf{w}_\ell) \bmod \Lambda'$$

where the shaping region $\mathcal{R}_{\Lambda'}$ is chosen as the Voronoi region.

Sometimes, a *translated* nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ can be used to further reduce the energy consumption. Such techniques are well studied in the area of Voronoi constellations (see, e.g., [53, 54]). Specifically, a translated version of a generic LNC scheme consists of an encoder $\mathcal{E} : W \to \mathbb{C}^n$

$$\mathbf{x}_\ell = \mathcal{E}(\mathbf{w}_\ell) \triangleq (\mathbf{d} + \tilde{\varphi}(\mathbf{w}_\ell)) \bmod \Lambda'$$

and a decoder $\mathcal{D} : \mathbb{C}^n \times \mathbb{C}^L \times R^L \to W$

$$\hat{\mathbf{u}} = \mathcal{D}(\mathbf{y} \mid \mathbf{h}, \mathbf{a}) \triangleq \varphi\left(\mathcal{Q}_\Lambda\left(\alpha\mathbf{y} - \sum_{\ell=1}^{L} a_\ell \mathbf{d}\right)\right).$$

Clearly, the encoding is a mapping from a message $\mathbf{w}_\ell$ to its corresponding codeword in the translated nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$. To understand the decoder, note that

$$\begin{aligned}\mathbf{y}' &= \sum_\ell \alpha h_\ell \mathbf{x}_\ell + \alpha\mathbf{z} - \sum_\ell a_\ell \mathbf{d} \\ &= \sum_\ell a_\ell(\mathbf{x}_\ell - \mathbf{d}) + \sum_\ell (\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha\mathbf{z} \\ &= \sum_\ell a_\ell \tilde{\varphi}(\mathbf{w}_\ell) + \mathbf{n}_{\mathsf{eff}},\end{aligned}$$

where $\mathbf{n}_{\mathsf{eff}} \triangleq \sum_\ell (\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha\mathbf{z}$ is the effective noise. In other words, the operation $\mathbf{y}' = \alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d}$ induces a "virtual" point-to-point channel with channel input $\sum_\ell a_\ell \tilde{\varphi}(\mathbf{w}_\ell)$ and channel noise $\mathbf{n}_{\mathsf{eff}}$. Since the labeling φ is T-linear, we have

$$\varphi\left(\sum_\ell a_\ell \tilde{\varphi}(\mathbf{w}_\ell)\right) = \sum_\ell a_\ell \varphi(\tilde{\varphi}(\mathbf{w}_\ell)) = \sum_\ell a_\ell \mathbf{w}_\ell.$$

Hence, decoding is correct if and only if $\varphi(\mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}})) = 0$, or equivalently, $\mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}}) \in \Lambda'$. Therefore, Proposition 3.1 holds unchanged in this case.

Finally, note that the message rate of an LNC scheme can be computed geometrically as well as algebraically, as

$$\begin{aligned}\mathsf{R}_{\mathsf{mes}} &= \frac{1}{n}\log_2\left(V(\Lambda')/V(\Lambda)\right) \\ &= \frac{1}{n}\sum_{i=1}^{k}\log_2 |T/\langle\pi_i\rangle|.\end{aligned}$$

3.4.2 Construction of the Linear Labeling

In this section, by applying the Smith normal form theorem, we provide an explicit construction of the linear labeling φ and an embedding map $\tilde{\varphi}$.

Theorem 3.3. *Let Λ/Λ' be a finite nested T-lattice quotient. Then there exist generator matrices $\mathbf{G}_\Lambda$*

and $\mathbf{G}_{\Lambda'}$ *for* Λ *and* Λ'*, respectively, satisfying*

$$\mathbf{G}_{\Lambda'} = \begin{bmatrix} \mathrm{diag}(\pi_1, \ldots, \pi_k) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-k} \end{bmatrix} \mathbf{G}_{\Lambda}. \tag{3.7}$$

In this case,

$$\Lambda/\Lambda' \cong T/\langle \pi_1 \rangle \times \cdots \times T/\langle \pi_k \rangle.$$

Moreover, the map

$$\varphi : \Lambda \to T/\langle \pi_1 \rangle \times \cdots \times T/\langle \pi_k \rangle$$

given by

$$\varphi(\mathbf{r}\mathbf{G}_{\Lambda}) = (r_1 + \langle \pi_1 \rangle, \ldots, r_k + \langle \pi_k \rangle)$$

is a surjective T*-module homomorphism with kernel* Λ'*.*

Proof. Let $\tilde{\mathbf{G}}_{\Lambda}$ and $\tilde{\mathbf{G}}_{\Lambda'}$ be any generator matrices for Λ and Λ', respectively. Then $\tilde{\mathbf{G}}_{\Lambda'} = \mathbf{J}\tilde{\mathbf{G}}_{\Lambda}$, for some nonsingular matrix $\mathbf{J} \in T^{n \times n}$. Since T is a PID, by Theorem 2.4, the matrix $\mathbf{J}$ has a Smith normal form $\mathbf{D} = \mathrm{diag}(d_1, \ldots, d_n)$. Since $\mathbf{J}$ is nonsingular, the diagonal entries $d_1, \ldots, d_n$ of $\mathbf{D}$ are all nonzero. Thus, $d_1, \ldots, d_n$ can be expressed as

$$d_1, \ldots, d_n = u_1, \ldots, u_{n-k}, \pi_1, \ldots, \pi_k$$

where $u_1, \ldots, u_{n-k}$ are units in T, $\pi_1, \ldots, \pi_k$ are nonzero, non-unit elements in T. It follows that

$$\mathbf{D} \approx \tilde{\mathbf{D}} \triangleq \begin{bmatrix} \mathrm{diag}(\pi_1, \ldots, \pi_k) & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-k} \end{bmatrix}.$$

Therefore, $\mathbf{J} \approx \tilde{\mathbf{D}}$ and there exist invertible matrices $\mathbf{P}, \mathbf{Q} \in \mathrm{GL}_n(T)$ such that $\tilde{\mathbf{D}} = \mathbf{P}\mathbf{J}\mathbf{Q}$. We take

$$\mathbf{G}_{\Lambda} = \mathbf{Q}^{-1}\tilde{\mathbf{G}}_{\Lambda}$$
$$\mathbf{G}_{\Lambda'} = \mathbf{P}\tilde{\mathbf{G}}_{\Lambda'}$$

as new generator matrices for Λ and Λ'. Clearly, we have $\mathbf{G}_{\Lambda'} = \tilde{\mathbf{D}}\mathbf{G}_{\Lambda}$. This proves the first statement.

Since the second statement follows immediately from the third statement and the First Isomorphism Theory, we need only to prove the third statement here. That is, we must show that the map φ is a surjective T-homomorphism with kernel Λ'. Since it is easy to check that the map φ is surjective and

T-linear, we will show that the kernel of φ is Λ'. Note that

$$\varphi(\mathbf{r}\mathbf{G}_\Lambda) = \mathbf{0} \iff \forall i \in \{1, \ldots, k\} r_i \in \langle \pi_i \rangle.$$

Note also that

$$\Lambda' = \{\mathbf{r}\mathbf{G}_\Lambda : r_i \in \langle \pi_i \rangle\},$$

because $\mathbf{G}_{\Lambda'} = \tilde{\mathbf{D}}\mathbf{G}_\Lambda$. Hence, the kernel of φ is indeed Λ'. □

Theorem 3.3 constructs a linear labeling $\varphi : \Lambda \to W$ explicitly. The key step is to find two generator matrices $\mathbf{G}_\Lambda$ and $\mathbf{G}_{\Lambda'}$ satisfying the relation (3.7). This can be achieved by using the Smith normal form theorem. To construct an embedding map $\tilde{\varphi}$, one shall find a pre-image for each message $\mathbf{w} = (r_1 + \langle \pi_1 \rangle, \ldots, r_k + \langle \pi_k \rangle)$. Clearly, one natural choice of $\tilde{\varphi}(\mathbf{w})$ is given by

$$\tilde{\varphi}(\mathbf{w}) = (r_1, \ldots, r_k, \underbrace{0, \ldots, 0}_{n-k})\mathbf{G}_\Lambda,$$

which provides an explicit expression for $\tilde{\varphi}(\mathbf{w})$.

The use of the Smith normal form in coding theory is not new. In the work of Forney [54, 55], it was applied to study the structure of convolutional codes as well as the linear labeling for real lattices. The goal of the Smith normal form theorem is to reduce an arbitrary matrix to a diagonal matrix, whose diagonal entries are the invariant factors. In the context of complex T-lattices, such a diagonal matrix reveals the nesting structure between the fine lattice and the coarse lattice, leading to a transparent linear labeling.

3.4.3 End-to-End Perspective

In this section, we briefly outline the use of LNC in a non-coherent network model (where destinations have no knowledge of the operations of relay nodes) rather than the coherent network model described in [11]. To provide a context, we consider a Gaussian relay network in which a generic LNC scheme is used in conjunction with a scheduling algorithm. The scheduling algorithm indicates, at each time slot, which nodes are transmitters and which nodes are receivers. As a transmitter, a node first computes a random linear combination of the packets in its buffer and then maps this combination to a transmitted signal. As a receiver, a node first decodes the received signal into one or more linear combinations of the transmitted packets and then performs (some form of) Gaussian elimination in order to discard redundant (linearly dependent) packets in the buffer.

Initially, only the source nodes have nonempty buffers containing the message packets. When the communication ends, each destination node will have collected sufficiently many linear combinations of the message packets. This induces an end-to-end linear network-coding channel in which the message space W is, in general, a T-module $T/\langle\pi_1\rangle \times \cdots \times T/\langle\pi_k\rangle$. Since modules over PIDs share much in common with vector spaces over finite fields, it would be natural to expect that many useful techniques for non-coherent network coding can be adapted here.

We use the technique of headers as an illustrating example in this section. For convenience, we rewrite the message space as

$$W = T/\langle\pi_k\rangle \times \cdots \times T/\langle\pi_1\rangle.$$

Similar to the vector-space case, we use the first m components to store headers, and the last $k-m$ components to store payloads, where m is the number of message packets. Specifically, the header for the ith message packet is a length-m tuple with $1+\langle\pi_{k-i+1}\rangle$ at position i and $0+\langle\pi_{k-j+1}\rangle$ at other positions (where $1 \le j \le m$ and $j \ne i$).

Example 3.5. *Let the message space $W = \mathbb{Z}/\langle 12\rangle \times \mathbb{Z}/\langle 6\rangle \times \mathbb{Z}/\langle 2\rangle \times \mathbb{Z}/\langle 2\rangle$. Suppose there are 2 original messages in the system. Then the matrix $\mathbf{W}$ of the source messages is of the form*

$$\mathbf{W} = \begin{bmatrix} 1+\langle 12\rangle & 0+\langle 6\rangle & a+\langle 2\rangle & b+\langle 2\rangle \\ 0+\langle 12\rangle & 1+\langle 6\rangle & c+\langle 2\rangle & d+\langle 2\rangle \end{bmatrix},$$

where $a, b, c, d \in \mathbb{Z}$.

Recall that, when the message space is a vector space, Gauss-Jordan elimination is used to recover the payloads at the destinations. As one may expect, for a more general message space, some modification of Gauss-Jordan elimination is needed. It turns out that the key step in the modification is to transform a 2×1 matrix to a row echelon form: given $a, b \in T$, return $s, t, u, v, g \in T$ such that

$$\begin{bmatrix} s & t \\ u & v \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} g \\ 0 \end{bmatrix}$$

where the determinant, $sv - tu$, is a unit from T.

Example 3.6. *Suppose that the matrix $\mathbf{W}$ of the message packets is given in Example 3.5. Suppose that a destination has received two linear combinations, $2\mathbf{w}_1 + 3\mathbf{w}_2$ and $3\mathbf{w}_1 + 2\mathbf{w}_2$, from some relay nodes performing generic C&F decoding. Then the matrix $\mathbf{Y}$ of the received packets at the destination*

is $\mathbf{Y} = \begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix} \mathbf{W}$*, which is in the form of*

$$\mathbf{Y} = \begin{bmatrix} 2+\langle 12\rangle & 3+\langle 6\rangle & c+\langle 2\rangle & d+\langle 2\rangle \\ 3+\langle 12\rangle & 2+\langle 6\rangle & a+\langle 2\rangle & b+\langle 2\rangle \end{bmatrix}.$$

To recover the payloads, we reduce the first column of $\mathbf{Y}$ *to a row echelon form. Since*

$$\begin{bmatrix} 2 & -1 \\ -3 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

over $\mathbb{Z}$ *and the determinant,* $2 \times 2 - (-1) \times (-3) = 1$*, is a unit in* $\mathbb{Z}$*, we multiply the matrix* $\begin{bmatrix} 2 & -1 \\ -3 & 2 \end{bmatrix}$ *with* $\mathbf{Y}$*, obtaining*

$$\begin{aligned} \mathbf{Y}_1 &= \begin{bmatrix} 2 & -1 \\ -3 & 2 \end{bmatrix} \mathbf{Y} \\ &= \begin{bmatrix} 1+\langle 12\rangle & 4+\langle 6\rangle & a+\langle 2\rangle & b+\langle 2\rangle \\ 0+\langle 12\rangle & 1+\langle 6\rangle & c+\langle 2\rangle & d+\langle 2\rangle \end{bmatrix}. \end{aligned}$$

In this way, we transform the matrix $\mathbf{Y}$ *to a row echelon form. Next, we transform the matrix* $\mathbf{Y}$ *to a reduced row echelon form, which can be done by subtracting* 4 *times the second row from the first row, i.e.,*

$$\mathbf{Y}_2 = \begin{bmatrix} 1 & -4 \\ 0 & 1 \end{bmatrix} \mathbf{Y}_1.$$

Now it is easy to check that $\mathbf{Y}_2 = \mathbf{W}$*. In other words, the payloads are recovered correctly.*

Although Example 3.6 only illustrates the decoding procedure for the case of $m = 2$, it can be extended to the case of $m > 2$ through a simple mathematical induction.

Finally, we would like to point out that the design of headers in Example 3.5 is suboptimal, and a better design can be made by using matrix canonical forms. The development of this idea is beyond the scope of this chapter and will instead be discussed thoroughly in Chapter 5.

3.5 Performance Analysis for Lattice Network Coding

In this section, we turn from algebra to geometry, presenting an error-probability analysis as well as its implications.

3.5.1 Error Probability for LNC

Recall that, according to Proposition 3.1, the error probability of decoding a linear function $\mathbf{u}$ is $\Pr[\hat{\mathbf{u}} \neq \mathbf{u}] = \Pr[\mathcal{Q}_\Lambda(\mathbf{n}_{\mathsf{eff}}) \notin \Lambda']$, where $\mathsf{\_eff}$ is the effective noise given by (3.5). Note that the effective noise $\mathbf{n}_{\mathsf{eff}}$ is not necessarily Gaussian, making the analysis nontrivial. To alleviate this difficulty, we focus on a special case in which the shaping region $\mathcal{R}_{\Lambda'}$ is a (rotated) hypercube in $\mathbb{C}^n$, i.e.,

$$\mathcal{R}_{\Lambda'} = \gamma \mathbf{U} \mathcal{H}_n \tag{3.8}$$

where $\gamma > 0$ is a scalar factor, $\mathbf{U}$ is any $n \times n$ unitary matrix, and $\mathcal{H}_n$ is a unit hypercube in $\mathbb{C}^n$ defined by $\mathcal{H}_n = ([-1/2, 1/2) + i[-1/2, 1/2))^n$. This case corresponds to the so-called *hypercube shaping* in [56] and has been widely used in practice[2]. The use of hypercube shaping not only simplifies the error analysis, but also leads to very simple shaping operations. However, as we will see later, there is no shaping gain under hypercube shaping. This is expected, since similar results hold for the use of lattice codes in point-to-point channels [54, 56].

In the sequel, we will provide an approximate upper bound for the error probability for LNC schemes admitting hypercube shaping. This upper bound is closely related to certain geometrical parameters of a lattice quotient as defined below.

Let us define the *minimum (inter-coset) distance* of a lattice quotient Λ/Λ' as

$$\begin{aligned} d(\Lambda/\Lambda') &\triangleq \min_{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2 \in \Lambda : \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 \notin \Lambda'} ||\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2|| \\ &= \min_{\boldsymbol{\lambda} \in \Lambda \setminus \Lambda'} ||\boldsymbol{\lambda}|| \end{aligned}$$

where $\Lambda \setminus \Lambda'$ denotes the set difference $\{\boldsymbol{\lambda} \in \Lambda : \boldsymbol{\lambda} \notin \Lambda'\}$. Note that $d(\Lambda/\Lambda')$ corresponds to the length of the shortest vectors in $\Lambda \setminus \Lambda'$. Let $K(\Lambda/\Lambda')$ denote the number of these shortest vectors.

We have the following union bound estimate on the error probability.

Theorem 3.4 (Probability of Decoding Error)**.** *Suppose that the shaping region $\mathcal{R}_{\Lambda'}$ is a (rotated) hypercube and that all the transmitted vectors are independent and uniformly distributed over $\mathcal{R}_{\Lambda'}$. Sup-*

[2]Note that all of our constructions in the next section admit hypercube shaping.

pose that $\mathcal{Q}_\Lambda(\cdot)$ is a nearest-neighbor quantizer. Then a union bound estimate on the error probability in decoding a specified linear combination is

$$\begin{aligned} &P_e(\mathbf{h}, \mathbf{a}) \\ &\lessapprox \min_{\alpha \in \mathbb{C}} K(\Lambda/\Lambda') \exp\left(-\frac{d^2(\Lambda/\Lambda')}{4N_0(|\alpha|^2 + \mathsf{SNR}\|\alpha\mathbf{h} - \mathbf{a}\|^2)}\right). \end{aligned} \tag{3.9}$$

Moreover, the optimal value of α, i.e., the value of α that minimizes the right-hand side of (3.9), is given by (3.1), which results in

$$P_e(\mathbf{h}, \mathbf{a}) \lessapprox K(\Lambda/\Lambda') \exp\left(-\frac{d^2(\Lambda/\Lambda')}{4N_0\mathbf{a}\mathbf{M}\mathbf{a}^H}\right) \tag{3.10}$$

where the matrix $\mathbf{M}$ is given by (3.2).

The proof is given in Appendix A. Note that the proof assumes the use of random dithering (translation by a random vector chosen uniformly at random from the shaping region) at the encoders, so that the transmitted vectors are uniformly distributed over the shaping region. If a fixed translation vector is used at each encoder instead of a random dither, then our result above can be viewed as an approximation.

Theorem 3.4 implies that the lattice quotient Λ/Λ' should be designed such that $K(\Lambda/\Lambda')$ is minimized and $d(\Lambda/\Lambda')$ is maximized (under a given message rate $\mathsf{R}_{\mathsf{mes}}$ and SNR), which will be discussed fully in Sec. 3.6. Further, if the receiver has the freedom to choose the coefficient vector $\mathbf{a}$, it needs to minimize the term $\mathbf{a}\mathbf{M}\mathbf{a}^{\mathsf{H}}$, which, as observed in [20], is a shortest vector problem. Theorem 3.4 can be extended to other shaping methods. A particular example is provided in [46].

3.5.2 Nominal Coding Gain

Similarly to the point-to-point case, we define the *nominal coding gain* of Λ/Λ' as

$$\gamma_c(\Lambda/\Lambda') \triangleq \frac{d^2(\Lambda/\Lambda')}{V(\Lambda)^{1/n}}.$$

Note that the nominal coding gain is invariant to scaling. For an LNC scheme with hypercube shaping, we have $V(\Lambda') = \gamma^{2n}$ and $P = \gamma^2/6$ where $\gamma > 0$ is the scalar factor in (3.8). Thus, $V(\Lambda')^{1/n} = 6P$. Note also that $V(\Lambda)^{1/n} = 2^{-\mathsf{R}_{\mathsf{mes}}} V(\Lambda')^{1/n}$. It follows that the union bound estimate in (3.10) can be expressed as

$$P_e(\mathbf{h}, \mathbf{a}) \lessapprox K(\Lambda/\Lambda') \exp\left(-\frac{3}{2}\gamma_c(\Lambda/\Lambda')2^{-\mathsf{R}_{\mathsf{mes}}}\frac{\mathsf{SNR}}{\mathbf{a}\mathbf{M}\mathbf{a}^{\mathsf{H}}}\right).$$

Thus, for a given spectral efficiency $\mathsf{R}_{\mathsf{mes}}$, the performance of such an LNC scheme can be characterized by the parameters $K(\Lambda/\Lambda')$ and $\gamma_c(\Lambda/\Lambda')$.

Note that the nominal coding gain of a baseline lattice quotient $\mathbb{Z}[i]^n/\pi\mathbb{Z}[i]^n$ is equal to 1 for all $\pi \in \mathbb{Z}[i]^*$. Thus, $\gamma_c(\Lambda/\Lambda')$ provides a first-order estimate of the performance improvement of an LNC scheme over a baseline LNC scheme. For this reason, $\gamma_c(\Lambda/\Lambda')$ will be used as a figure of merit of LNC schemes in the rest of this chapter; yet the effect of $K(\Lambda/\Lambda')$ cannot be ignored in a more detailed assessment of LNC schemes.

3.6 Design of Nested Lattices

In this section, we adapt several known lattice constructions to produce pairs of nested lattices with simple message space and high coding gain.

3.6.1 Constructions of Nested Lattices

Known methods for designing lattices include Construction A and Construction D as well as their complex versions (see, e.g., [36]). Here, we adapt these methods to construct pairs of nested lattices. In all of our examples, the Voronoi region of the coarse lattice is chosen as its fundamental region.

Nested Lattices via Construction A

Let $p > 0$ be a prime number in $\mathbb{Z}$. Let $\mathcal{C}$ be a linear code of length n over $\mathbb{Z}/\langle p\rangle$. Without loss of generality, we may assume the linear code $\mathcal{C}$ is systematic. Define a "real Construction A lattice" [36] as

$$\Lambda_r \triangleq \{\boldsymbol{\lambda} \in \mathbb{Z}^n : \sigma(\boldsymbol{\lambda}) \in \mathcal{C}\},$$

where $\sigma : \mathbb{Z}^n \to (\mathbb{Z}/\langle p\rangle)^n$ is the natural projection map. (Here, the subscript r stands for "real.") Define

$$\Lambda_r' \triangleq \{p\mathbf{r} : \mathbf{r} \in \mathbb{Z}^n\}.$$

It is easy to see that Λ_r' is a sublattice of Λ_r. Hence, we obtain a pair of nested $\mathbb{Z}$-lattices $\Lambda_r \supseteq \Lambda_r'$ from the linear code $\mathcal{C}$.

Now we "lift" this pair of nested $\mathbb{Z}$-lattices to a pair of nested $\mathbb{Z}[i]$-lattices. Let $\Lambda = \Lambda_r + i\Lambda_r$, i.e.,

$$\Lambda = \{\boldsymbol{\lambda} \in \mathbb{Z}[i]^n : \mathrm{Re}\{\boldsymbol{\lambda}\}, \mathrm{Im}\{\boldsymbol{\lambda}\} \in \Lambda_r\}.$$

Similarly, let $\Lambda' = \Lambda'_r + i\Lambda'_r$. In this way, we obtain a pair of nested $\mathbb{Z}[i]$-lattices $\Lambda \supseteq \Lambda'$. A variant of this construction was used by Nazer and Gastpar in [11].

To study the message space induced by Λ/Λ', we specify two generator matrices satisfying the relation (3.7). On the one hand, we note that the lattice Λ_r has a generator matrix $\mathbf{G}_{\Lambda_r}$ given by

$$\mathbf{G}_{\Lambda_r} = \begin{bmatrix} \mathbf{I}_k & \mathbf{B}_{k\times(n-k)} \\ \mathbf{0}_{(n-k)\times k} & p\mathbf{I}_{n-k} \end{bmatrix},$$

where $\sigma([\mathbf{I}\ \mathbf{B}])$ is a generator matrix for $\mathcal{C}$. The lifted lattice Λ has a generator matrix $\mathbf{G}_\Lambda$ that is identical to $\mathbf{G}_{\Lambda_r}$, but over $\mathbb{Z}[i]$. On the other hand, we note that the lattice Λ' has a generator matrix $\mathbf{G}_{\Lambda'}$ given by

$$\mathbf{G}_{\Lambda'} = \begin{bmatrix} p\mathbf{I}_k & p\mathbf{B}_{k\times(n-k)} \\ \mathbf{0}_{(n-k)\times k} & p\mathbf{I}_{n-k} \end{bmatrix}.$$

These two generator matrices $\mathbf{G}_\Lambda$ and $\mathbf{G}_{\Lambda'}$ satisfy

$$\mathbf{G}_{\Lambda'} = \begin{bmatrix} p\mathbf{I}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-k} \end{bmatrix} \mathbf{G}_\Lambda.$$

It follows from Theorem 3.3 that $\Lambda/\Lambda' \cong (\mathbb{Z}[i]/\langle p\rangle)^k$. That is, the message space under this construction is $W = (\mathbb{Z}[i]/\langle p\rangle)^k$. In particular, the message rate $\mathsf{R}_{\mathsf{mes}} = \frac{k}{n}\log_2(p^2)$, since $\mathbb{Z}[i]/\langle p\rangle$ contains p^2 elements.

Note that the message space W can be viewed as a free $\mathbb{Z}[i]/\langle p\rangle$-module of rank k. In particular, W is a vector space if and only if the prime number p is a Gaussian prime, which is equivalent to saying that p is of the form $4j+3$.

To study the nominal coding gain $\gamma_c(\Lambda/\Lambda')$ as well as $K(\Lambda/\Lambda')$, we relate them to certain parameters of the linear code $\mathcal{C}$. To each codeword $\mathbf{c} = (c_1 + \langle p\rangle, \ldots, c_n + \langle p\rangle) \in \mathcal{C}$, there corresponds a coset $(c_1, \ldots, c_n) + p\mathbb{Z}^n$ whose minimum-norm coset leader, denoted by $\sigma^*(\mathbf{c})$, is given by

$$\sigma^*(\mathbf{c}) = (c_1 - \lfloor c_1/p\rceil \times p, \ldots, c_n - \lfloor c_n/p\rceil \times p),$$

where $\lfloor x\rceil$ is a rounding operation. The Euclidean weight $w_E(\mathbf{c})$ of $\mathbf{c}$ can then be defined as the squared Euclidean norm of $\sigma^*(\mathbf{c})$, that is, $w_E(\mathbf{c}) = \|\sigma^*(\mathbf{c})\|^2$. Thus, for example, when $\mathbf{c} = (1 + \langle 5\rangle, 3 + \langle 5\rangle)$, $\sigma^*(\mathbf{c}) = (1, -2)$. Clearly, the Euclidean weight of $\mathbf{c}$ is equivalent to the 2-norm of $\mathbf{c}$ defined in [57]. Let

$w_E^{\min}(\mathcal{C})$ be the minimum Euclidean weight of nonzero codewords in $\mathcal{C}$, i.e.,

$$w_E^{\min}(\mathcal{C}) = \min\{w_E(\mathbf{c}) : \mathbf{c} \neq \mathbf{0},\ \mathbf{c} \in \mathcal{C}\}.$$

Let $A(w_E^{\min})$ be the number of codewords in $\mathcal{C}$ with minimum Euclidean weight $w_E^{\min}(\mathcal{C})$. Then we have the following result.

Proposition 3.2. *Let $\mathcal{C}$ be a linear code over $\mathbb{Z}/\langle p\rangle$ and let $\Lambda \supseteq \Lambda'$ be a pair of nested lattices constructed from $\mathcal{C}$. Then*

$$\gamma_c(\Lambda/\Lambda') = \frac{w_E^{\min}(\mathcal{C})}{p^{2(1-k/n)}}$$

and

$$K(\Lambda/\Lambda') = \begin{cases} 2A\left(w_E^{\min}(\mathcal{C})\right) 2^{w_E^{\min}(\mathcal{C})}, & \text{when } p = 2, \\ 2A\left(w_E^{\min}(\mathcal{C})\right), & \text{when } p > 2. \end{cases}$$

The proof is in Appendix B.

Proposition 3.2 suggests that optimizing the nominal coding gain $\gamma_c(\Lambda/\Lambda')$ amounts to maximizing the minimum Euclidean weight $w_E^{\min}(\mathcal{C})$ of $\mathcal{C}$, and that optimizing $K(\Lambda/\Lambda')$ amounts to minimizing $A(w_E^{\min})$.

Nested Lattices via Complex Construction A

Let π be a prime in T. Let $\mathcal{C}$ be a linear code of length n over $T/\langle\pi\rangle$. Without loss of generality, we may assume the linear code $\mathcal{C}$ is systematic. Define a "complex Construction A lattice" [36] as

$$\Lambda \triangleq \{\boldsymbol{\lambda} \in T^n : \sigma(\boldsymbol{\lambda}) \in \mathcal{C}\},$$

where $\sigma : T^n \to (T/\langle\pi\rangle)^n$ is the natural projection map. Define

$$\Lambda' \triangleq \{\pi\mathbf{r} : \mathbf{r} \in T^n\}.$$

It is easy to see Λ' is a sublattice of Λ. Hence, we obtain a pair of nested lattices $\Lambda \supseteq \Lambda'$ from the linear code $\mathcal{C}$.

To study the message space induced by Λ/Λ', we specify two generator matrices satisfying the relation

(3.7). It is well-known that Λ has a generator matrix $\mathbf{G}_\Lambda$ given by

$$\mathbf{G}_\Lambda = \begin{bmatrix} \mathbf{I}_k & \mathbf{B}_{k\times(n-k)} \\ \mathbf{0}_{(n-k)\times k} & \pi\mathbf{I}_{n-k} \end{bmatrix},$$

and that Λ' has a generator matrix $\mathbf{G}_{\Lambda'}$ given by

$$\mathbf{G}_{\Lambda'} = \begin{bmatrix} \pi\mathbf{I}_k & \pi\mathbf{B}_{k\times(n-k)} \\ \mathbf{0}_{(n-k)\times k} & \pi\mathbf{I}_{n-k} \end{bmatrix}.$$

These two generator matrices satisfy

$$\mathbf{G}_{\Lambda'} = \begin{bmatrix} \pi\mathbf{I}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-k} \end{bmatrix} \mathbf{G}_\Lambda.$$

Hence, we have $\Lambda/\Lambda' \cong (T/\langle\pi\rangle)^k$. That is, the message space under this construction is $W = (T/\langle\pi\rangle)^k$. Since π is a prime in T, $T/\langle\pi\rangle$ is a finite field and W is a vector space of dimension k. Thus, this construction is preferable to the previous construction, if the message space is required to be a vector space. For instance, if $T = \mathbb{Z}[\omega]$ and $\pi = 2$, then the message space W is a vector space over $\mathbb{F}_4$. This never happens under the previous construction, since 2 is not a prime in $\mathbb{Z}[i]$.

To study the nominal coding gain $\gamma_c(\Lambda/\Lambda')$ as well as $K(\Lambda/\Lambda')$, we again relate them to the parameters of the linear code $\mathcal{C}$ with a particular focus on $T = \mathbb{Z}[i]$ (due to hypercube shaping). The definition of the minimum Euclidean weight $w_E^{\min}(\mathcal{C})$ is the same as the previous definition, except for the fact that the minimum-norm coset leader $\sigma^*(\mathbf{c})$ is given by

$$\sigma^*(\mathbf{c}) = (c_1 - \lfloor c_1/\pi \rceil \times \pi, \ldots, c_n - \lfloor c_n/\pi \rceil \times \pi),$$

where the rounding operation $\lfloor x \rceil$ sends $x \in \mathbb{C}$ to the closest Gaussian integer in the Euclidean distance.

Proposition 3.3. *Let $\mathcal{C}$ be a linear code over $\mathbb{Z}[i]/\langle\pi\rangle$ and let $\Lambda \supseteq \Lambda'$ be a pair of nested lattices constructed from $\mathcal{C}$. Then*

$$\gamma_c(\Lambda/\Lambda') = \frac{w_E^{\min}(\mathcal{C})}{|\pi|^{2(1-k/n)}}$$

and

$$K(\Lambda/\Lambda') = \begin{cases} A\left(w_E^{\min}(\mathcal{C})\right) 4^{w_E^{\min}(\mathcal{C})}, & \text{when } |\pi|^2 = 2, \\ A\left(w_E^{\min}(\mathcal{C})\right), & \text{otherwise.} \end{cases}$$

The proof is in Appendix C.

Nested Lattices via Construction D

Let $p > 0$ be a prime in $\mathbb{Z}$. Let $\mathcal{C}_1 \subseteq \cdots \subseteq \mathcal{C}_s$ be nested linear codes of length n over $\mathbb{Z}/\langle p\rangle$, where $\mathcal{C}_i$ has parameters $[n, k_i]$ for $i = 1, \ldots, s$. As shown in [36], there exists a basis $\{\mathbf{g}_1, \ldots, \mathbf{g}_n\}$ for the vector space $(\mathbb{Z}/\langle p\rangle)^n$ such that

1. $\mathbf{g}_1, \ldots, \mathbf{g}_{k_i}$ span $\mathcal{C}_i$ for $i = 1, \ldots, s$; and
2. if $\mathbf{G}$ denotes the matrix with rows $\mathbf{g}_1, \ldots, \mathbf{g}_n$, some permutation of the rows of $\mathbf{G}$ gives an upper triangular matrix with diagonal elements equal to $1 + \langle p\rangle$.

(In fact, $\mathbf{G}$ can be constructed by applying Gaussian elimination to the generator matrices of the nested linear codes iteratively.)

Using the nested linear codes $\{\mathcal{C}_i, 1 \le i \le s\}$, we define a "real Construction D lattice" [36] as

$$\Lambda_r \triangleq \left\{ \sum_{i=1}^{s} \sum_{j=1}^{k_i} p^{i-1} \beta_{ij} \tilde{\sigma}(\mathbf{g}_j) : \beta_{ij} \in \{0, \ldots, p-1\} \right\} + p^s \mathbb{Z}^n \tag{3.11}$$

where $\tilde{\sigma}$ is the natural embedding map from $(\mathbb{Z}/\langle p\rangle)^n$ to $\{0, \ldots, p-1\}^n$. (For completeness, we will show in Appendix D that Λ_r is indeed a lattice; we will also give an explicit generator matrix for Λ_r.)

Note that the lattice defined by $\Lambda'_r \triangleq \{p^s \mathbf{r} : \mathbf{r} \in \mathbb{Z}^n\}$ is a sublattice of Λ_r. Hence, we obtain a pair of nested $\mathbb{Z}$-lattices $\Lambda_r \supseteq \Lambda'_r$ from the nested linear codes $\{\mathcal{C}_i, 1 \le i \le s\}$.

Next, we lift this pair of nested $\mathbb{Z}$-lattices to a pair of nested $\mathbb{Z}[i]$-lattices. That is, we set $\Lambda = \Lambda_r + i\Lambda_r$ and $\Lambda' = \Lambda'_r + i\Lambda'_r$. In this way, we obtain a pair of nested $\mathbb{Z}[i]$-lattices $\Lambda \supseteq \Lambda'$. In Appendix E, we will show that there exist two generator matrices $\mathbf{G}_\Lambda$ and $\mathbf{G}_{\Lambda'}$ satisfying

$$\mathbf{G}_{\Lambda'} = \mathrm{diag}(\underbrace{p^s, \ldots, p^s}_{k_1}, \underbrace{p^{s-1}, \ldots, p^{s-1}}_{k_2 - k_1}, \ldots, \underbrace{1, \ldots, 1}_{n - k_s}) \mathbf{G}_\Lambda. \tag{3.12}$$

It follows from Theorem 3.3 that

$$\Lambda/\Lambda' \cong (\mathbb{Z}[i]/\langle p^s\rangle)^{k_1} \times \cdots \times (\mathbb{Z}[i]/\langle p\rangle)^{k_s - k_{s-1}}.$$

In particular, the message rate $\mathsf{R}_{\mathsf{mes}} = \frac{\sum_i k_i}{n} \log_2(p^2)$. When $s = 1$, this construction is reduced to the first construction. Although this construction induces a more complicated message space, it is able to

produce pairs of nested lattices with higher nominal coding gains, as shown in the following result.

Proposition 3.4. *Let $\mathcal{C}_1 \subseteq \cdots \subseteq \mathcal{C}_s$ be nested linear codes of length n over $\mathbb{Z}/\langle p\rangle$ and let $\Lambda \supseteq \Lambda'$ be a pair of nested lattices constructed from $\{\mathcal{C}_i\}$. Then $\gamma_c(\Lambda/\Lambda')$ is lower bounded by*

$$\gamma_c(\Lambda/\Lambda') \geq \frac{\min_{1\leq i\leq s}\{p^{2(i-1)} w_E^{\min}(\mathcal{C}_i)\}}{p^{2(s-\sum_{i=1}^{a} k_i/n)}},$$

and $K(\Lambda/\Lambda')$ is upper bounded by

$$K(\Lambda/\Lambda') \leq \begin{cases} 2\sum_{i=1}^{s} 2^{A_i} A_i, & \text{when } p = 2 \\ 2\sum_{i=1}^{s} A_i, & \text{when } p > 2 \end{cases}$$

where A_i is the number of codewords in $\mathcal{C}_i$ with minimum Euclidean weight $w_E^{\min}(\mathcal{C}_i)$.

The proof is given in Appendix F.

Now we will apply Propositions 3.2 and 3.4 to show the advantage of pairs of nested lattices constructed via Construction D. Let $\Lambda_{\mathrm{A}} \supseteq \Lambda'_{\mathrm{A}}$ be a pair of nested lattices constructed from a linear $[n, k]$ code $\mathcal{C}$ (over $\mathbb{Z}/\langle p\rangle$) via Construction A. Then by Proposition 3.2, $\gamma_c(\Lambda_{\mathrm{A}}/\Lambda'_{\mathrm{A}}) = w_E^{\min}(\mathcal{C})/p^{2(1-k/n)}$. Suppose that the linear code $\mathcal{C}$ has an $[n, k']$ subcode $\mathcal{C}'$ with $w_E^{\min}(\mathcal{C}') \geq p^2 w_E^{\min}(\mathcal{C})$. Let $\Lambda_{\mathrm{D}} \supseteq \Lambda'_{\mathrm{D}}$ be a pair of nested lattices constructed from $\mathcal{C}$ and $\mathcal{C}'$ via Construction D. Then by Proposition 3.4,

$$\begin{aligned} \gamma_c(\Lambda_{\mathrm{D}}/\Lambda'_{\mathrm{D}}) &\geq \frac{p^2 w_E^{\min}(\mathcal{C})}{p^{2(2-(k+k')/n)}} \\ &= \frac{w_E^{\min}(\mathcal{C})}{p^{2(1-(k+k')/n)}} \\ &> \gamma_c(\Lambda_{\mathrm{A}}/\Lambda'_{\mathrm{A}}). \end{aligned}$$

In other words, given a pair of nested lattices via Construction A, there exists a pair of nested lattices via Construction D with higher nominal coding gain if the linear code $\mathcal{C}$ has a subcode $\mathcal{C}'$ with $w_E^{\min}(\mathcal{C}') \geq p^2 w_E^{\min}(\mathcal{C})$.

3.6.2 Design Examples

We present four design examples to illustrate the design tools developed in Sec. 3.6.1. All of our design examples feature short packet length and reasonable decoding complexity, since the purpose of this chapter is to demonstrate the potential of LNC schemes in practical settings. (A more elaborate scheme, based on signal codes [58], is described in [59].)

Example 3.7. *Consider a rate-*$1/2$ *terminated (feed-forward) convolutional code over* $\mathbb{Z}/\langle 2\rangle$*. Suppose the input sequence* $u(D)$ *is a polynomial of degree less than* μ*. Then this terminated convolutional code can be regarded as a* $[2(\mu+\nu),\mu]$ *linear block code* $\mathcal{C}$*. Using the method based on Construction A, we obtain a pair of nested lattices* $\Lambda \supseteq \Lambda'$ *with nominal coding gain*

$$\gamma_c(\Lambda/\Lambda') = \frac{d_{\min}(\mathcal{C})}{2^{2(1-\mu/(2(\mu+\nu)))}} \approx \frac{d_{\min}(\mathcal{C})}{2}$$

when $\mu \gg \nu$*, where* $d_{\min}(\mathcal{C})$ *is the minimum distance of* $\mathcal{C}$*. Table 3.1 lists three convolutional codes widely used in practice. The nominal coding gains of the corresponding nested lattices are also provided.*

Table 3.1: Rate-$1/2$ convolutional codes and corresponding nominal coding gains.

ν	$\mathbf{g}(D)$	$\gamma_c(\Lambda/\Lambda')$	Application
4	[31, 33]	3.5 (5.44 dB)	GSM
6	[155, 117]	5 (6.99 dB)	802.11a
8	[657, 435]	6 (7.78 dB)	IS-95

Note that the encoder state-space sizes are 16*,* 64*,* 256*, respectively. Note also that the lattice decoder* $\mathcal{D}_\Lambda$ *can be implemented through a modified Viterbi decoder as discussed in Appendix G. Thus, this example demonstrates that a nominal coding gain of* 5 *to* 7 *dB can be easily obtained with reasonable decoding complexity.*

Example 3.8. *Consider a rate-*$1/2$ *terminated (feed-forward) convolutional code over* $\mathbb{Z}[i]/\langle 3\rangle$ *with* ν *memory elements. Suppose the input sequence* $u(D)$ *is a polynomial of degree less than* μ*. Then this terminated convolutional code can be regarded as a* $[2(\mu+\nu),\mu]$ *linear block code* $\mathcal{C}$*. Using the method based on complex Construction A, we obtain a pair of nested lattices* $\Lambda \supseteq \Lambda'$*.*

Note that the minimum Euclidean weight $w_E^{\min}(\mathcal{C})$ *of* $\mathcal{C}$ *can be bounded as*

$$w_E^{\min}(\mathcal{C}) \le 3(1+\nu),$$

for all rate-$1/2$ *terminated (feed-forward) convolutional codes over* $\mathbb{Z}[i]/\langle 3\rangle$*. This upper bound can be verified by considering the input sequence* $u(D) = 1$*. Hence, the nominal coding gain* $\gamma_c(\Lambda/\Lambda')$ *satisfies*

$$\gamma_c(\Lambda/\Lambda') \le 1+\nu.$$

When $\nu = 1, 2$ *and* $\mu \gg \nu$*, this upper bound can be asymptotically achieved by polynomial convolutional encoders shown in Table 3.2.*

Our next example illustrates how to use our design tools to improve an existing construction presented

Table 3.2: Polynomial convolutional encoders that asymptotically achieve the upper bound.

ν	$\mathbf{g}(D)$	$\gamma_c(\Lambda/\Lambda')$
1	$[1+(1+i)D,\ (1+i)+D]$	2 (3 dB)
2	$[1+D+(1+i)D^2,\ (1+i)+(1-i)D+D^2]$	3 (4.77 dB)

in [60].

Example 3.9. *Consider nested linear codes $\mathcal{C}_1 \subseteq \mathcal{C}_2$ of length n over $\mathbb{Z}/\langle 2\rangle$, where $\mathcal{C}_1$ is an $[n, k_1, d_1]$ code with $d_1 \geq 4$ and $\mathcal{C}_2$ is the $[n, n]$ trivial code. Using the method based on Construction D, we obtain a pair of nested lattices $\Lambda \supseteq \Lambda'$.*

In this case, we will show that the nominal coding gain $\gamma_c(\Lambda/\Lambda') = 4/4^{(1-k_1/n)}$. On the one hand, by Proposition 3.4,

$$\gamma_c(\Lambda/\Lambda') \geq \frac{\min\{w_H^{\min}(\mathcal{C}_1), 4w_H^{\min}(\mathcal{C}_2)\}}{4^{(2-\sum_{i=1}^{2} k_i/n)}} = 4/4^{(1-k_1/n)}.$$

On the other hand, by definition,

$$\begin{aligned}\gamma_c(\Lambda/\Lambda') &= d^2(\Lambda/\Lambda')/V(\Lambda')^{1/n} \\ &= d^2(\Lambda/\Lambda')/4^{(2-\sum_{i=1}^{2} k_i/n)} \qquad (3.13)\\ &\leq 4/4^{(1-k_1/n)} \qquad (3.14)\end{aligned}$$

where (3.13) follows from the facts that $V(\Lambda') = V(\Lambda)4^{k_1+k_2}$ and $V(\Lambda') = 4^{2n}$; (3.14) follows from the fact that $(2, 0, \ldots, 0)$ is a lattice point in Λ but not in Λ'.

Finally, in Table 3.3 we list several candidates for $\mathcal{C}_1$ as well as their corresponding nominal coding gains. These candidates are all extended Hamming codes with $d_1 = 4$.

Table 3.3: Several extended Hamming codes and corresponding nominal coding gains.

n	k	$\gamma_c(\Lambda/\Lambda')$
32	26	3.08 (4.89 dB)
64	57	3.44 (5.36 dB)
128	120	3.67 (5.64 dB)
256	247	3.81 (5.81 dB)

We note that Ordentlich-Erez's construction in [60] can be regarded as a special case of Example 3.9. In their construction, $\mathcal{C}_1$ is chosen as a rate 5/6 cyclic LDPC code of length 64800. Example 3.9 suggests that their nominal coding gain is $4/4^{1/6}$ (5.02 dB) with message rate $2(1+5/6) \approx 3.67$. Example 3.9 also suggests that there are many ways to improve the nominal coding gain. For example, when $\mathcal{C}_1$ is chosen as a $[256, 247]$ extended Hamming code, the nominal coding gain is 5.81 dB with message rate $2(1+\frac{247}{256}) \approx 3.93$.

Our fourth example illustrates how to design high-coding-gain nested lattices based on turbo lattices [61].

Example 3.10. *Consider nested Turbo codes $\mathcal{C}_1 \subseteq \mathcal{C}_2$ over $\mathbb{Z}/\langle 2\rangle$. As shown in [61], $\mathcal{C}_1$ can be a rate-$1/3$ Turbo code with $d_1 = 28$ and $\mathcal{C}_2$ can be a rate-$1/2$ Turbo code with $d_2 = 13$. Using the method via Construction D, we obtain a pair of nested lattices $\Lambda \supseteq \Lambda'$. In this case, by Proposition 3.4,*

$$\gamma_c(\Lambda/\Lambda') \geq \frac{\min\{d_1, 4d_2\}}{4^{(2-\sum_{i=1}^{2} k_i/n)}} = 28/4^{(2-1/2-1/3)} = 7.45 \ dB.$$

The message rate is given by $R_{mes} = 5/3 \approx 1.67$.

Finally, some other design examples of high-performance nested lattice codes, which are of a similar spirit, can be found, e.g., in [45–47, 59, 62], Also, similar methods of designing practical compute-and-forward have been recently proposed. See, e.g., [21, 43, 63].

3.7 Decoding Multiple Linear Combinations

In this section, we consider the problem when a receiver has the freedom to choose coefficient vectors. For ease of presentation, we mainly focus on the case of complex Construction A in which the message space is a vector space over $T/\langle\pi\rangle$. The main result of this section is that, under separate decoding, the problem of decoding multiple linear combinations is related to the *shortest independent vectors problem* [40], and can be solved through some existing methods. The motivation of studying this problem mainly comes from successive C&F [64] as well as its applications [22].

In general, upon deciding the coefficient vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$, the receiver can perform joint decoding or separate decoding to recover the linear combinations $\mathbf{u}_i = \mathbf{a}_i\mathbf{W}$. Here, we confine our attention to separate decoding in which each linear combination $\mathbf{u}_i = \mathbf{a}_i\mathbf{W}$ is decoded independently through the use of $\mathcal{D}(\mathbf{y} \mid \mathbf{h}, \mathbf{a}_i)$. In this case, the union bound estimate on the decoding error for each $\mathbf{a}_i$ is

$$P_e(\mathbf{h}, \mathbf{a}_i) \lessapprox K(\Lambda/\Lambda') \exp\left(-\frac{d^2(\Lambda/\Lambda')}{4N_0 \mathbf{a}_i \mathbf{M} \mathbf{a}_i^{\mathsf{H}}}\right).$$

To optimize the above union bound estimates, the coefficient vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ should be chosen such that each $\mathbf{a}_i\mathbf{M}\mathbf{a}_i^{\mathsf{H}}$ is made as small as possible under the constraint that $\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_m$ are *linearly independent* over $T/\langle\pi\rangle$, where $\bar{\mathbf{a}}_i = \sigma(\mathbf{a}_i)$ is the natural projection of $\mathbf{a}_i$ (from T to $T/\langle\pi\rangle$). Clearly, this constraint ensures that every recovered linear combination $\mathbf{u}_i$ is useful over $T/\langle\pi\rangle$.

We say a solution $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ is *feasible* if $\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_m$ are linearly independent over $T/\langle\pi\rangle$. Since

each $\bar{\mathbf{a}}_i$ is of dimension L, we assume that $m \leq L$ because otherwise no feasible solution exists.

In the sequel, we will show that there exists a feasible solution that *simultaneously* optimizes each $\mathbf{a}_i\mathbf{M}\mathbf{a}_i^{\mathsf{H}}$. We call such feasible solutions *dominant solutions*. Formally, let $\mathbf{M} = \mathbf{L}\mathbf{L}^{\mathsf{H}}$ be the Cholesky decomposition of $\mathbf{M}$, where $\mathbf{L}$ is some lower triangular matrix. (The existence of $\mathbf{L}$ comes from the fact that $\mathbf{M}$ is Hermitian and positive-definite.) Clearly, $\mathbf{a}\mathbf{M}\mathbf{a}^{\mathsf{H}} = \|\mathbf{a}\mathbf{L}\|^2$.

Definition 3.1 (Dominant Solutions)**.** *A feasible solution* $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ *(with* $\|\mathbf{a}_1\mathbf{L}\| \leq \ldots \leq \|\mathbf{a}_m\mathbf{L}\|$*) is called a* dominant solution *if for any feasible solution* $\mathbf{a}'_1, \ldots, \mathbf{a}'_m$ *(with* $\|\mathbf{a}'_1\mathbf{L}\| \leq \ldots \leq \|\mathbf{a}'_m\mathbf{L}\|$*), the following inequalities hold*

$$\|\mathbf{a}_i\mathbf{L}\| \leq \|\mathbf{a}'_i\mathbf{L}\|, \ i = 1, \ldots, m.$$

Although the dominant solutions seem to be a natural concept, the existence of them is not immediate from the definition, and a separate argument is needed.

Theorem 3.5. *A feasible solution* $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ *defined by*

$$\begin{aligned}
\mathbf{a}_1 &= \arg\min \{\|\mathbf{a}\mathbf{L}\| \mid \bar{\mathbf{a}} \text{ is nonzero}\} \\
\mathbf{a}_2 &= \arg\min \{\|\mathbf{a}\mathbf{L}\| \mid \bar{\mathbf{a}}, \bar{\mathbf{a}}_1 \text{ are linearly independent}\} \\
&\vdots \\
\mathbf{a}_m &= \arg\min \{\|\mathbf{a}\mathbf{L}\| \mid \bar{\mathbf{a}}, \bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_{m-1} \text{ are linearly ind.}\}
\end{aligned}$$

always exists, and is a dominant solution.

The proof is given in Appendix H.

We now propose a three-step method of finding a dominant solution. In the first step, we construct a ball $\mathcal{B}(\rho) = \{\mathbf{x} \in \mathbb{C}^L \mid \|\mathbf{x}\| \leq \rho\}$ that contains m lattice points $\mathbf{v}_1\mathbf{L}, \ldots, \mathbf{v}_m\mathbf{L}$ such that $\bar{\mathbf{v}}_1, \ldots, \bar{\mathbf{v}}_m$ are linearly independent, where $\bar{\mathbf{v}}_i = \sigma(\mathbf{v}_i)$ is the natural projection of $\mathbf{v}_i$. In the second step, we order all lattice points within $\mathcal{B}(\rho)$ based on their lengths, producing an ordered set $\mathcal{S}_\rho$ with $\|\mathbf{v}_1\mathbf{L}\| \leq \|\mathbf{v}_2\mathbf{L}\| \leq \cdots \leq \|\mathbf{v}_{|\mathcal{S}_\rho|}\mathbf{L}\|$. Finally, we find a dominant solution $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ by using a greedy search algorithm given as Algorithm 1.

The correctness of our proposed method follows immediately from Theorem 3.5. Our proposed method is in the spirit of sphere-decoding algorithms, since sphere-decoding algorithms also enumerate all lattice points within a ball centered at a given vector. The selection of the radius ρ plays an important role here, just as it does for sphere-decoding algorithms. If ρ is too large, then the second step may incur excessive computations. If ρ is too small, then the first step may fail to construct a ball that contains

Algorithm 1 *Greedy Search for Dominant Solution*

Input: An ordered set $\mathcal{S}_\rho = \{\mathbf{v}_1\mathbf{L}, \mathbf{v}_2\mathbf{L}, \ldots, \mathbf{v}_{|\mathcal{S}_\rho|}\mathbf{L}\}$ with $\|\mathbf{v}_1\mathbf{L}\| \leq \|\mathbf{v}_2\mathbf{L}\| \leq \cdots \leq \|\mathbf{v}_{|\mathcal{S}_\rho|}\mathbf{L}\|$.
Output: An optimal solution $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$.

1. Set $\mathbf{a}_1 = \mathbf{v}_1$. Set $i = 1$ and $j = 1$.
2. **while** $i < |\mathcal{S}_b|$ and $j < m$ **do**
3. Set $i = i + 1$.
4. **if** $\bar{\mathbf{v}}_i, \bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_j$ are linearly independent **then**
5. Set $j = j + 1$. Set $\mathbf{a}_j = \mathbf{v}_i$.
6. **end if**
7. **end while**

m linearly independent $\bar{\mathbf{v}}_1, \ldots, \bar{\mathbf{v}}_m$.

In practice, lattice-reduction algorithms [65] may be used to determine an appropriate radius ρ, as shown in the following proposition.

Proposition 3.5. *Let* $\{\mathbf{b}_1, \ldots, \mathbf{b}_L\}$ *be a* reduced basis *[65] for* $\mathbf{L}$. *If* ρ *is set to be* $\|\mathbf{b}_m\|$, *then the set* $\mathcal{S}_\rho$ *contains at least* m *lattice points* $\mathbf{v}_1\mathbf{L}, \ldots, \mathbf{v}_m\mathbf{L}$ *such that* $\bar{\mathbf{v}}_1, \ldots, \bar{\mathbf{v}}_m$ *are linearly independent.*

Proof. Let $\mathbf{v}_i = \mathbf{b}_i\mathbf{L}^{-1}$ for $i = 1, \ldots, L$. Let $\mathbf{V}$ be an $L \times L$ matrix with $\mathbf{v}_i$ as its ith row. Since $\{\mathbf{b}_1, \ldots, \mathbf{b}_L\}$ is a reduced basis, it follows that the matrix $\mathbf{V}$ is invertible. In particular, $\bar{\mathbf{v}}_1, \ldots, \bar{\mathbf{v}}_m$ are linearly independent for all integers $m \leq L$. □

There are many existing lattice-reduction algorithms in the literature. Among them, the Lenstra-Lenstra-Lovász (LLL) algorithm [66] is of particular importance. Moreover, the LLL algorithm has been extended from real lattices to complex lattices over Euclidean domains [67, 68]. Since $\mathbb{Z}[i]$ and $\mathbb{Z}[\omega]$ are special cases of Euclidean domains, the extended LLL algorithm can be used to handle the cases of $T = \mathbb{Z}[i]$ and $T = \mathbb{Z}[\omega]$.

Interestingly, when L is small, some efficient lattice-reduction algorithms can directly output dominant solutions. Such algorithms, which are generalizations of Gauss' algorithm (see, e.g., [69]), are described in [70, 71].

3.8 Simulation Results

As described in Section 6.1, there are many potential application scenarios for LNC, the most promising of which may involve multicasting from one (or more) sources to multiple destinations via a wireless relay network. Since we wish to avoid introducing higher-layer issues (e.g., scheduling), in this chapter, we focus here on a two-transmitter, single receiver multiple-access configuration, which may be regarded as a building block component of a more complicated and realistic network application. In particular, we focus on the following three scenarios:

1. The channel gains are fixed; the receiver chooses a single linear function.

2. The channel gains are Rayleigh faded; the receiver chooses a single linear function.

3. The channel gains are Rayleigh faded; the receiver chooses two linear functions.

In each scenario, we evaluate the performance of four LNC schemes: the Nazer-Gastpar scheme, two LNC schemes proposed in Example 3.7 with $\nu = 4$ and $\nu = 6$, and the baseline LNC scheme using uncoded BPSK. Since we are interested in LNC schemes with relatively short packet lengths, each transmitted signal consists of 800 complex symbols in our simulations.

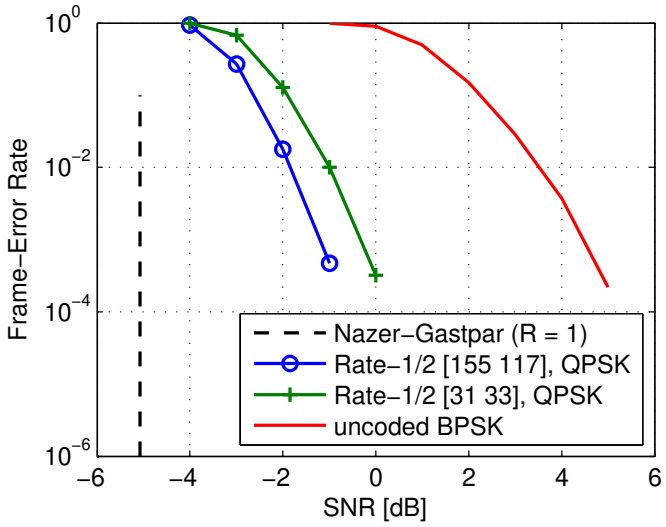


Figure 3.6: Error performance of four LNC schemes in Scenario 1.

3.8.1 Scenario 1 (Fixed Channel Gains; Single Coefficient Vector)

Fig. 3.6 depicts the frame-error rates of four LNC schemes as a function of SNR. Here, the channel-gain vector $\mathbf{h}$ is set to $\mathbf{h} = [-1.17 + 2.15i \;\; 1.25 - 1.63i]$. Nevertheless, as we have shown in Sec. 3.6, the results are not particularly sensitive to the choice for $\mathbf{h}$; similar results are achieved for other fixed choices for $\mathbf{h}$. For the two LNC schemes proposed in Example 3.7, the parameter $\mu + \nu$ is set to 400 and the corresponding message rates are $\frac{396}{400}$ ($\nu = 4$) and $\frac{394}{400}$ ($\nu = 6$), respectively. For the Nazer-Gastpar scheme, the message rate is set to 1, which is quite close to the previous two message rates. The decoding rule for the Nazer-Gastpar scheme is as follows: a frame error occurs if and only if $\log_2(\mathsf{SNR}\,/\mathbf{a}\mathbf{M}\mathbf{a}^{\mathsf{H}}) \leq 1$, where $\mathbf{a}$ is the single coefficient vector. From Fig. 3.6, we observe that the gap to the Nazer-Gastpar scheme is around 3 dB at an error-rate of 1%. We also observe that the second LNC scheme (with $\nu = 6$) outperforms the first LNC scheme (with $\nu = 4$) by about 1 dB.

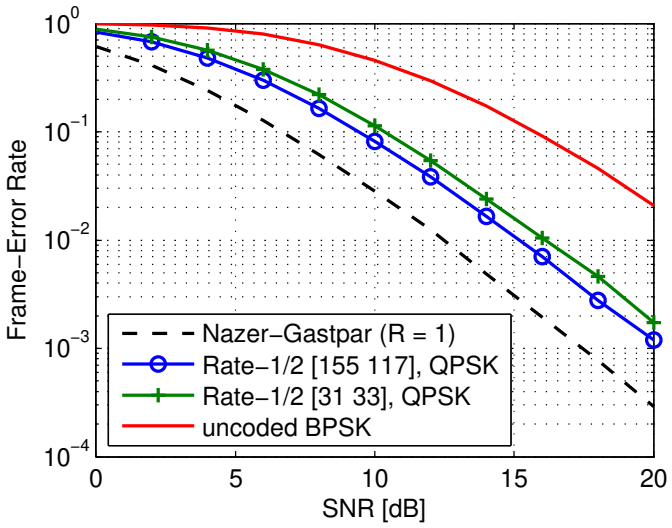


Figure 3.7: Error performance of four LNC schemes in Scenario 2.

3.8.2 Scenario 2 (Rayleigh-faded Channel Gains; Single Coefficient Vector)

Fig. 3.7 shows the frame-error rates of four LNC schemes as a function of SNR. The setup is the same as in Scenario 1, except that the coefficient vector $\mathbf{a}$ changes with $\mathbf{h}$. As seen in Fig. 3.7, the gap to the Nazer-Gastpar scheme is around 2.5 dB at an error-rate of 1%. Once again, the second LNC scheme (with $\nu = 6$) outperforms the first LNC scheme (with $\nu = 4$) by about 1 dB, which agrees with our theoretical results.

3.8.3 Scenario 3 (Rayleigh-faded Channel Gains; Two Coefficient Vectors)

Fig. 3.8 depicts the failure rates of four LNC schemes as a function of SNR. Here the two coefficient vectors are chosen by using the lattice-reduction algorithm proposed in [70]. The configurations for the four LNC schemes are precisely the same as those in Fig. 3.7. We say a failure occurs if the receiver fails to decode both linear functions. From Fig. 3.8, we observe similar trends of error rates as in Fig. 3.7. For instance, the gap to the Nazer-Gastpar scheme is around 3 dB at an error-rate of 1%.

3.9 Summary

In this chapter, the problem of constructing LNC schemes via finite-dimensional nested lattices has been studied. A generic LNC scheme has been defined based on an arbitrary pair of nested lattices. The message space of the generic scheme is a finite module in general, whose structure may be analyzed using the Smith normal form theorem. These results not only give rise to a convenient characterization of the

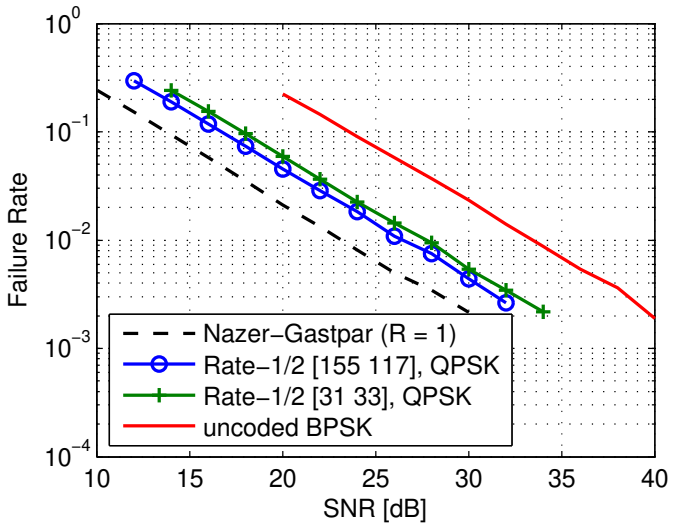


Figure 3.8: Error performance of four LNC schemes in Scenario 3.

message space of the Nazer-Gastpar scheme, but also lead to several generalized constructions of LNC schemes. All of these constructions are compatible with header-based random linear network coding.

An estimate of the error probability for hypercube-shaped LNC schemes has been derived, showing that the pair of nested lattices $\Lambda \supseteq \Lambda'$ should be designed such that $d(\Lambda/\Lambda')$ is maximized and $K(\Lambda/\Lambda')$ is minimized. These criteria lead to several specific methods for optimizing nested lattices. In particular, the nominal coding gain for pairs of nested lattices has been introduced, which serves as an important figure of merit for comparing various LNC schemes. In addition, several concrete examples of practical LNC schemes have been provided, showing that a nominal coding gain of 3 to 7.5 dB is easily obtained under reasonable decoding complexity and short packet length. Finally, the problem of choosing multiple coefficient vectors is discussed, which is connected to some well-studied lattice problems, such as the shortest independent vectors problem and the lattice reduction problem.

We believe that there is still much work to be done in this area. One direction for follow-up work would be the design and analysis of higher-layer scheduling algorithms for LNC schemes. Another direction would be the study of more general shaping methods beyond hypercube shaping. A particular example along this direction is given in [46]. A third direction would be the construction of more powerful LNC schemes, which has been partially explored in several recent papers, e.g., [45, 47, 59, 62]. We believe that the algebraic framework given in this chapter can serve as a good basis for these developments.

Chapter 4

Blind Compute-and-Forward

This chapter studies the feasibility of eliminating the need for channel state information (CSI) in C&F. Conventional C&F schemes usually require CSI at the receivers so that an "optimal" scaling factor can be computed for the purposes of decoding. In this chapter, a blind C&F scheme—i.e., one not requiring CSI—is developed. Rather than attempting to compute the optimal scaling factor, this new scheme seeks one or more "good" scalars, i.e., scalars which allow correct decoding despite possibly being sub-optimal. The region of all such good scalars is characterized. To find a good scalar, a computationally efficient scheme is proposed, which involves three key components: error-detection, a hierarchically organized list, as well as a use of the Smoothing Lemma from lattice theory. Simulation results show that this blind C&F scheme achieves—for a certain class of nested lattice codes—almost the same throughput as its CSI-enabled counterpart, at the expense of, approximately, a two-fold increase in computational complexity in the high-throughput region.

4.1 Introduction

As we have seen in previous chapters, the basic idea behind C&F is to enable relay nodes to compute linear combinations of concurrently transmitted messages directly from interfering signals. In its simplest form, a relay node receives $\mathbf{y} = \sum_\ell h_\ell \mathbf{x}_\ell + \mathbf{z}$, where h_ℓ are channel gains, and $\mathbf{x}_\ell$ are points in a multidimensional lattice. Based on the fact that any integer combination of lattice points is again a lattice point, the relay node selects integer coefficients a_ℓ and a scalar α, and then attempts to decode

the lattice point $\sum_\ell a_\ell \mathbf{x}_\ell$ from the scaled signal

$$\begin{aligned} \alpha \mathbf{y} &= \sum_\ell \alpha h_\ell \mathbf{x}_\ell + \alpha \mathbf{z} \\ &= \sum_\ell a_\ell \mathbf{x}_\ell + \mathbf{n}_{\mathsf{eff}}, \end{aligned} \tag{4.1}$$

where $\mathbf{n}_{\mathsf{eff}} \triangleq \sum_\ell (\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha \mathbf{z}$ is the so-called "effective noise." The scalar α and integer coefficients a_ℓ are carefully chosen based on channel gains h_ℓ so that the effective noise is made (in some sense) small, as explained in Chapter 3. Hence, the "optimal" scalar α and integer coefficients a_ℓ critically depend on CSI.

In this chapter, we aim to eliminate the need for CSI in C&F. We consider the case when *no* CSI is available, hereafter called blind C&F. This is motivated by the fact that C&F is sensitive to channel estimation error [27] and the fact that the requirement of accurate CSI (even if only at the receivers) is quite demanding when the number of concurrent transmissions is large [28][1].

The basic idea of our approach to blind C&F is simple. Although the optimal scalar is nearly impossible to acquire without CSI, some "good" scalars (that allow correct decoding of linear combinations) can be obtained with a reasonable effort. In particular, when the underlying nested lattice codes of C&F are asymptotically-good (in the sense of [26]), we are able to characterize the region of all such good scalars, showing that it is bounded, symmetric, and consisting of a union of disks. Based on these properties, we propose a generic blind C&F scheme that finds a good scalar by "probing" a list of points until a codeword is found. Such a decision can always be made sufficiently reliable by concatenating an outer error-detection code.

To control the computational complexity of our blind C&F scheme, we propose three strategies that are complementary to each other. The first strategy creates a "smart" list of points that can be used to effectively control the number of probings. The second strategy reduces the complexity of the probing operation by using the Smoothing Lemma from lattice theory. The third strategy allows us to handle more general nested lattice codes without increasing the computational complexity. Using these strategies, a computationally efficient blind C&F scheme is obtained. This scheme achieves almost the same performance as coherent C&F (its CSI-enabled counterpart as described in Chapter 3) with a modest increase in computational complexity. In particular, our simulation results show that the complexity of this scheme is roughly twice the complexity of coherent C&F in the high-throughput region.

[1] A quantitative discussion will be provided in Sec. 4.2.1.

4.2 Blind Compute-and-Forward: General Framework

In this section, we present a general framework for blind C&F. At first glance, it seems very difficult to design a blind C&F scheme, since it is nearly impossible to acquire an optimal scalar α and coefficients a_ℓ without CSI. Our key observation is as follows: the receiver does not have to know an optimal scalar α and coefficients a_ℓ to ensure successful decoding; instead, it only needs to know some "good" scalars as well as $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda'$. As we will soon see, equipped with a good scalar and $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda'$, the receiver is always able to recover a linear combination $\sum_\ell a_\ell \mathbf{w}_\ell$ correctly[2].

4.2.1 Properties of good scalars

Here, we formally define good scalars and study their basic properties.

Definition 4.1. *A scalar α is said to be* good *if $\mathcal{Q}_\Lambda^{NN}(\mathbf{n}_{\mathsf{eff}}) \in \Lambda'$ for some coefficients $(a_1, \ldots, a_L) \in T^L \setminus \{\mathbf{0}\}$, and is said to be* bad *otherwise.*

Since the decoding is correct if and only if $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{n}_{\mathsf{eff}}) \in \Lambda'$, this justifies the above definition. Note that the effective noise $\mathbf{n}_{\mathsf{eff}}$ depends on the channel gains as well as $\mathbf{x}_\ell$'s and $\mathbf{z}$. Hence, whether a scalar α is good or bad relies on the channel gains and the realizations of $\mathbf{x}_\ell$'s and $\mathbf{z}$.

Definition 4.2. *The* good region *of scalars, denoted by $\mathcal{G}_s$, is the set of all good α's, i.e., $\mathcal{G}_s = \{\alpha \in \mathbb{C} : \alpha$ is good$\}$.*

The good region depends on the channel gains as well as the realizations of $\mathbf{x}_\ell$'s and $\mathbf{z}$. Although the good region is unknown to the receiver without CSI, it is still beneficial to understand some basic properties of the good region, which will play an important role in the design of our blind C&F schemes.

When the nested lattice code is asymptotically good (in the sense of [26]), the good region $\mathcal{G}_s$ has a number of interesting properties. Moreover, these properties still hold (or approximately hold) for commonly-used nested lattice codes.

We note that for asymptotically-good nested lattice codes, a scalar α is good if and only if the message rate R is less than the computation rate

$$R(\alpha, \mathbf{a}) \triangleq \log_2 \left(\frac{\mathsf{SNR}}{\mathsf{SNR}\,\|\alpha\mathbf{h} - \mathbf{a}\|^2 + |\alpha|^2} \right)$$

[2]Note that after recovering a linear combination $\sum_\ell a_\ell \mathbf{w}_\ell$, the receiver in our blind C&F scheme still does not know the coefficients a_ℓ. This is not an issue, since one can apply the techniques developed in [72, 73] to recover the original messages. These techniques extend non-coherent network coding from finite fields to finite rings, and will be discussed in Chapter 5.

for some $\mathbf{a} \in T^L \setminus \{\mathbf{0}\}$. This allows us to show that the good region $\mathcal{G}_s$ is bounded, symmetric, and consisting of a union of disks. (Note that the good region $\mathcal{G}_s$ depends on the message rate R.)

Proposition 4.1. *The good region $\mathcal{G}_s$ is bounded: every good scalar α satisfies $|\alpha|^2 < \mathsf{SNR}/2^R$.*

Proof: If α is good, then

$$\log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\,\|\alpha\mathbf{h}-\mathbf{a}\|^2+|\alpha|^2}\right) > R \text{ for some } \mathbf{a} \in T^L \setminus \{\mathbf{0}\}.$$

Since

$$\log_2\left(\frac{\mathsf{SNR}}{|\alpha|^2}\right) \geq \log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\,\|\alpha\mathbf{h}-\mathbf{a}\|^2+|\alpha|^2}\right),$$

we have

$$\log_2\left(\frac{\mathsf{SNR}}{|\alpha|^2}\right) > R.$$

Hence, every good α is bounded by $|\alpha|^2 < \mathsf{SNR}/2^R$. □

Proposition 4.2. *The good region $\mathcal{G}_s$ is* symmetric *with respect to rotations by some angle θ. The angle θ is determined by T: $\theta = 90^\circ$ when $T = \mathbb{Z}[i]$; $\theta = 60^\circ$ when $T = \mathbb{Z}[\omega]$.*

Proof: It suffices to show that if α is good, so is $e^{i\theta}\alpha$ for some angle θ. We need the following fact (from abstract algebra): Let T be a discrete subring of $\mathbb{C}$ forming a principle ideal domain. Let $\mathcal{U}$ be the set of all the units in T. Then $\mathcal{U} = \{e^{2\pi ki/n} : k = 0, 1, \ldots, n-1\}$ for some positive integer n. For example, when $T = \mathbb{Z}[i]$, the set of units $\mathcal{U} = \{e^0, e^{2\pi i/4}, e^{4\pi i/4}, e^{6\pi i/4}\}$; when $T = \mathbb{Z}[\omega]$, the set of units $\mathcal{U} = \{e^{2\pi ki/6} : k = 0, \ldots, 5\}$. Clearly, the units of T are also the roots of unity.

Now let us choose a unit $u = e^{2\pi i/n}$. If α is good, then

$$\log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\,\|\alpha\mathbf{h}-\mathbf{a}\|^2+|\alpha|^2}\right) > R \text{ for some } \mathbf{a} \in T^L \setminus \{\mathbf{0}\}.$$

It follows that

$$\log_2\left(\frac{\mathsf{SNR}}{\mathsf{SNR}\,\|u\alpha\mathbf{h}-u\mathbf{a}\|^2+|u\alpha|^2}\right) > R \text{ for } u\mathbf{a} \in T^L \setminus \{\mathbf{0}\}.$$

Therefore, $u\alpha$ is also good. In other words, the good region $\mathcal{G}_s$ is symmetric with respect to rotations by $360^\circ/n$, where $n = 4$ when $T = \mathbb{Z}[i]$ and $n = 6$ when $T = \mathbb{Z}[\omega]$. □

Proposition 4.3. *The good region $\mathcal{G}_s$ consists of* a union of disks. *These disks are pairwise disjoint if the message rate $R \geq 2$.*

Proof: Recall that α is good if and only if $R < R(\alpha, \mathbf{a})$ for some $\mathbf{a} \in T^L \setminus \{\mathbf{0}\}$, or equivalently,

$$\mathsf{SNR}\,\|\alpha\mathbf{h} - \mathbf{a}\|^2 + |\alpha|^2 < \mathsf{SNR}\,/2^R \text{ for some } \mathbf{a} \in T^L \setminus \{\mathbf{0}\}.$$

Now observe that the term $\mathsf{SNR}\,\|\alpha\mathbf{h}-\mathbf{a}\|^2+|\alpha|^2$ is equal to the squared distance between two vectors $(\alpha h_1\sqrt{\mathsf{SNR}}, \ldots, \alpha h_L\sqrt{\mathsf{SNR}}, \alpha)$ and $(a_1\sqrt{\mathsf{SNR}}, \ldots, a_L\sqrt{\mathsf{SNR}}, 0)$. Hence, we have

$$\begin{aligned} & \mathsf{SNR}\,\|\alpha\mathbf{h} - \mathbf{a}\|^2 + |\alpha|^2 \\ = \;& \mathsf{SNR}\,\|\alpha^*\mathbf{h} - \mathbf{a}\|^2 + |\alpha^*|^2 + |\alpha - \alpha^*|^2(1 + \mathsf{SNR}\,\|\mathbf{h}\|^2), \end{aligned}$$

where α^* is the MMSE coefficient given by

$$\alpha^* = \frac{\mathsf{SNR}\,\mathbf{a}\mathbf{h}^{\mathsf{H}}}{1 + \mathsf{SNR}\,\|\mathbf{h}\|^2}. \tag{4.2}$$

Recall that $\mathsf{SNR}\,\|\alpha^*\mathbf{h} - \mathbf{a}\|^2 + |\alpha^*|^2 = \mathsf{SNR}\,/2^{R(\alpha^*, \mathbf{a})}$. Therefore, α is good if and only if

$$|\alpha - \alpha^*|^2 < \frac{\mathsf{SNR}}{1 + \mathsf{SNR}\,\|\mathbf{h}\|^2}\left(\frac{1}{2^R} - \frac{1}{2^{R(\alpha^*, \mathbf{a})}}\right) \tag{4.3}$$

for some $\mathbf{a} \in T^L \setminus \{\mathbf{0}\}$. That is, a good α is in some disk of center α^*. This proves the first part of Proposition 4.3.

We proceed to the second part. If two disks overlap, then there exists a scalar α such that

$$\mathsf{SNR}\,\|\alpha\mathbf{h} - \mathbf{a}\|^2 + |\alpha|^2 < \mathsf{SNR}\,/2^R$$

and

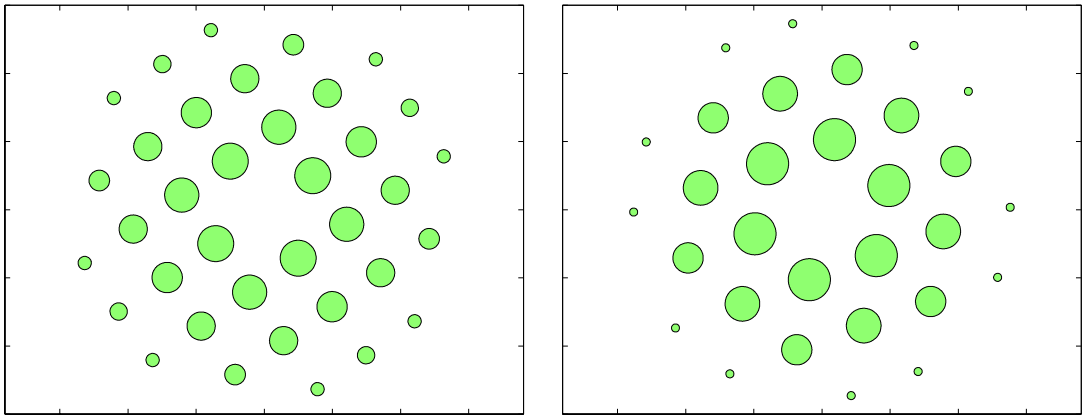
$$\mathsf{SNR}\,\|\alpha\mathbf{h} - \mathbf{b}\|^2 + |\alpha|^2 < \mathsf{SNR}\,/2^R$$

for some $\mathbf{a}, \mathbf{b} \in T^L \setminus \{\mathbf{0}\}$ with $\mathbf{a} \neq \mathbf{b}$. It follows that

$$\mathsf{SNR}(\|\alpha\mathbf{h} - \mathbf{a}\|^2 + \|\alpha\mathbf{h} - \mathbf{b}\|^2) + 2|\alpha|^2 < \mathsf{SNR}\,/2^{R-1}.$$

On the other hand, we have

$$\begin{aligned} \mathsf{SNR}(\|\alpha\mathbf{h} - \mathbf{a}\|^2 + \|\alpha\mathbf{h} - \mathbf{b}\|^2) &\geq \mathsf{SNR}\,\|\mathbf{a} - \mathbf{b}\|^2/2 \\ &\geq \mathsf{SNR}\,/2, \end{aligned}$$



(a) (b)

Figure 4.1: Good regions for asymptotically-good nested lattice codes: (a) $T = \mathbb{Z}[i]$, $h_1 = -0.93 + 0.65i$, $h_2 = -0.04i$, $\mathsf{SNR} = 20$ dB, and $R = \log_2 10$; (b) $T = \mathbb{Z}[\omega]$, $h_1 = 0.72 + 0.61i$, $h_2 = -0.05i$, $\mathsf{SNR} = 20$ dB, and $R = \log_2 10$.

where the first inequality follows from the fact that $\|\alpha\mathbf{h} - \mathbf{a}\| + \|\alpha\mathbf{h} - \mathbf{b}\| \geq \|\mathbf{a} - \mathbf{b}\|$, and the second inequality follows from the fact that the norm of any nonzero element in T must be no less than 1. Hence, we have $\mathsf{SNR}/2 < \mathsf{SNR}/2^{R-1}$, or equivalently, $R < 2$. In other words, if the message rate $R \geq 2$, there are no overlapping disks. □

Fig. 4.1(a) and 4.1(b) show some typical good regions for asymptotically-good nested lattice codes with $T = \mathbb{Z}[i]$ and $T = \mathbb{Z}[\omega]$, respectively. The rotation angles in Fig. 4.1(a) and 4.1(b) are 90° and 60°, respectively, as explained in Proposition 4.2.

Fig. 4.2 depicts a typical good region for a simple nested lattice code $\mathcal{L}(\mathbb{Z}[i]^{400}, 2\mathbb{Z}[i]^{400}, \mathbf{d})$ with $\mathbf{d} = \frac{1}{2}(1 + i, \ldots, 1 + i)$, which is also known as uncoded 4-QAM with four constellation points $\left\{\frac{1}{2}(\pm 1 \pm i)\right\}$. Since this nested lattice code is not asymptotically good, the disjoint areas are not quite disk-like. That is, Proposition 4.3 approximately holds here. Nevertheless, the good region is still bounded and symmetric (with respect to rotations by 90°). That is, Propositions 4.1 and 4.2 still hold here.

Based on the above properties, we can give a quantitative discussion on the challenge of designing C&F schemes based on channel estimation. Suppose that $\hat{\mathbf{h}} = \mathbf{h} + \mathbf{e}$, where $\mathbf{e} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I}_L)$ models the channel estimation error. This model is widely used in training-based channel estimation. For simplicity, assume that the channel gains follow Rayleigh fading, i.e., $\mathbf{h} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_L)$. Then the key parameter capturing the channel estimation error is

$$\mathsf{SNR}_{\mathsf{est}} = 1/\sigma^2.$$

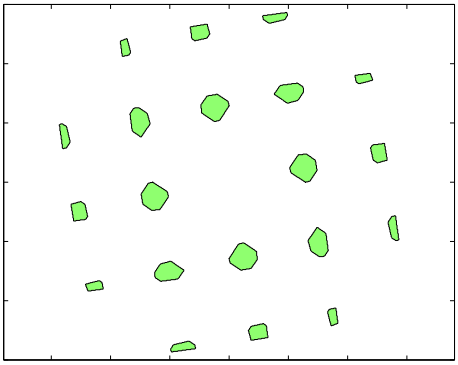


Figure 4.2: A good region for the nested lattice code $\mathcal{L}(\mathbb{Z}[i]^{400}, 2\mathbb{Z}[i]^{400}, \mathbf{d})$, where $h_1 = 0.11 + 0.73i$, $h_2 = 0.78 + 0.19i$, and $\mathsf{SNR} = 35$ dB.

We will show that even when $\mathsf{SNR}_{\mathsf{est}} \gg 1$ (i.e., the estimation error e_ℓ is much smaller than the variance of h_ℓ), the impact of the channel estimation error on the performance of coherent C&F is still significant.

Recall that without estimation error, the best computation rate is given by

$$R_{\mathsf{comp}}(\mathbf{h}) = \log_2 \left(\frac{\mathsf{SNR}}{\mathsf{SNR} \, \|\alpha^* \mathbf{h} - \mathbf{a}^*\|^2 + |\alpha^*|^2} \right), \tag{4.4}$$

where $(\alpha^*, \mathbf{a}^*)$ is an optimal solution to the optimization problem

$$\begin{aligned} \min \ & \mathsf{SNR} \, \|\alpha \mathbf{h} - \mathbf{a}\|^2 + |\alpha|^2 \\ \text{s.t.} \ & \mathbf{0} \neq \mathbf{a} \in \mathbb{Z}[i]^L \\ & \alpha \in \mathbb{C} \end{aligned}$$

Similarly, with channel estimation error, the computation rate is given by

$$\hat{R}_{\mathsf{comp}}(\mathbf{h}) = \log_2 \left(\frac{\mathsf{SNR}}{\mathsf{SNR} \, \|\hat{\alpha}^* \mathbf{h} - \hat{\mathbf{a}}^*\|^2 + |\hat{\alpha}^*|^2} \right), \tag{4.5}$$

where $(\hat{\alpha}^*, \hat{\mathbf{a}}^*)$ is an optimal solution to the optimization problem

$$\begin{aligned} \min \ & \mathsf{SNR} \, \|\alpha \hat{\mathbf{h}} - \mathbf{a}\|^2 + |\alpha|^2 \\ \text{s.t.} \ & \mathbf{0} \neq \mathbf{a} \in \mathbb{Z}[i]^L \\ & \alpha \in \mathbb{C} \end{aligned}$$

Clearly, $\hat{R}_{\mathsf{comp}}(\mathbf{h})$ is always less than $R_{\mathsf{comp}}(\mathbf{h})$ in the presence of estimation error. To better compare

(4.4) and (4.5), let us consider a special case where $\hat{\mathbf{a}}^* = \mathbf{a}^*$. In this case, (4.5) can be rewritten as

$$\hat{R}_{\mathsf{comp}}(\mathbf{h}) = \log_2 \left(\frac{\mathsf{SNR}}{\mathsf{SNR}\,\|\alpha^*\mathbf{h} - \mathbf{a}^*\|^2 + |\alpha^*|^2 + |\hat{\alpha}^* - \alpha^*|^2(1 + \mathsf{SNR}\,\|\mathbf{h}\|^2)} \right). \tag{4.6}$$

Hence, the term $|\hat{\alpha}^* - \alpha^*|^2$ should be in the order of $1/(1 + \mathsf{SNR}\,\|\mathbf{h}\|^2)$ to ensure that (4.6) is close to (4.4) as SNR increases. We note that this requirement is demanding, since $\hat{\alpha}^*$ is sensitive to channel estimation error $\mathbf{e}$, as shown in our extensive simulations. In fact, even if we set $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}$, we still observe a significant rate loss defined as

$$E_{\mathbf{h}} \left[\frac{R_{\mathsf{comp}}(\mathbf{h}) - \hat{R}_{\mathsf{comp}}(\mathbf{h})}{R_{\mathsf{comp}}(\mathbf{h})} \right].$$

Fig. 4.3 depicts the average rate loss when $\mathsf{SNR}_{\mathsf{est}}$ is set to 10 dB, 15 dB and 20 dB. As we can see from Fig. 4.3, even if we set $\mathsf{SNR}_{\mathsf{est}} = 20$ dB, the average rate loss is as high as 20% when SNR approaches 20 dB, and is around 10% when SNR is 14 dB (which is 6 dB less than $\mathsf{SNR}_{\mathsf{est}}$). Hence, it suggests that C&F with channel estimation does not scale well with the number of users.

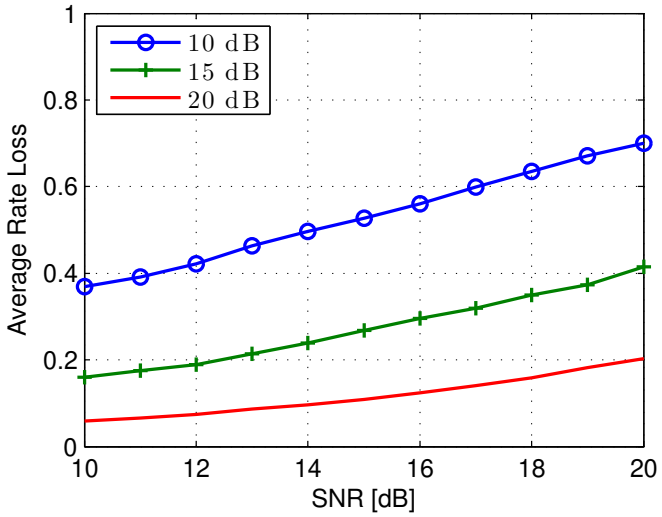


Figure 4.3: Average rate loss when $\mathsf{SNR}_{\mathsf{est}}$ is set to 10 dB, 15 dB and 20 dB.

4.2.2 The use of $a_{\mathsf{sum}}\mathbf{d}$ mod Λ' in the decoding

Here, we explain why knowledge of $a_{\mathsf{sum}}\mathbf{d}$ mod Λ' is sufficient for successful decoding. Recall that a good scalar α ensures successful decoding, i.e., $\varphi\left(\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d})\right) = \sum_{\ell} a_{\ell}\mathbf{w}_{\ell}$ for some coefficients $(a_1, \ldots, a_L) \in T^L \setminus \{\mathbf{0}\}$. Note that if the term $a_{\mathsf{sum}}\mathbf{d}$ is replaced by $a_{\mathsf{sum}}\mathbf{d}$ mod Λ', the decoding is still

successful. To see this, recall that $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda' = a_{\mathsf{sum}}\mathbf{d} - \boldsymbol{\lambda}'$ for some $\boldsymbol{\lambda}' \in \Lambda'$. Hence, we have

$$\begin{aligned}\varphi\left(\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\alpha\mathbf{y} - (a_{\mathsf{sum}}\mathbf{d} - \boldsymbol{\lambda}'))\right) &= \varphi\left(\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d}) + \boldsymbol{\lambda}'\right) \\ &= \varphi\left(\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d})\right) + \varphi(\boldsymbol{\lambda}') \\ &= \sum_{\ell} a_{\ell}\mathbf{w}_{\ell}.\end{aligned}$$

Therefore, the receiver only needs to know $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda'$ after obtaining a good scalar α.

We observe that $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda'$ often has a limited number of choices for commonly-used nested lattice codes, especially those obtained from linear codes [74]. For instance, we can construct a nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ via an $[n, k]$ binary code $\mathcal{C}_{n,k}$ through lifted Construction A [74]. In particular, we have $\Lambda' = 2\mathbb{Z}[i]^n$, $\mathbf{d} = \frac{1}{2}(1 + i, \ldots, 1 + i)$, and $\boldsymbol{\lambda} \in \Lambda$ if and only if $\mathrm{Re}\{\boldsymbol{\lambda}\} \bmod 2$, $\mathrm{Im}\{\boldsymbol{\lambda}\} \bmod 2$ are codewords in $\mathcal{C}_{n,k}$. Such a nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ (known as coded 4-QAM) admits only eight possible choices of $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda'$, namely, $\{0\mathbf{d}, \pm\mathbf{d}, \pm i\mathbf{d}, , (1 \pm i)\mathbf{d}, 2\mathbf{d}\}$. This greatly reduces the search space of $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda'$ for our blind C&F schemes.

4.2.3 Generic blind C&F scheme

In previous sections, we have observed a number of properties of good scalars. We now present a generic blind C&F scheme based on these observations.

The first idea behind our generic scheme is to reduce the search space of good scalars as much as possible. Proposition 4.1 suggests that, to find a good scalar, it suffices to consider a bounded region. Proposition 4.2 shows that, to find a good scalar, it suffices to "ignore" some unnecessary areas. For instance, only the region in the first quadrant is worth investigating for nested lattice codes with $T = \mathbb{Z}[i]$. Proposition 4.3 implies that, to find a good scalar, it suffices to "probe" a discrete set of points. The denser this set, the better the performance.

The second idea is to use error detection to "probe" a given point, deciding whether this point is good or bad. Specifically, the transmitters embed a linear error-detecting code $\mathcal{C}$ into the message space W so that each *valid* message $\mathbf{w}_{\ell}$ (as well as any linear combinations) is a codeword in $\mathcal{C}$. The receiver performs a basic probing operation as described in Algorithm 2.

If the point α is good, Algorithm 2 always declares α to be good and outputs some non-zero codeword $\hat{\mathbf{u}}$. If the point α is bad, Algorithm 2 might declare that α is good due to an undetected error. Such an error is called a Type-I error. The probability of a Type-I error can be made very small in practice by endowing $\mathcal{C}$ with sufficiently many parity checks.

Algorithm 2 *Basic probing operation*

Input: a point α.
Output: α is bad, or a good α with its associated $\hat{\mathbf{u}}$.

1. **for** each $\mathbf{t} \in \{a\mathbf{d} \bmod \Lambda' : a \in T\}$ **do**
2. Compute $\hat{\mathbf{u}} = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - \mathbf{t})\right)$.
3. **if** $\hat{\mathbf{u}}$ is a non-zero codeword in $\mathcal{C}$ **then**
4. Declare α is good, output $\hat{\mathbf{u}}$, and then stop.
5. **end if**
6. **end for**
7. Declare α is bad.

When Algorithm 2 finds a good scalar α, it does not necessarily mean that its output $\hat{\mathbf{u}} = \sum_\ell a_\ell \mathbf{w}_\ell$ for some a_ℓ. This is because the associated $\mathbf{t}$ can be different from $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda'$ due to an undetected error. We call such an error a Type-II error. As we will see in Sec. 4.3, Type-II errors can be eliminated for certain nested lattice codes.

Now we are ready to describe a generic blind C&F scheme. The input to the scheme is an ordered list containing a discrete set of points. The scheme probes the points in the list one by one until it finds a good scalar or until it reaches the end of the list. The output is either a non-zero codeword $\hat{\mathbf{u}}$ (when the scheme finds a good scalar) or nothing (when the scheme finds no good scalars).

We note that the performance of the above generic scheme depends on the points in the list (but not on their probing order), whereas the computational cost of the scheme depends on the probing order of these points. In other words, two ordered lists containing exactly the same points achieve the same performance with possibly quite different computational complexity. We also note that the computational cost of the basic probing operation can be greatly reduced for some commonly-used nested lattice codes. All of these will be discussed in the next section.

4.3 Blind Compute-and-Forward: Efficient Algorithms

In this section, we propose three (complementary) strategies to reduce the computational complexity of the generic blind C&F scheme presented in Sec. 4.2. The first strategy attempts to create some "smart" probing lists. The second strategy aims to detect bad scalars at a low cost. The third strategy further reduces the complexity of the basic probing operation.

4.3.1 Hierarchically-organized list-building

The choice of the probing list is crucial to attaining good performance with low complexity. For instance, when the good region consists of many large disjoint areas, the probing points can be made relatively

sparse. On the other hand, when the good region consists of a few small disjoint areas, the probing points should be made relatively dense. Based on this observation, we propose a heuristic method for creating the list.

First, we choose a well-shaped region $\mathcal{R}$ to avoid unnecessary probing (see discussions in Sec. 4.2.3). For nested lattice codes with $T = \mathbb{Z}[i]$, we note that $\mathcal{R}$ can be chosen heuristically as $[0, \log_{10}(\mathsf{SNR})/R] \times [0, \log_{10}(\mathsf{SNR})/R]$ (where R is the message rate). For example, if $\mathsf{SNR} = 10$ dB and $R = 1$, then $\mathcal{R} = [0, 1] \times [0, 1]$.

Then, we construct an m-level lattice-partition chain [75] $\mathcal{L}_0/\mathcal{L}_1/\ldots/\mathcal{L}_m$ in $\mathbb{C}$ (i.e., each $\mathcal{L}_j$ is a one-dimensional complex lattice and $\mathcal{L}_0 \supset \mathcal{L}_1 \supset \cdots \supset \mathcal{L}_m$). Note that the lattice-partition chain, together with the region $\mathcal{R}$, induces $m+1$ probing grids $\{\mathcal{L}_j \cap \mathcal{R}\}$ satisfying $\{\mathcal{L}_m \cap \mathcal{R}\} \subset \cdots \subset \{\mathcal{L}_0 \cap \mathcal{R}\}$ (see Fig. 4.4 for a concrete example). For nested lattice codes with $T = \mathbb{Z}[i]$, we heuristically set $\mathcal{L}_j = \frac{1}{16}\log_{10}(\mathsf{SNR})(1+i)^j\mathbb{Z}[i]$, where $j = 0, \ldots, 8$.

Figure 4.4: An illustration of three (self-similar) probing grids. We choose $\mathcal{L}_j = (1+i)^j\mathbb{Z}[i]$ $(j = 0, 1, 2)$ and $\mathcal{R} = [0, 3] \times [0, 3]$. The sparsest grid consists of 4 solid points. The second sparsest grid consists of 4 solid points and 4 partially solid points.

With these grids, a list-building algorithm is described in Algorithm 3. The basic idea is to design a list such that the points in the sparser grids will appear before the points in the denser grids. That is, the points in $\mathcal{L}_j \cap \mathcal{R}$ are ordered to appear before the points in $\mathcal{L}_{j+1} \cap \mathcal{R}$.

Algorithm 3 *Hierarchically-organized list-building algorithm*

Input: a lattice-partition chain $\mathcal{L}_0/\ldots/\mathcal{L}_m$ with a region $\mathcal{R}$.
Output: an ordered list of probing points.

1. Set $\mathsf{list} = \emptyset$.
2. Set $j = m$ and set $\mathcal{L}_{m+1} = \{\mathbf{0}\}$.
3. **while** $j \geq 0$ **do**
4. Let $\mathcal{S} = (\mathcal{L}_j \setminus \mathcal{L}_{j+1}) \cap \mathcal{R}$.
5. **while** $|\mathcal{S}| > 0$ **do**
6. Find a point α in $\mathcal{S}$ of the smallest L_1-norm.
7. Set $\mathsf{list} = \mathsf{list} \cup \{\alpha\}$. Set $\mathcal{S} = \mathcal{S} \setminus \{\alpha\}$.
8. **end while**
9. Set $j = j - 1$.
10. **end while**

4.3.2 Quick detection for bad scalars

Note that the basic probing operation needs to compute $\hat{\mathbf{u}} = \varphi\left(\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\alpha\mathbf{y} - a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda')\right)$ for each possible $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda'$. Such computations are often costly, due to the use of the nearest-neighbor-quantizer $\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\cdot)$. Here, we propose a new probing method that identifies certain bad scalars without the use of $\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\cdot)$. Our new method only involves very simple computations, which is inspired by the *Smoothing Lemma* from lattice theory [76, 77].

Definition 4.3 (Smoothing parameter)**.** *For a complex lattice Λ and for any $\epsilon > 0$, the* smoothing parameter $\eta_{\Lambda}(\epsilon)$ *is the smallest $\sigma > 0$ such that $\sum_{\boldsymbol{\lambda}^* \in \Lambda^* \setminus \{\mathbf{0}\}} e^{-\pi^2\sigma^2\|\boldsymbol{\lambda}^*\|^2} \leq \epsilon$, where Λ^* is the dual lattice of Λ.*

Clearly, $\eta_{\Lambda}(\epsilon)$ is a monotonically decreasing function of ϵ. That is, for $\epsilon_1 < \epsilon_2$, we have $\eta_{\Lambda}(\epsilon_1) > \eta_{\Lambda}(\epsilon_2)$. The smoothing parameter bounds the variational distance between the Gaussian distribution mod Λ and the uniform distribution u_{Λ} on the Voronoi region $\mathcal{V}_{\Lambda}(\mathbf{0})$.

Lemma 4.1 (Smoothing Lemma)**.** *Let $\mathbf{n}$ be an i.i.d. circularly-symmetric complex Gaussian random vector with mean μ and variance σ^2, i.e., $\mathbf{n} \sim \mathcal{CN}(\mu\mathbf{1}, \sigma^2\mathbf{I}_{n\times n})$. Let $f_{\Lambda}(\cdot)$ be the probability density function of $\mathbf{n}$ mod Λ. Then for any $\sigma > \eta_{\Lambda}(\epsilon)$, the variational distance between f_{Λ} and u_{Λ} is bounded by ϵ, i.e.,*

$$\int_{\mathcal{V}_{\Lambda}(\mathbf{0})} |f_{\Lambda}(\mathbf{t}) - u_{\Lambda}(\mathbf{t})| d\mathbf{t} \leq \epsilon.$$

The Smoothing Lemma says that $\mathbf{n}$ mod Λ tends to be uniform over the Voronoi region $\mathcal{V}_{\Lambda}(\mathbf{0})$ as σ grows. This facilitates the detection of bad scalars. For any nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$, we have

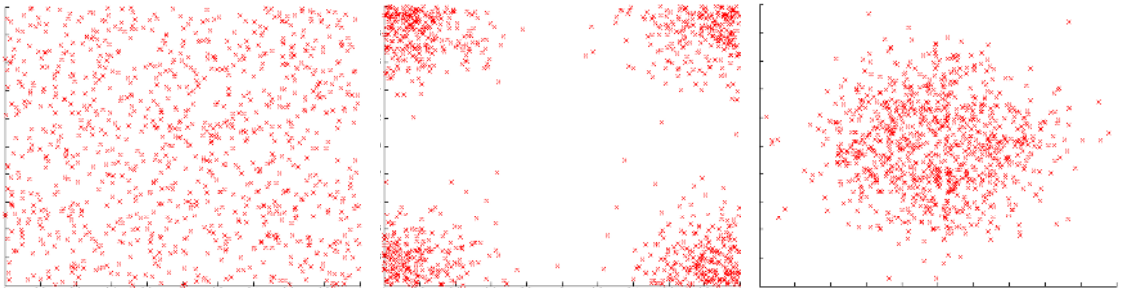
$$\begin{aligned}\alpha\mathbf{y} \bmod \Lambda &= \left(\sum_{\ell} a_{\ell}\tilde{\varphi}(\mathbf{w}_{\ell}) + \mathbf{n}_{\mathsf{eff}} + a_{\mathsf{sum}}\mathbf{d}\right) \bmod \Lambda \\ &= (\mathbf{n}_{\mathsf{eff}} + a_{\mathsf{sum}}\mathbf{d}) \bmod \Lambda.\end{aligned}$$

More generally, let Λ_0 be a lattice that contains Λ (i.e., $\Lambda \subset \Lambda_0$), then we have

$$\alpha\mathbf{y} \bmod \Lambda_0 = (\mathbf{n}_{\mathsf{eff}} + a_{\mathsf{sum}}\mathbf{d}) \bmod \Lambda_0.$$

When the nested lattice code is asymptotically good (in the sense of [26]), we have $\mathbf{n}_{\mathsf{eff}} \sim \mathcal{CN}(\mathbf{0}, \sigma^2\mathbf{I}_{n\times n})$, where $\sigma^2 = P\|\alpha\mathbf{h} - \mathbf{a}\|^2 + N_0|\alpha|^2$. Hence, by Lemma 4.1, $\alpha\mathbf{y}$ mod Λ_0 tends to be uniform over $\mathcal{V}_{\Lambda_0}(\mathbf{0})$ as σ becomes larger (or equivalently, as the scalar α becomes worse). Interestingly, this property still (approximately) holds for many commonly-used nested lattice codes, as suggested by our extensive

numerical studies.



(a) (b) (c)

Figure 4.5: Scatter-plots for $\alpha\mathbf{y} \bmod \Lambda_0$ with $h_1 = -1.17 + 1.40i$, $h_2 = -0.01 - 0.71i$, and $\mathsf{SNR} = 16$ dB: (a) a bad scalar $\alpha = 1.98 + 1.01i$; (b) a good scalar $\alpha = -0.12 + 1.52i$ with $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda_0 = \mathbf{d}$; (c) a good scalar $\alpha = 1.21 - 0.24i$ with $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda_0 = \mathbf{0}$.

To illustrate this, we use a nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ obtained via a $[1000, 500]$ binary LDPC code through lifted Construction A [74]. In particular, we have $\Lambda' = 2\mathbb{Z}[i]^{1000} \subset \Lambda \subset \mathbb{Z}[i]^{1000}$, $\mathbf{d} = \frac{1}{2}(1 + i, \ldots, 1 + i)$, and we choose $\Lambda_0 = \mathbb{Z}[i]^{1000}$. Fig. 4.5(a), 4.5(b), and 4.5(c) provide scatter-plots for $\alpha\mathbf{y} \bmod \Lambda_0$ with a bad scalar and two good scalars, respectively. Clearly, $\alpha\mathbf{y} \bmod \Lambda_0$ is close to uniform for a bad scalar, and is highly non-uniform for good scalars. In particular, $\alpha\mathbf{y} \bmod \Lambda_0$ distributes around four corners for a good scalar with $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda_0 = \mathbf{d}$ (see Fig. 4.5(b)), and is centered at the origin for a good scalar with $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda_0 = \mathbf{0}$ (see Fig. 4.5(c)). It is easy to check that $a_{\mathsf{sum}}\mathbf{d} \bmod \Lambda_0$ takes values in $\{\mathbf{0}, \mathbf{d}\}$. So, we have two cases here in total.

Let $\mathbf{u}_{\Lambda_0}$ be a random vector uniform over the Voronoi region $\mathcal{V}_{\Lambda_0}(\mathbf{0})$. Clearly, $\mathbf{u}_{\Lambda_0}$ consists of i.i.d. random variables with variance $1/6$. On the other hand, the sample variances of $\alpha\mathbf{y} \bmod \Lambda_0$ in Fig. 4.5(a), 4.5(b), and 4.5(c) are 0.165, 0.322, and 0.051, respectively. As expected, the sample variance of $\alpha\mathbf{y} \bmod \Lambda_0$ is close to $1/6 \approx 0.167$ for some bad scalars, and is away from $1/6$ for good scalars. This example confirms our previous observations. More importantly, it suggests a quick detection algorithm to identify some bad scalars. For ease of presentation, Algorithm 4 assumes that $T = \mathbb{Z}[i]$ and $\Lambda_0 = \mathbb{Z}[i]^n$ (which can be extended as we will soon see).

Note that when $\Lambda_0 = \mathbb{Z}[i]^n$, the cost of computing $\alpha\mathbf{y} \bmod \Lambda_0$ is very low. In this case, $\alpha\mathbf{y} \bmod \Lambda_0 = \alpha\mathbf{y} - \mathrm{round}(\alpha\mathbf{y})$, where $\mathrm{round}(\cdot)$ is the standard rounding operation. Note also that the complexity of Algorithm 4 can be further reduced by operating on a subset of $\alpha\mathbf{y}$. For instance, in our previous numerical example, if we only operate on the first 100 elements of $\alpha\mathbf{y}$, then the new sample variances are 0.166, 0.328, and 0.054, respectively, which are quite close to the original sample variances. Hence, it suffices to consider only a subset of $\alpha\mathbf{y}$ in practice. Therefore, the complexity of Algorithm 4 can be made very low. Also, note that Algorithm 4 can be easily extended to other cases, such as $T = \mathbb{Z}[\omega]$ and

Algorithm 4 *Quick detection for bad scalars when $T = \mathbb{Z}[i]$ and $\Lambda_0 = \mathbb{Z}[i]^n$*

Input: a point α, a threshold δ.
Output: α is bad, or a good α with its associated $\hat{\mathbf{u}}$.

1. Compute $\mathbf{v} = \alpha\mathbf{y} \bmod \Lambda_0$.
2. Compute the sample mean $\bar{v} = \frac{1}{n}\sum_i v_i$.
3. Compute the sample variance $s^2 = \frac{1}{n-1}\sum_i |v_i - \bar{v}|^2$.
4. **if** $s^2 \in [1/6 - \delta,\ 1/6 + \delta]$ **then**
5. Declare α is bad.
6. **else**
7. Perform the basic probing operation.
8. **end if**

$\Lambda_0 = \mathbb{Z}[\omega]^n$. Clearly, its complexity remains to be low as long as Λ_0 is simple enough.

4.3.3 Fast probing operation

Next, we present a fast probing method that reduces the use of $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\cdot)$ in the basic probing operation. Our method requires the underlying nested lattice code to satisfy some mild conditions. For ease of presentation, we focus on a case study in which the conditions are $T = \mathbb{Z}[i]$ and $(1+i)\mathbf{d} \in \Lambda$.

Algorithm 5 *Fast probing when $T = \mathbb{Z}[i]$ and $(1+i)\mathbf{d} \in \Lambda$*

Input: a point α.
Output: α is bad, or a good α with its associated $\hat{\mathbf{u}}$.

1. Compute $\hat{\mathbf{u}}_1 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y})\right)$.
2. **for** each $\mathbf{t} \in \{b(1+i)\mathbf{d} \bmod \Lambda' : b \in \mathbb{Z}[i]\}$ **do**
3. **if** $\hat{\mathbf{u}}_1 - \varphi(\mathbf{t})$ is a non-zero codeword in $\mathcal{C}$ **then**
4. Declare α is good, output $\hat{\mathbf{u}} = \hat{\mathbf{u}}_1 - \varphi(\mathbf{t})$, and then stop.
5. **end if**
6. **end for**
7. Compute $\hat{\mathbf{u}}_2 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - \mathbf{d})\right)$.
8. **for** each $\mathbf{t} \in \{b(1+i)\mathbf{d} \bmod \Lambda' : b \in \mathbb{Z}[i]\}$ **do**
9. **if** $\hat{\mathbf{u}}_2 - \varphi(\mathbf{t})$ is a non-zero codeword in $\mathcal{C}$ **then**
10. Declare α is good, output $\hat{\mathbf{u}} = \hat{\mathbf{u}}_2 - \varphi(\mathbf{t})$, and then stop.
11. **end if**
12. **end for**
13. Declare α is bad.

Our fast probing operation presented in Algorithm 5 requires at most two uses of $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\cdot)$, while still achieving the same performance as the basic probing operation, as shown in the following theorem.

Theorem 4.1. *When $T = \mathbb{Z}[i]$ and $(1+i)\mathbf{d} \in \Lambda$, Algorithm 5 is equivalent to Algorithm 2.*

Proof: We only need to show that for each computation $\hat{\mathbf{u}} = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - \mathbf{t})\right)$ in Algorithm 2, there is a corresponding computation in Algorithm 5. Suppose that $\mathbf{t} = a\mathbf{d} \bmod \Lambda'$ for some $a \in \mathbb{Z}[i]$. Then $\mathbf{t} = a\mathbf{d} - \boldsymbol{\lambda}'$ for some $\boldsymbol{\lambda}' \in \Lambda'$. Note that every $a \in \mathbb{Z}[i]$ can be expressed as $a = b(1+i) + c$, where

$b \in \mathbb{Z}[i]$ and $c \in \{0,1\}$. (This is a natural generalization of the binary expansion.) Hence, we have $\mathbf{t} = b(1+i)\mathbf{d} + c\mathbf{d} - \boldsymbol{\lambda}'$ for some $b \in \mathbb{Z}[i]$ and $c \in \{0,1\}$. Therefore,

$$\begin{aligned}\hat{\mathbf{u}} &= \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - c\mathbf{d} - b(1+i)\mathbf{d} + \boldsymbol{\lambda}')\right)\\ &= \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - c\mathbf{d}) - b(1+i)\mathbf{d} + \boldsymbol{\lambda}'\right)\\ &= \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - c\mathbf{d})\right) - \varphi(b(1+i)\mathbf{d})\\ &= \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - c\mathbf{d})\right) - \varphi\left(b(1+i)\mathbf{d} \bmod \Lambda'\right),\end{aligned}$$

which is indeed a computation in Algorithm 5. □

We note that $(1+i)\mathbf{d} \in \Lambda$ is a mild constraint for certain nested lattice codes. For example, for a nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ obtained via an $[n,k]$ binary code $\mathcal{C}_{n,k}$ through lifted Construction A, $(1+i)\mathbf{d} \in \Lambda$ simply means that the binary code $\mathcal{C}_{n,k}$ contains the all-ones codeword, which implies that each parity-check equation for $\mathcal{C}_{n,k}$ involves an even number of bits. When $\mathcal{C}_{n,k}$ is an LDPC code, this means that all the check degrees are even, which can be easily satisfied in practice.

4.3.4 Combining our strategies together

Now, we are ready to combine the quick detection strategy and fast probing strategy together. We note that such a combination has several unique advantages for nested lattice codes obtained via $[n,k]$ binary codes through lifted Construction A. For this family of nested lattice codes, $\alpha\mathbf{y} \bmod \mathbb{Z}[i]^n$ reveals whether $a_{\mathsf{sum}}\mathbf{d} \bmod \mathbb{Z}[i]^n = \mathbf{0}$ or $\mathbf{d}$ for good scalars, as discussed in Sec. 4.3.2. Note that $a_{\mathsf{sum}}\mathbf{d} \bmod \mathbb{Z}[i]^n = \mathbf{0}$ means that $a_{\mathsf{sum}} = b(1+i)$ for some $b \in \mathbb{Z}[i]$. In this case, it suffices to compute $\hat{\mathbf{u}}_1 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y})\right)$. Similarly, $a_{\mathsf{sum}}\mathbf{d} \bmod \mathbb{Z}[i]^n = \mathbf{d}$ means that $a_{\mathsf{sum}} = b(1+i)+1$ for some $b \in \mathbb{Z}[i]$, and it suffices to compute $\hat{\mathbf{u}}_2 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - \mathbf{d})\right)$. Hence, only one use of $\mathcal{Q}_\Lambda^{\mathrm{NN}}(\cdot)$ is needed with the help of $\alpha\mathbf{y} \bmod \mathbb{Z}[i]^n$. This leads to a faster probing method presented in Algorithm 6. Moreover, Algorithm 6 has another advantage: Type-II errors can be completely eliminated if the syndrome of $\varphi\left((1+i)\mathbf{d}\right)$ contains a unit.

Theorem 4.2. *Suppose that $a_{\mathsf{sum}}\mathbf{d}$ mod $\mathbb{Z}[i]^n$ is revealed correctly. Then Type-II errors cannot occur as long as the syndrome of $\varphi\left((1+i)\mathbf{d}\right)$ contains a unit.*

Proof: We assume, without loss of generality, that $a_{\mathsf{sum}} = b(1+i)+1$ for some $b \in \mathbb{Z}[i]$. In this case, we have $\hat{\mathbf{u}}_0 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - \mathbf{d})\right)$, since $a_{\mathsf{sum}}\mathbf{d} \bmod \mathbb{Z}[i]^n$ is revealed correctly. To prove that there is no Type-II error, we only need to show that there is a unique $\mathbf{t} \in \{b(1+i)\mathbf{d} \bmod \Lambda' : b \in \mathbb{Z}[i]\}$ such that $\hat{\mathbf{u}}_0 - \varphi(\mathbf{t})$ is a non-zero codeword in $\mathcal{C}$. Suppose that there exist $\mathbf{t}_1$ and $\mathbf{t}_2$ such that both $\hat{\mathbf{u}}_0 - \varphi(\mathbf{t}_1)$ and $\hat{\mathbf{u}}_0 - \varphi(\mathbf{t}_2)$ are non-zero codewords in $\mathcal{C}$. Then, $\varphi(\mathbf{t}_1 - \mathbf{t}_2) = \varphi(\mathbf{t}_1) - \varphi(\mathbf{t}_2)$ must be a codeword in $\mathcal{C}$.

Algorithm 6 *Faster probing when* $T = \mathbb{Z}[i]$, $(1+i)\mathbf{d} \in \Lambda$, *and* $\Lambda_0 = \mathbb{Z}[i]^n$

Input: a point α, a threshold δ.
Output: α is bad, or a good α with its associated $\hat{\mathbf{u}}$.

1. Compute $\mathbf{v} = \alpha\mathbf{y} \bmod \mathbb{Z}[i]^n$.
2. Compute the sample mean $\bar{v} = \frac{1}{n}\sum_i v_i$.
3. Compute the sample variance $s^2 = \frac{1}{n-1}\sum_i |v_i - \bar{v}|^2$.
4. **if** $s^2 \in [1/6 - \delta,\ 1/6 + \delta]$ **then**
5. Declare α is bad and stop.
6. **else if** $s^2 < 1/6 - \delta$ **then**
7. Compute $\hat{\mathbf{u}}_0 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y})\right)$.
8. **else if** $s^2 > 1/6 + \delta$ **then**
9. Compute $\hat{\mathbf{u}}_0 = \varphi\left(\mathcal{Q}_\Lambda^{\mathrm{NN}}(\alpha\mathbf{y} - \mathbf{d})\right)$.
10. **end if**
11. **for** each $\mathbf{t} \in \{b(1+i)\mathbf{d} \bmod \Lambda' : b \in \mathbb{Z}[i]\}$ **do**
12. **if** $\hat{\mathbf{u}}_0 - \varphi(\mathbf{t})$ is a non-zero codeword in $\mathcal{C}$ **then**
13. Declare α is good, output $\hat{\mathbf{u}} = \hat{\mathbf{u}}_0 - \varphi(\mathbf{t})$, and then stop.
14. **end if**
15. **end for**
16. Declare α is bad.

In particular, it means that the syndrome of $\varphi(\mathbf{t}_1 - \mathbf{t}_2)$ is the zero vector. Recall that the syndrome of $\varphi\left((1+i)\mathbf{d}\right)$ contains a unit. Hence, the syndrome of $\varphi(\mathbf{t}_1 - \mathbf{t}_2)$ is the zero vector if and only if $\mathbf{t}_1 = \mathbf{t}_2$. This proves the uniqueness. □

It is very easy to ensure that the syndrome of $\varphi\left((1+i)\mathbf{d}\right)$ contains a unit. This is because even if the error-detecting code $\mathcal{C}$ does not satisfy this condition automatically, one can always add an extra parity-check in $\mathcal{C}$ to enforce this condition.

4.4 Simulation results

In this section, we illustrate the feasibility of our blind C&F schemes through simulations. The nested lattice code $\mathcal{L}(\Lambda, \Lambda', \mathbf{d})$ is constructed from a $[1000, 500]$ binary LDPC code with column weight 3 and row weight 6 following lifted Construction A in [74]. In particular, we have $\Lambda' = 2\mathbb{Z}[i]^{1000} \subset \Lambda \subset \mathbb{Z}[i]^{1000}$ (and the message space is over $\mathbb{Z}[i]/\langle 2\rangle$). The linear error-detecting code $\mathcal{C}$ is based on the standard CRC-32 code (lifted from $\mathbb{Z}_2$ to $\mathbb{Z}[i]/\langle 2\rangle$). Clearly, the rate of $\mathcal{C}$ is $468/500 = 0.936$. The region $\mathcal{R}$ is set to $[0, \log_{10}(\mathsf{SNR})/R] \times [0, \log_{10}(\mathsf{SNR})/R]$, and the lattice-partition chain is $\mathcal{L}_j = \frac{1}{16}\log_{10}(\mathsf{SNR})(1+i)^j\mathbb{Z}[i]$ $(j = 0, \ldots, 8)$ as suggested in Sec. 4.3.1. The threshold is set to $\delta = 0.0175$.

We consider a two-transmitter, single receiver configuration, which can be viewed as a building block of a more complicated and realistic network scenario. Communication occurs in rounds. In each round, the channel gains are assumed to follow independent Rayleigh fading. A round is said to be successful if the receiver correctly recovers a linear combination. The throughput is defined as the fraction of

successful rounds in the simulation. (In other words, the throughput equals to one minus the outage probability.)

We have evaluated four blind C&F schemes through simulation by carrying out $10,000$ rounds. These four schemes apply the list-building algorithm in Sec. 4.3.1 and the probing strategies presented in Algorithms 2, 4, 5, and 6, respectively. Note that the throughputs of these schemes are the same, since they use the same probing list. Table 4.1 compares the throughput of these blind schemes with that of coherent C&F under various SNRs. It is observed that these schemes are able to approach the throughput of coherent C&F. In addition, we also have evaluated the performance of estimation-based C&F schemes with $\mathsf{SNR}_{\mathsf{est}}$ set to SNR, SNR + 3 dB, and SNR + 6 dB. We note that even if we set $\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR} + 6$ dB, our blind schemes still outperform the estimation-based scheme, which, however, requires a significant amount of overhead when the number of users is large.

Table 4.1: Throughput (%) of coherent and blind C&F schemes.

SNR	6 dB	8 dB	10 dB	12 dB	14 dB
Coherent	78.35	87.03	93.17	97.16	98.87
Blind	73.23	84.35	92.21	96.91	98.63
$\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR}$	37.46	48.86	61.96	74.03	82.41
$\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR} + 3$ dB	48.99	61.62	75.24	84.31	91.58
$\mathsf{SNR}_{\mathsf{est}} = \mathsf{SNR} + 6$ dB	58.30	72.99	83.83	91.01	95.13

We next examine the complexity of our blind C&F schemes under various SNRs. Recall that the complexity of our blind schemes is dominated by the use of $\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\cdot)$. As such, the complexity is measured by the number of uses of $\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\cdot)$. Table 4.2 compares the complexity of these blind schemes under various SNRs. It is observed that our proposed strategies, especially the quick detection strategy, are effective in controlling the complexity. Recall that, in order to identify a bad scalar, Algorithm 2 requires eight uses of $\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\cdot)$, whereas Algorithm 5 only requires two uses of $\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\cdot)$. Hence, the complexity of Algorithm 2 is roughly four times the complexity of Algorithm 5. Similarly, the complexity of Algorithm 4 is roughly eight times that of Algorithm 6.

Table 4.2: Complexity of four blind C&F schemes.

SNR	6 dB	8 dB	10 dB	12 dB	14 dB
Algorithm 2	557.61	348.95	202.44	113.57	67.68
Algorithm 4	51.68	28.77	18.06	12.46	9.33
Algorithm 5	139.92	87.75	51.09	28.86	17.40
Algorithm 6	6.83	4.01	2.69	2.01	1.63

To summarize, our simulation results suggest that for certain nested lattice codes, our blind C&F scheme using Algorithm 6 is able to approach the throughput of coherent C&F with just twice the complexity in the high-throughput region.

4.5 Summary

In this chapter, the problem of designing blind C&F schemes has been considered. A framework based on error-detection has been proposed, which eliminates the need for CSI in C&F. In particular, a generic blind C&F scheme has been developed, and several strategies have been suggested to make it computationally efficient. The effectiveness of our approach has been demonstrated through simulations. The simulation results show that our proposed blind C&F schemes can approach the throughput of coherent C&F with a modest increase in computational complexity.

We believe that there is still much work to be done in this direction, including investigating the effect of the threshold δ as well as devising more efficient probing lists based on the properties of good scalars.

Chapter 5

End-to-End Error Control

The linear labeling in Chapter 3 induce a noncoherent end-to-end network coding channel with a message space over certain finite rings. This chapter considers the problem of communication over a finite-ring matrix channel $Y = AX+BE$, where X is the channel input, Y is the channel output, E is random error, and A and B are random transfer matrices. Such a matrix channel captures the effect of decoding errors at relays. Tight capacity results are obtained and simple polynomial-complexity capacity-achieving coding schemes are provided under certain distributions of A, B, and E, extending the work of Silva, Kschischang and Kötter (2010), who handled the case of finite fields. This extension is based on several new results, which may be of independent interest, that generalize concepts and methods from matrices over finite fields to matrices over finite chain rings.

5.1 Introduction

Matrix channels provide a useful abstraction for studying error control for linear network coding schemes. Transmitted and received packets, drawn from some ambient message space Ω, can be gathered into the rows of a transmitted matrix X and a received matrix Y, respectively, while error packets injected into the network can be described by the rows of an error matrix E. Due to the nature of linear network coding, the linear transformation of transmitted packets X and the linear propagation of error packets E can be modelled as a multiplicative-additive matrix channel (MAMC), defined via

$$Y = AX + BE \tag{5.1}$$

for appropriate transfer matrices A, B. One typically assumes that A, B, and E are random matrices (drawn according to certain distributions) and independent of X. This type of stochastic model is appropriate in situations where random network coding is performed and the error matrix E arises due to decoding errors, rather than from the malicious actions of an adversary.

When the ambient space Ω is a vector space over a finite field, tight capacity bounds and simple, asymptotically capacity-achieving, coding schemes are developed in [78], under certain distributions of A, B, and E. Similar work along this line can be found, e.g., in [79–82]. Prior work on matrix channels for linear network coding has mainly focused on the finite-field case.

In this chapter, we consider a more general ambient space Ω introduced in Chapter 3 of the form

$$\Omega = T/\langle d_1 \rangle \times T/\langle d_2 \rangle \times \cdots \times T/\langle d_m \rangle, \tag{5.2}$$

where T is a sub-ring of $\mathbb{C}$ forming a principal ideal domain and $d_1, d_2, \ldots, d_m \in T$ are nonzero non-unit elements. To handle such an ambient space, we need to generalize the work of [78] from finite fields to finite chain rings. As in [78], we gather insight by first studying two variations: the noise-free multiplicative matrix channel (MMC) $Y = AX$, and the multiplication-free additive matrix channel (AMC) $Y = X + BE$.

The essential step in handling the MMC over finite fields is based on the concept of reduced row echelon form (RREF) [78]. Due to the presence of zero divisors, the extension to finite chain rings of this concept is not straightforward. Whereas over a finite field any echelon form of a matrix will have the same number of nonzero rows (equal to the matrix rank), this is not the case for matrices over finite chain rings. To address this difficulty, several possible extensions of the RREF have been proposed in the literature, including the Howell form [83, 84] and the p-basis [85]. In this chapter, we use the row canonical form defined in the dissertation of Kiermaier [86], which is itself a variant of the matrix canonical form described in an exercise in [29], and traces back to earlier ideas of Fuller [87] and Birkhoff [88]; see Section 5.3 for more details. This row canonical form is particularly suitable for studying matrix channels with an ambient space of the form (5.2). We provide a new elementary proof for the existence and uniqueness of this row canonical form. Based on these results, we introduce a notion of (combinatorially dominant) *principal* row canonical forms, which allows us to obtain simple, capacity-achieving, coding schemes for the MMC.

The key step in handling the AMC over finite fields is counting the number of matrices of a given rank t. The rank t may be regarded as a measure of "noise level" of the matrix BE. For matrices over finite chain rings, the concept of "rank" is more subtle, and must be suitably generalized. We first show how

the concept of "shape"—the appropriate chain-ring-theoretic generalization of dimension—can be used to indicate the noise level. We then derive an enumeration result that counts the number of matrices of a given shape. This enables us to obtain capacity results and simple capacity-achieving coding schemes for the MMC.

Building upon the generalizations for the two special cases, we derive tight capacity bounds and simple, polynomial-complexity, asymptotically capacity-achieving coding schemes for the MAMC model related to (5.1). We also consider several possible extensions of the MAMC model.

5.2 Motivating Examples

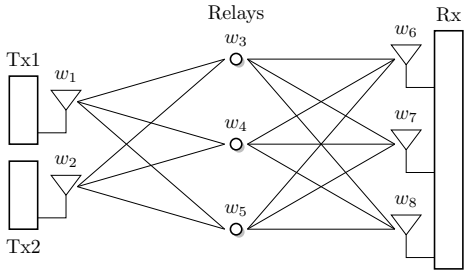


Figure 5.1: A wireless relay network with three relays.

In this section, we introduce an end-to-end matrix model for wireless relay networks with C&F. Fig. 5.1 illustrates a wireless relay network consisting of two transmitters, three relays, and a single receiver (with three antennas). Suppose that the network employs a (nested-lattice-based) C&F scheme and the packets are over some finite ring R. Let w_1, w_2 be the packets at the transmitters, and let w_6, w_7, w_8 be the packets at the receiver. Using PNC, each relay node first decodes a linear combination w_j $(j = 3, 4, 5)$ of the packets w_1, w_2, and then transmits this combination simultaneously. Hence, we have $w_j = a_{1j}w_1 + a_{2j}w_2$ for some $a_{1j}, a_{2j} \in R$, where $j = 3, 4, 5$. Similarly, $w_j = a_{3j}w_3 + a_{4j}w_4 + a_{5j}w_5$, where $j = 6, 7, 8$. Clearly, the relation between the transmitted packets and the received packets is given by $Y = AX$, where

$$X = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \; Y = \begin{bmatrix} w_6 \\ w_7 \\ w_8 \end{bmatrix}$$

and

$$A = \begin{bmatrix} a_{36} & a_{46} & a_{56} \\ a_{37} & a_{47} & a_{57} \\ a_{38} & a_{48} & a_{58} \end{bmatrix} \begin{bmatrix} a_{13} & a_{23} \\ a_{14} & a_{24} \\ a_{15} & a_{25} \end{bmatrix} \in R^{3\times 2}.$$

This gives rise to a matrix channel for the receiver.

Note that relays may sometimes introduce decoding errors. Suppose that the relay at the bottom of Fig. 5.1 makes a decoding error, i.e., $w_5 = a_{15}w_1 + a_{25}w_2 + e$, where e represents the error packet. In this case, the receiver observes $Y = AX + Z$, where A is the same as before, and

$$Z = \begin{bmatrix} a_{56} \\ a_{57} \\ a_{58} \end{bmatrix} e.$$

The above example can be generalized to a large network. Suppose that we now have n transmitters, N relays, and N receivers (each with a single antenna). Suppose that these receivers are connected to a central processor (similar to the architecture of small cells or cloud-based radio access networks). Clearly, the central processor observes a matrix channel $Y = AX + Z$, where A is of size $N \times n$.

To sum up, the matrix model $Y = AX + Z$ (over some finite ring) provides a general abstraction for studying wireless relay networks with nested-lattice-based C&F.

5.3 Row Canonical Form

The main algebraic tools for studying matrix channels over finite fields include Gaussian elimination and reduced row echelon forms. The generalization of these tools to finite chain rings is, however, not straightforward. Consider the 3×4 matrix

$$A = \begin{bmatrix} 2 & 1 & 1 & 2 \\ 6 & 3 & 7 & 2 \\ 6 & 7 & 1 & 0 \end{bmatrix}$$

over $\mathbb{Z}_8$. On the one hand, we have

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} A = \begin{bmatrix} 2 & 1 & 1 & 2 \\ 0 & 4 & 0 & 4 \\ 0 & 0 & 2 & 2 \end{bmatrix}.$$

On the other hand, we have

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 7 & 1 & 2 \end{bmatrix} A = \begin{bmatrix} 2 & 1 & 1 & 2 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

In both cases we have transformed A to echelon form using elementary row operations. Recall that, over finite fields, the rank of a matrix is precisely the number of nonzero rows in its echelon form. This property, however, does not hold for matrices over finite chain rings.

To address this difficulty, several possible generalizations of reduced row echelon forms have been proposed in the literature, including the Howell form [83, 84], the matrix canonical form [29, 87], and the p-basis [85]. In this section, we will describe a row canonical form that is particularly suitable for studying matrix channels over finite chain rings. This row canonical form is essentially the same as the reduced row echelon form defined in Kiermaier's thesis [86, Definition 2.2.2] (written in German), which itself is a variant of the matrix canonical form in [29, p. 329, Exercise XVI.7]. It appears that the key idea behind these forms was proposed by Fuller [87] based on an earlier result of Birkhoff [88]. We provide in this section a new elementary proof for the existence and uniqueness of the row canonical form.

Throughout this section, R is a (q, s) chain ring with maximal ideal $\langle \pi \rangle$. We fix a complete set of residues $\mathcal{R}(R, \pi)$ (including 0), i.e., a representation of the residue field $R/\langle \pi \rangle$, and, for $1 < l < s$, we choose the complete set of residues for π^l as

$$\mathcal{R}(R, \pi^l) = \left\{ \sum_{i=0}^{l-1} a_i \pi^i \colon a_0, \ldots, a_{l-1} \in \mathcal{R}(R, \pi) \right\}.$$

Finally, we set $\mathcal{R}(R, \pi^0) = \{0\}$.

5.3.1 Definitions

We start with a few definitions.

Let A be matrix with entries from R. The ith row of A is said to occur *above* the (i')th row of A (or the (i')th row occurs *below* the ith row) if $i < i'$. Similarly the jth column of A is said to occur *earlier* than the (j')th column (or the (j')th column occurs *later* than the jth column) if $j < j'$. This terminology extends to the entries of A: $A[i, j]$ is above $A[i', j']$ if $i < i'$ and $A[i, j]$ is earlier than $A[i', j']$ if $j < j'$. If P is some property obeyed by at least one of the entries in the ith row of A, then the *first* entry in row i with property P occurs earlier than every other entry in row i having property P.

The *pivot* of a nonzero row of a matrix is the first entry among the entries having least degree in that row. For example, 6 and 2 are the entries of least degree in the row [0 4 6 2] over $\mathbb{Z}_8$, and 6 occurs earlier. Thus, 6 is the pivot of the row [0 4 6 2]. Note that the pivot of a row is not necessarily the first nonzero entry of the row.

Definition 5.1. *A matrix A is in* row canonical form *if it satisfies the following conditions.*

1. *Nonzero rows of A are above any zero rows.*
2. *If A has two pivots of the same degree, the one that occurs earlier is above the one that occurs later. If A has two pivots of different degree, the one with smaller degree is above the one with larger degree.*
3. *Every pivot is of the form π^l for some $l \in \{0, \ldots, s-1\}$.*
4. *For every pivot (say π^l), all entries below and in the same column as the pivot are zero, and all entries above and in the same column as the pivot are elements of $\mathcal{R}(R, \pi^l)$.*

Example 5.1. *Consider the matrix*

$$A = \begin{bmatrix} 0 & 2 & 0 & \bar{1} \\ \bar{2} & 2 & 0 & 0 \\ 0 & 0 & \bar{2} & 0 \\ 0 & \bar{4} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

over $\mathbb{Z}_8$ with $\pi = 2$ and $\mathbb{Z}_8/\langle 2 \rangle = \{0, 1\}$, in which the pivots have been identified with an overline. Clearly, A satisfies all of the conditions to be in row canonical form.

The following facts follow immediately from the definition of row canonical form.

Proposition 5.1. *Let $A \in R^{n \times m}$ be a matrix in row canonical form, let p_k be the pivot of the kth row, let c_k be the index of the column containing p_k. (If the kth row is zero, let $p_k = 0$ and $c_k = 0$.) Let $d_k = \deg(p_k)$, and let $w = (w_1, \ldots, w_m)$ be an arbitrary element of* row A.

1. *Any column of A contains at most one pivot.*
2. *If A has more than one row, deleting a row of A results in a matrix also in row canonical form.*
3. *$i \geq k$ implies $\deg(A[i, j]) \geq d_k$.*
4. *($i \geq k$ and $j < c_k$) or ($i > k$ and $j \leq c_k$) implies $\deg(A[i, j]) > d_k$.*

5. p_1 *divides* $w_1, w_2, \ldots, w_m$.

6. $j < c_1$ *implies* $\deg(w_j) > d_1$.

The proof is provided in Appendix I. For any $A \in R^{n\times m}$, we say a matrix $B \in R^{n\times m}$ is a *row canonical form of* A, if (i) B is in row canonical form, and (ii) B is left-equivalent to A. We will show that any $A \in R^{n\times m}$ has a unique row canonical form. For this reason, we denote by RCF(A) *the* row canonical form of A.

5.3.2 Existence and Uniqueness

First, we demonstrate the existence of a row canonical form for any matrix A by presenting a simple algorithm that performs elementary row operations to reduce A into row canonical form. Here, the allowable elementary row operations (over R) are:

- Interchange two rows.
- Add a multiple of one row to another.
- Multiply a row by a unit in R.

Each of these operations is invertible, and so a matrix obtained from A by any sequence of these operations will have the same row span as A.

The algorithm proceeds in a series of steps. In the kth step, the algorithm selects the kth pivot, moves it to the kth row, and uses elementary row operations to reduce into row canonical form the submatrix consisting of the top k rows. The pivot selection procedure operates on any given set of rows. If the rows are all zero, the procedure should return with the result that no pivot can be found. Otherwise, among all entries of least degree in the given rows, an entry must be chosen that occurs as early as possible. This entry must certainly be the pivot of its row. The procedure should return the row and column index of the selected element.

Now we are ready to describe the algorithm in detail. In step $k = 1$, apply pivot selection to all of the rows of A. If no pivot can be found, then A is a zero matrix, and is already in row canonical form. Otherwise, we call this pivot the *first pivot* and place it in the first row by an interchange of rows (if necessary). If this pivot is not of the form π^l ($l = 0, \ldots, s-1$), we multiply the first row by a suitable unit so that the first pivot is a power of π. Note that nonzero entries in the same column below the first pivot have degrees no less than the pivot, which means that they are all multiples of the first pivot. By a sequence of elementary row operations, these entries can be cancelled, so that we arrive at a matrix,

say A_1, in which the first row is in row canonical form and all entries in the same column below the first pivot are zero. We can now increment k and proceed to the next step.

For $k \geq 2$, we apply pivot selection to the rows of A_{k-1}, excluding the first $k-1$ rows. If no pivot can be found, then the remaining rows are all zero and A_{k-1} is in row canonical form. Otherwise we call this pivot the kth pivot and place it in the kth row by an exchange of rows (if necessary). As in the first step, if this pivot is not an integer power of π, we multiply the kth row by a suitable unit so that the kth pivot is a power of π, say π^l. Nonzero entries in the same column below the kth pivot can be cancelled using elementary row operations. A nonzero entry, say a, in the same column above the kth pivot has π-adic decomposition

$$\begin{aligned} a &= a_0 + \cdots + a_{s-1}\pi^{s-1} \\ &= a_0 + \cdots + a_{l-1}\pi^{l-1} + \pi^l(a_l + \cdots + a_{s-1}\pi^{s-l-1}). \end{aligned}$$

Thus by subtracting $(a_l + \cdots + a_{s-1}\pi^{s-l-1})$ times the kth row from the row containing a, we change a to $a_0 + \cdots + a_{l-1}\pi^{l-1} \in \mathcal{R}(R, \pi^l)$, without affecting the pivot of that row. Reducing all nonzero entries in the same column as the kth pivot in this way, we arrive at a matrix, say A_k, in which the top k rows are in row canonical form and all entries in the same column below the first, second, ..., kth pivots are zero.

The above algorithm stops when no more pivots can be found. Note that, at the end of the kth step, the matrix A_k is left-equivalent to A and the submatrix formed by the top k rows of A_k is in row canonical form. It follows that the final matrix must be in row canonical form.

Therefore, we have the following result.

Proposition 5.2. *For any $A \in R^{n\times m}$, the algorithm described above computes a row canonical form of A.*

A simple count shows that this algorithm requires

$$\mathcal{O}(nm \min\{n, m\})$$

basic operations over R.

Example 5.2. *Consider the matrix*

$$A = \begin{bmatrix} 4 & 6 & 2 & \bar{1} \\ 0 & 0 & 0 & 2 \\ 2 & 4 & 6 & 1 \\ 2 & 0 & 2 & 1 \end{bmatrix}$$

over $\mathbb{Z}_8$*. There are three* 1*s in the last column of* A*, namely,* $A[1,4]$*,* $A[3,4]$ *and* $A[4,4]$*, which are the elements of least degree in* A*. We can choose any of them as the first pivot. Here, we choose* $A[1,4]$ *(indicated by an overline). After some elementary row operations, we can make the entries below the pivot zero to obtain*

$$A_1 = \begin{bmatrix} 4 & 6 & 2 & 1 \\ 0 & 4 & 4 & 0 \\ \bar{6} & 6 & 4 & 0 \\ 6 & 2 & 0 & 0 \end{bmatrix}.$$

Now consider the submatrix formed by omitting the first row of A_1*. There are four entries of least degree, namely,* $A_1[3,1] = 6$*,* $A_1[3,2] = 6$*,* $A_1[4,1] = 6$*, and* $A_1[4,2] = 2$*, among which* $A_1[3,1]$ *and* $A_1[4,1]$ *are valid choices for the second pivot. Here, we choose* $A_1[3,1]$ *(indicated by an overline). We interchange the second row and third row of* A_1*, and then multiply the new second row by* 3*, obtaining*

$$A_1' = \begin{bmatrix} 4 & 6 & 2 & 1 \\ \bar{2} & 2 & 4 & 0 \\ 0 & 4 & 4 & 0 \\ 6 & 2 & 0 & 0 \end{bmatrix}.$$

By some elementary row operations, we can make the entries below the second pivot zero. After that, we subtract 2 *times the second row from the first row, obtaining*

$$A_2 = \begin{bmatrix} 0 & 2 & 2 & 1 \\ 2 & 2 & 4 & 0 \\ 0 & \bar{4} & 4 & 0 \\ 0 & 4 & 4 & 0 \end{bmatrix}.$$

Clearly, the submatrix formed by the top two rows of A_2 *is in row canonical form. Next, consider the submatrix formed by omitting the top two rows of* A_2*. We choose the entry* $A_2[3,2]$ *(indicated by an*

overline) as the third pivot. We subtract the third row from the fourth row and obtain

$$A_3 = \begin{bmatrix} 0 & 2 & 2 & \bar{1} \\ \bar{2} & 2 & 4 & 0 \\ 0 & \bar{4} & 4 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Clearly, the submatrix formed by the top three rows of A_3 is in row canonical form (with all the pivots indicated). Since no more pivots can be found, our algorithm outputs A_3, which is indeed in row canonical form.

As expected, the row canonical form is unique.

Proposition 5.3. *For any $A \in R^{n\times m}$, the row canonical form of A is unique.*

The proof is provided in Appendix I.

5.4 Matrices under Row Constraints

In this section, we study a class of matrices in $R^{n\times m}$ whose rows are constrained to be elements of R^{μ}. We provide several new counting results and a construction of principal row canonical forms for this class of matrices. These results are of primary importance to our study of capacities and coding schemes in later sections.

5.4.1 π-adic Decomposition

Let $R^{n\times\mu}$ denote the set of matrices in $R^{n\times m}$ whose rows are elements of R^{μ}. Then the size of $R^{n\times\mu}$ is

$$|R^{n\times\mu}| = |R^{\mu}|^n = q^{n|\mu|}, \tag{5.3}$$

since there are $|R^{\mu}| = q^{|\mu|}$ choices for each row. Taking the logarithm on both sides of (5.3), we obtain

$$\log_q |R^{n\times\mu}| = n|\mu|. \tag{5.4}$$

Every matrix $X \in R^{n\times\mu}$ can be constructed based on its π-adic decomposition

$$X = X_0 + \pi X_1 + \cdots + \pi^{s-1} X_{s-1},$$

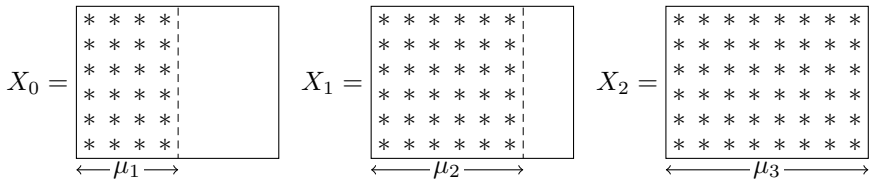


Figure 5.2: Illustration of a π-adic decomposition for $s = 3$ and $\mu = (4, 6, 8)$.

with each auxiliary matrix X_i $(i = 0, \ldots, s-1)$ satisfying:

1. $X_i[1{:}n, 1{:}\mu_{i+1}]$ is an arbitrary matrix over $\mathcal{R}(R, \pi)$, and

2. all other entries in X_i are zero.

The construction is illustrated in Fig. 5.2. Clearly, this construction provides a one-to-one mapping from sequences of $n|\mu|$ q-ary symbols to matrices in $R^{n\times\mu}$.

5.4.2 Row Canonical Forms in $\mathcal{T}_\kappa(R^{n\times\mu})$

Let $\mathcal{T}_\kappa(R^{n\times\mu})$ denote the set of matrices in $R^{n\times\mu}$ whose shape is κ. Then $|\mathcal{T}_\kappa(R^{n\times\mu})| = 0$ unless $\kappa \preceq n$ and $\kappa \preceq \mu$ (written $\kappa \preceq n, \mu$ for short). The first constraint comes from the fact that the row canonical form of a matrix in $R^{n\times\mu}$ has at most n nonzero rows. The second constraint comes from the fact that row A is a submodule of R^μ, for any $A \in R^{n\times\mu}$. Hence, we will assume that $\kappa \preceq n, \mu$ in the rest of this chapter. As we will see, the set $\mathcal{T}_\kappa(R^{n\times\mu})$, together with the row canonical forms in $\mathcal{T}_\kappa(R^{n\times\mu})$, plays a crucial role in our coding schemes.

We now enumerate the row canonical forms in $\mathcal{T}_\kappa(R^{n\times\mu})$. We need the following lemma.

Lemma 5.1. *There is a one-to-one correspondence between row canonical forms in $\mathcal{T}_\kappa(R^{n\times\mu})$ and submodules of R^μ with shape κ.*

The proof is provided in Appendix J. By Lemma 5.1, the number of row canonical forms in $\mathcal{T}_\kappa(R^{n\times\mu})$ is $\left[\!\!\left[{\mu \atop \kappa} \right]\!\!\right]_q$. It is helpful to bound this number as well as the logarithm of this number. Combining (2.3) and the fact that

$$q^{k(m-k)} \le \begin{bmatrix} m \\ k \end{bmatrix}_q \le 4q^{k(m-k)}$$

(see, e.g., [39, Lemma 4]), we have

$$q^{\sum_{i=1}^s \kappa_i(\mu_i-\kappa_i)} \le \left[\!\!\left[{\mu \atop \kappa} \right]\!\!\right]_q \le 4^s q^{\sum_{i=1}^s \kappa_i(\mu_i-\kappa_i)}. \tag{5.5}$$

Taking logarithms, we obtain

$$\sum_{i=1}^{s} \kappa_i(\mu_i - \kappa_i) \leq \log_q \left[\!\!\begin{bmatrix} \mu \\ \kappa \end{bmatrix}\!\!\right]_q \leq \sum_{i=1}^{s} \kappa_i(\mu_i - \kappa_i) + s \log_q 4. \tag{5.6}$$

Example 5.3. *Let $R = \mathbb{Z}_4$, and let $n = 2$, $\mu = (2, 3)$, $\kappa = (1, 2)$. Then by Lemma 5.1, there are* 18 *row canonical forms in $\mathcal{T}_\kappa(R^{n\times\mu})$. These* 18 *row canonical forms can be classified into* 4 *categories based on the positions of their pivots:*

$$\begin{bmatrix} 1 & * & * \\ 0 & 2 & * \end{bmatrix} \begin{bmatrix} 0 & 1 & * \\ 2 & 0 & * \end{bmatrix} \begin{bmatrix} 1 & * & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} * & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

The first category contains 8 *row canonical forms, namely,*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 2 \\ 0 & 2 & 2 \end{bmatrix}$$
$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & 2 \end{bmatrix}.$$

The second category contains 4 *row canonical forms, namely,*

$$\begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 2 & 0 & 2 \end{bmatrix} \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 2 \end{bmatrix}.$$

The third category contains 4 *row canonical forms, namely,*

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & 3 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

The fourth category contains 2 *row canonical forms, namely,*

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} 2 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}.$$

Clearly, the first category contains a significant portion of all possible row canonical forms.

Motivated by the above example, we introduce principal row canonical forms that make up a significant portion of all possible row canonical forms in $\mathcal{T}_\kappa(R^{n\times\mu})$.

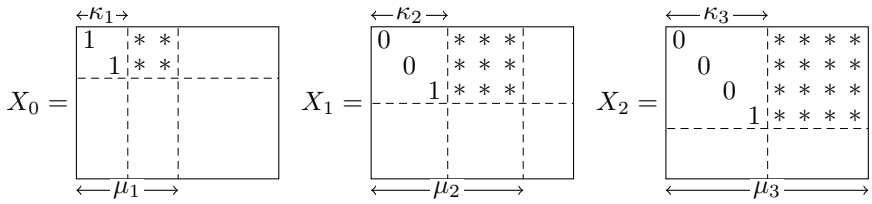


Figure 5.3: Illustration of the construction of principal row canonical forms for $\mathcal{T}_\kappa(R^{n\times\mu})$ with $s = 3$, $n = 6$, $\mu = (4, 6, 8)$, and $\kappa = (2, 3, 4)$.

A row canonical form in $\mathcal{T}_\kappa(R^{n\times\mu})$ is called *principal* if its diagonal entries $d_1, d_2, \ldots, d_r$ ($r = \min\{n, m\}$) have the following form:

$$d_1, \ldots, d_r = \underbrace{1, \ldots, 1}_{\kappa_1}, \underbrace{\pi, \ldots, \pi}_{\kappa_2-\kappa_1}, \ldots, \underbrace{\pi^{s-1}, \ldots, \pi^{s-1}}_{\kappa_s-\kappa_{s-1}}, \underbrace{0, \ldots, 0}_{r-\kappa_s}. \tag{5.7}$$

Clearly, the first category in Example 5.3 contains all principal row canonical forms for $\mathcal{T}_\kappa(\mathbb{Z}_4^{n\times\mu})$ with $n = 2$, $\mu = (2, 3)$ and $\kappa = (1, 2)$.

Proposition 5.4. *Every principal row canonical form $X \in \mathcal{T}_\kappa(R^{n\times\mu})$ can be constructed based on its π-adic decomposition*

$$X = X_0 + \pi X_1 + \cdots + \pi^{s-1} X_{s-1},$$

with each auxiliary matrix X_i ($i = 0, \ldots, s-1$) satisfying the following conditions:

1. $X_i[1{:}\kappa_{i+1}, 1{:}\kappa_{i+1}] = \mathrm{diag}(\underbrace{0, \ldots, 0}_{\kappa_i}, \underbrace{1, \ldots, 1}_{\kappa_{i+1}-\kappa_i})$,
2. $X_i[1{:}\kappa_{i+1}, \kappa_{i+1}+1{:}\mu_{i+1}]$ *can be any matrix over $\mathcal{R}(R, \pi)$, and*
3. *all other entries in X_i are zero.*

The proof is provided in Appendix J. The construction is illustrated in Fig. 5.3. Clearly, this construction provides a one-to-one mapping from sequences of $\sum_{i=1}^{s} \kappa_i(\mu_i - \kappa_i)$ q-ary symbols to principal row canonical forms in $\mathcal{T}_\kappa(R^{n\times\mu})$. Note that the number of principal row canonical forms in $\mathcal{T}_\kappa(R^{n\times\mu})$ is $q^{\sum_{i=1}^{s} \kappa_i(\mu_i-\kappa_i)}$, which is comparable to the number of row canonical forms in $\mathcal{T}_\kappa(R^{n\times\mu})$ in total.

5.4.3 General Matrices in $\mathcal{T}_\kappa(R^{n\times\mu})$

Next, we count the number of matrices in $R^{n\times\mu}$ of shape κ, which is a central result in this section. The proof is provided in Appendix J.

Theorem 5.1. *The size of $\mathcal{T}_\kappa(R^{n\times\mu})$ is given by*

$$|\mathcal{T}_\kappa(R^{n\times\mu})| = |R^{n\times\kappa}| \prod_{i=0}^{\kappa_s-1}(1-q^{i-n}) \begin{bmatrix}\mu\\ \kappa\end{bmatrix}_q. \tag{5.8}$$

In particular, when the chain length $s = 1$, R becomes $\mathbb{F}_q$, and this counting result becomes $\prod_{i=0}^{\kappa_1-1}(q^n - q^i)\begin{bmatrix}\mu_1\\ \kappa_1\end{bmatrix}_q$, which is the number of $n \times \mu_1$ matrices of rank κ_1. We note that Theorem 5.1 generalizes a theorem of [89] from square matrices to general matrices and from Galois rings to finite chain rings.

Taking logarithms on both sides of (5.8), we have

$$\log_q |\mathcal{T}_\kappa(R^{n\times\mu})| = \log_q \begin{bmatrix}\mu\\ \kappa\end{bmatrix}_q + \log_q |R^{n\times\kappa}| + \log_q \prod_{i=0}^{\kappa_s-1}(1-q^{i-n}).$$

Combining this with (5.4) and (5.6), we obtain

$$\sum_{i=1}^{s}\kappa_i(n+\mu_i-\kappa_i) + \log_q \prod_{i=0}^{\kappa_s-1}(1-q^{i-n}) \leq \log_q |\mathcal{T}_\kappa(R^{n\times\mu})| \leq \sum_{i=1}^{s}\kappa_i(n+\mu_i-\kappa_i) + \log_q \prod_{i=0}^{\kappa_s-1}(1-q^{i-n}) + s\log_q 4. \tag{5.9}$$

5.4.4 Notational Summary

Table 5.1 summarizes the notation that will be used extensively in the study of matrix channels. Also listed are finite-field counterparts, which facilitates comparisons of this chapter with [78].

Table 5.1: Notational Summary

notation	meaning	finite-field counterpart
μ	shape	rank
R^μ	R-module	vector space $\mathbb{F}_q^m$
$R^{n\times\mu}$	set of matrices with rows from R^μ	$\mathbb{F}_q^{n\times m}$
$\mathcal{T}_\kappa(R^{n\times\mu})$	set of matrices in $R^{n\times\mu}$ with shape κ	set of matrices in $\mathbb{F}_q^{n\times m}$ with rank t
RCF(A)	row canonical form of A	reduced row echelon form

5.5 Channel Decomposition

In this section, we introduce a channel decomposition technique that converts a matrix channel over certain finite rings into a set of independent parallel matrix channels over finite chain rings. This enables us to focus on matrix channels over finite chain rings, thereby greatly facilitating our study of capacity results and coding schemes in later sections.

As shown in our previous work [74], nested-lattice-based C&F induces a message space of the form $\Omega = T/\langle d_1\rangle \times \cdots \times T/\langle d_m\rangle$, where T is a PID and $d_m \mid \cdots \mid d_1$. Let $R \triangleq T/\langle d_1\rangle$. (Note that R is a PIR, but not necessarily a finite chain ring.) We can rewrite Ω as

$$\Omega = R \times (d_1/d_2)R \times \cdots \times (d_1/d_m)R;$$

this expression says that Ω can be viewed as a collection of m-tuples (over R) whose jth component is a multiple of d_1/d_j.

Example 5.4. *Let $\Omega = \mathbb{Z}_{12} \times \mathbb{Z}_6 \times \mathbb{Z}_6 \times \mathbb{Z}_2$. Then Ω can be expressed as $\mathbb{Z}_{12} \times 2\mathbb{Z}_{12} \times 2\mathbb{Z}_{12} \times 6\mathbb{Z}_{12}$ via the following map:*

$$(a_1 + (12), a_2 + (6), a_3 + (6), a_4 + (2)) \to (a_1, 2a_2, 2a_3, 6a_4),$$

where $a_1 \in \{0, \ldots, 11\}$, $a_2, a_3 \in \{0, \ldots, 5\}$, and $a_4 \in \{0, 1\}$. Clearly, this map is one-to-one.

With this expression, our matrix channel can be written as

$$Y = AX + BE \tag{5.10}$$

where $X \in R^{n\times m}$ and $Y \in R^{N\times m}$ are the input and output matrices whose rows are from Ω, $E \in R^{t\times m}$ is the error matrix whose rows (also from Ω) correspond to additive (random) error packets. The transfer matrices $A \in R^{N\times n}$ and $B \in R^{N\times t}$ are random matrices with some joint distribution, and X, (A, B), E are statistically independent. For simplicity of presentation, we sometimes write the channel model as $Y = AX + Z$, where $Z = BE$ is called the noise matrix. Clearly, the channel model is an instance of the discrete memoryless channel $(\mathcal{X}, p_{Y|X}, \mathcal{Y})$ with input alphabet $\mathcal{X} = R^{n\times m}$, output alphabet $\mathcal{Y} = R^{N\times m}$ and channel transition probability $p_{Y|X}$. The capacity of this channel is given by

$$C = \max_{p_X} I(X; Y)$$

where p_X is the input distribution.

Next, we illustrate how to decompose the matrix channel. To this end, we first decompose the message space Ω. Since T is a PID, $d_1 \in T$ can be factored as $d_1 = u_1 p_1^{t_{1,1}} \cdots p_L^{t_{L,1}}$, where u_1 is a unit in T, $p_1, \ldots, p_L$ are primes in T, and $t_{1,1}, \ldots, t_{L,1}$ are positive integers. Since $d_m \mid \cdots \mid d_1$, we have $d_j = u_j p_1^{t_{1,j}} \cdots p_L^{t_{L,j}}$ $(j = 2, \ldots, m)$, where u_j is a unit, and $t_{1,j}, \ldots, t_{L,j}$ are non-negative integers. Now, let

$$\Omega_\ell \triangleq T/\langle p_\ell^{t_{\ell,1}} \rangle \times \cdots \times T/\langle p_\ell^{t_{\ell,m}} \rangle, \ \ell = 1, \ldots, L.$$

By the Chinese remainder theorem, we have $\Omega \cong \Omega_1 \times \cdots \times \Omega_L$. This gives rise to a decomposition of Ω.

Example 5.5. *Let* $\Omega = \mathbb{Z}_{12} \times \mathbb{Z}_6 \times \mathbb{Z}_6 \times \mathbb{Z}_2$. *Then*

$$\begin{aligned} \Omega &\cong (\mathbb{Z}_4 \times \mathbb{Z}_3) \times (\mathbb{Z}_2 \times \mathbb{Z}_3) \times (\mathbb{Z}_2 \times \mathbb{Z}_3) \times \mathbb{Z}_2 \\ &\cong (\underbrace{\mathbb{Z}_4 \times \mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2}_{\Omega_1}) \times (\underbrace{\mathbb{Z}_3 \times \mathbb{Z}_3 \times \mathbb{Z}_3}_{\Omega_2}). \end{aligned}$$

Note that Ω_ℓ has an interesting interpretation: Ω_ℓ is a natural projection of Ω onto some finite chain ring. Let $R_\ell \triangleq T/\langle p_\ell^{t_{\ell,1}} \rangle$ (which is a finite chain ring). It is easy to check that $\Omega_\ell = R_\ell \times p_\ell^{(t_{\ell,1} - t_{\ell,2})} R_\ell \times \cdots \times p_\ell^{(t_{\ell,1} - t_{\ell,m})} R_\ell$ and that

$$\Omega_\ell = \{(r_1, \ldots, r_m) \bmod R_\ell \mid (r_1, \ldots, r_m) \in \Omega\}.$$

We are now ready to introduce the channel decomposition. For any matrix $X \in R^{n \times m}$, let $X^{[\ell]} \triangleq X \bmod R_\ell$, the projection of every entry of X onto R_ℓ. Applying this projection to the matrix channel, we obtain L sub-channels

$$Y^{[\ell]} = A^{[\ell]} X^{[\ell]} + Z^{[\ell]}, \tag{5.11}$$

for $\ell = 1, \ldots, L$, as illustrated in Fig. 5.4. Clearly, each row of $X^{[\ell]}$ (or, $Y^{[\ell]}$, $Z^{[\ell]}$) is from Ω_ℓ.

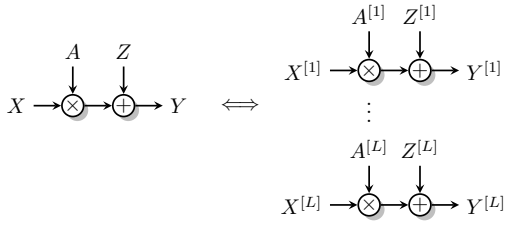


Figure 5.4: An illustration of the channel decomposition.

These sub-channels are, in general, correlated with each other. Hence, we have $C \geq \sum_{\ell=1}^{L} C_\ell$, where

C_ℓ is the capacity of sub-channel ℓ. The equality is achieved for certain distributions of A and Z. One such distribution is provided in Theorem 5.2. We need a few definitions. We say a matrix $A \in R^{n\times m}$ have *rank* t, if for all ℓ, $A^{[\ell]}$ has rank t. A matrix $A \in R^{n\times m}$ is *full rank* if $\operatorname{rank} A = \min\{n, m\}$.

Theorem 5.2. *Suppose that the transfer matrix $A \in R^{N\times n}$ ($N \geq n$) is uniform over all full-rank matrices and that the noise matrix $Z \in R^{N\times m}$ is uniform over all rank-t matrices (whose rows are from Ω). Suppose that A and Z are independent of each other. Then the channel decomposition induces L independent sub-channels*

$$Y^{[\ell]} = A^{[\ell]}X^{[\ell]} + Z^{[\ell]},\ \ell = 1, \ldots, L,$$

where $A^{[\ell]} \in R_\ell^{N\times n}$ is uniform over full-rank matrices (over R_ℓ), $Z^{[\ell]}$ is uniform over rank-t matrices whose rows are from Ω_ℓ, and $A^{[\ell]}$ is independent of $Z^{[\ell]}$. Clearly, these sub-channels form a product discrete memoryless channel (DMC). In particular, the capacity of this product DMC is $C = \sum_{\ell=1}^{L} C_\ell$.

Proof. Note that A is full rank over R, if and only if each $A^{[\ell]}$ is full rank over R_ℓ. Hence, the number of full-rank matrices in $R^{N\times n}$ is equal to the product of the number of full-rank matrices in $R_\ell^{N\times n}$ ($\ell = 1, \ldots, L$). In particular, it follows that when A is uniform over full-rank matrices, each $A^{[\ell]}$ is also uniform over full-rank matrices and independent of each other. Similarly, each $Z^{[\ell]}$ is uniform over rank-t matrices and independent of each other. Since $A^{[\ell]}$ and $Z^{[\ell]}$ are projections of A and Z, respectively, $A^{[\ell]}$ and $Z^{[\ell]}$ are independent. Therefore, the sub-channels $Y^{[\ell]} = A^{[\ell]}X^{[\ell]} + Z^{[\ell]}$ are independent of each other. In particular, $C = \sum_{\ell=1}^{L} C_\ell$. □

Theorem 5.2 says that when A and Z follow certain distributions, the channel decomposition incurs no loss of information. Hence, in this case, it suffices to study each sub-channel independently.

Next, we comment on the assumptions in Theorem 5.2. First, as we will soon see in later sections, these assumptions allow us to derive clean capacity results and simple coding schemes, based on which more general distributions can be studied (see Section 5.9).

Second, we note that the full-rank assumption on A and the rank-t assumption on Z are reasonable, when the system size is large. To see this, observe that the portion of full-rank matrices in $R^{N\times n}$ is lower-bounded by

$$1 - \sum_{\ell=1}^{L} \frac{n}{|p_\ell|^{2(1+N-n)}}.$$

Clearly, this lower bound tends to 1 as n and N grow. For example, if we set $n = 100$, $N = 110$, and choose $R = \mathbb{Z}_2[i] = \mathbb{Z}[i]/\langle(1+i)^2\rangle$, then the lower bound is around 0.999976. Using the same argument, we can show that rank-t matrices make up a significant portion of all possible noise matrices $Z = BE$ for large t, m, and N.

Third, we note that the uniformness assumptions on A and Z provide us with "worst-case" scenarios, which will be elaborated in Section 5.9.

Without loss of generality, we will focus on the case $L = 1$, and so R is a finite chain ring for the remainder of the chapter. Suppose that R be a (q, s) chain ring. Let μ be the shape of Ω. Then, we can write $X \in R^{n\times\mu}$ and $Y, Z \in R^{N\times\mu}$. That is, we may think of the rows of X, Y and Z as packets over the ambient space R^{μ}. (To support this ambient space, the length of a packet, denoted by m, is equal to μ_s.)

In many situations, it is useful to understand the capacity scaling as the system size and packet length grow. For that reason, we introduce a notion of asymptotic capacity

$$\bar{C} = \lim_{m\to\infty} \frac{1}{n|\mu|} C = \lim_{m\to\infty} \frac{1}{\bar{n}|\bar{\mu}|m^2} C,$$

where we assume that $\bar{n} = n/m$ and $\bar{\mu} = (\bar{\mu}_1, \ldots, \bar{\mu}_s) = \mu/m$ are fixed. Here, logarithms are taken to the base q, so that the capacity C is given in q-ary units per channel use and that $\bar{C}$ is normalized such that $\bar{C} = 1$ if the channel is noiseless (i.e., $A = I$ and $Z = 0$).

5.6 The Multiplicative Matrix Channel

As a first special case, following [78], we consider the *multiplicative matrix channel (MMC)* defined by the law

$$Y = AX,$$

where $A \in R^{N\times n}$ is uniform over all full-column-rank matrices and independent from $X \in R^{n\times\mu}$. This model is a special case of the channel model (5.11) with $Z = 0$.

5.6.1 Capacity

The capacity of the MMC can be obtained by investigating the channel transition probabilities. Since full-column-rank matrices preserve the row span, we have $\operatorname{row} X = \operatorname{row} Y$. It follows that the channel transition probability $p_{Y|X}(Y|X) > 0$ if and only if $\operatorname{row} X = \operatorname{row} Y$. Moreover, we have the following lemma:

Lemma 5.2. *The channel transition probabilities satisfy the following two properties.*

1. $p_{Y|X}(Y_1|X) = p_{Y|X}(Y_2|X) > 0$, *if* $\operatorname{row} X = \operatorname{row} Y_1 = \operatorname{row} Y_2$.
2. $p_{Y|X}(Y|X_1) = p_{Y|X}(Y|X_2) > 0$, *if* $\operatorname{row} X_1 = \operatorname{row} X_2 = \operatorname{row} Y$.

Proof. Since $\operatorname{row} Y_1 = \operatorname{row} Y_2$, there exists some invertible matrix P such that $PY_1 = Y_2$. Let $\mathcal{A}_j = \{A \in \mathcal{T}_n(R^{N\times n}) \mid AX = Y_j\}$ be the set of transfer matrices such that $AX = Y_j$. Then $\mathcal{A}_1$ and $\mathcal{A}_2$ have the same size (i.e., $|\mathcal{A}_1| = |\mathcal{A}_2|$), because $A \in \mathcal{A}_1$ if and only if $PA \in \mathcal{A}_2$. Hence, we have $p_{Y|X}(Y_1|X) = p_{Y|X}(Y_2|X)$. In particular, when $\operatorname{row} X = \operatorname{row} Y_1$, the set $\mathcal{A}_1$ is non-empty, and so $p_{Y|X}(Y_1|X) > 0$. This proves Part 1). Similarly, we can prove Part 2). □

Lemma 5.2 characterizes the structure of the channel transition probabilities, based on which one can show that the capacity only depends on the number of all possible submodules generated by X.

Theorem 5.3. *The capacity of the MMC, in q-ary symbols per channel use, is given by*

$$C_{MMC} = \log_q \sum_{\lambda \preceq n,\mu} \begin{bmatrix} \mu \\ \lambda \end{bmatrix}_q .$$

A capacity-achieving code $\mathcal{C} \subseteq R^{n\times\mu}$ consists of all possible row canonical forms in $R^{n\times\mu}$.

Theorem 5.3 suggests that information should be encoded in the choice of submodules. That is, "transmission via submodules" is optimal here. This naturally generalizes the "transmission via subspaces" strategy in [39].

Corollary 5.1. *The capacity C_{MMC} is bounded by*

$$\sum_{i=1}^{s} \kappa_i(\mu_i - \kappa_i) \le C_{MMC} \le \sum_{i=1}^{s} \kappa_i(\mu_i - \kappa_i) + \log_q 4^s \begin{pmatrix} n+s \\ s \end{pmatrix} \tag{5.12}$$

where $\kappa_i = \min\{n, \lfloor \mu_i/2 \rfloor\}$ for all i.

Proof. First, since $\kappa = (\kappa_1, \ldots, \kappa_s) \preceq n, \mu$, we have

$$\begin{aligned} C_{\mathrm{MMC}} &= \log_q \sum_{\lambda \preceq n,\mu} \begin{bmatrix} \mu \\ \lambda \end{bmatrix}_q \\ &\ge \log_q \begin{bmatrix} \mu \\ \kappa \end{bmatrix}_q \\ &\ge \sum_{i=1}^{s} \kappa_i(\mu_i - \kappa_i), \end{aligned}$$

where the second inequality follows from (5.6).

Second, we have

$$\begin{aligned}
C_{\text{MMC}} &= \log_q \sum_{\lambda \preceq n,\mu} \begin{bmatrix} \mu \\ \lambda \end{bmatrix}_q \\
&\leq \log_q \sum_{\lambda \preceq n,\mu} 4^s q^{\sum_i \lambda_i(\mu_i - \lambda_i)} \\
&\leq \log_q \sum_{\lambda \preceq n,\mu} 4^s q^{\sum_i \kappa_i(\mu_i - \kappa_i)} \\
&\leq \log_q 4^s \binom{n+s}{s} q^{\sum_i \kappa_i(\mu_i - \kappa_i)} \\
&= \sum_{i=1}^{s} \kappa_i(\mu_i - \kappa_i) + \log_q 4^s \binom{n+s}{s}.
\end{aligned}$$

where the first inequality follows from (5.6), the second inequality follows from the fact that κ maximizes the quantity $\sum_i \lambda_i(\mu_i - \lambda_i)$ subject to the constraint $\lambda \preceq n, \mu$, and the third inequality follows from the fact that the number of shapes satisfying $\lambda \preceq n, \mu$ is upper-bounded by $\binom{n+s}{s}$. □

We next turn to the asymptotic capacity of the MMC.

Theorem 5.4. *The asymptotic capacity $\bar{C}_{MMC}$ is given by*

$$\bar{C}_{MMC} = \frac{\sum_{i=1}^{s} \bar{\kappa}_i(\bar{\mu}_i - \bar{\kappa}_i)}{\bar{n}|\bar{\mu}|}, \tag{5.13}$$

where $\bar{\kappa} = \kappa/m$ with $\kappa_i = \min\{n, \lfloor \mu_i/2 \rfloor\}$ for all i.

Proof. This follows from Corollary 5.1 and the fact that $\frac{1}{m^2} \log_q 4^s \binom{n+s}{s} \to 0$, as $m \to \infty$. □

Theorem 5.4 implies that the shape κ given by $\kappa_i = \min\{n, \lfloor \mu_i/2 \rfloor\}$ $(1 \leq i \leq s)$ is "typical" among the shapes of all possible row canonical forms in $R^{n \times \mu}$. In other words, the row canonical forms of shape κ make up a significant portion of all possible row canonical forms. Hence, the transmitter may encode information in the choice of row canonical forms of shape κ instead of all row canonical forms.

5.6.2 A Simple Coding Scheme

In this section, we present a simple coding scheme that achieves the asymptotic capacity in Theorem 5.4. The key idea is to make the codebook the set of all principal row canonical forms for $\mathcal{T}_\kappa(R^{n \times \mu})$. In other words, we employ two "reductions" in the code construction. First, we move from all row canonical forms in $R^{n \times \mu}$ to all row canonical forms in $\mathcal{T}_\kappa(R^{n \times \mu})$, as suggested by Theorem 5.4. Then, we move from all row canonical forms in $\mathcal{T}_\kappa(R^{n \times \mu})$ to all principal row canonical forms in $\mathcal{T}_\kappa(R^{n \times \mu})$. With these two

reductions, our coding scheme not only achieves the asymptotic capacity, but also admits fast encoding and decoding.

Encoding

The input matrix X is chosen from the set of principal row canonical forms for $\mathcal{T}_\kappa(R^{n\times\mu})$ by using the construction presented in Section 5.4.2. Clearly, the encoding rate of the scheme is $R_{\text{MMC}} = \sum_{i=1}^{s} \kappa_i(\mu_i - \kappa_i)$.

Decoding

Upon receiving $Y = AX$, the decoder simply computes the row canonical form of Y. The decoding is always correct by the uniqueness of the row canonical form. By comparing the encoding rate with the asymptotic capacity, we have the following theorem.

Theorem 5.5. *The coding scheme described above achieves the asymptotic capacity (5.13).*

5.7 The Additive Matrix Channel

In this section, we consider the *additive matrix channel (AMC)* defined by the law

$$Y = X + Z,$$

where Z is uniform over $\mathcal{T}_\tau(R^{n\times\mu})$ and independent from X. This model is a special case of the channel model (5.11) with $A = I$.

5.7.1 Capacity

Theorem 5.6. *The capacity of the AMC, in q-ary symbols per channel use, is given by*

$$C_{AMC} = \log_q |R^{n\times\mu}| - \log_q |\mathcal{T}_\tau(R^{n\times\mu})|,$$

achieved by the uniform input distribution.

Proof. The AMC is an example of a symmetric discrete memoryless channel, whose capacity is achieved by the uniform input distribution. Note that when X is uniform over $R^{n\times\mu}$, so is Y. Thus, we have

$$C_{\text{AMC}} = H(Y) - H(Z) = \log_q |R^{n\times\mu}| - \log_q |\mathcal{T}_\tau(R^{n\times\mu})|.$$

□

Corollary 5.2. *The capacity C_{AMC} is bounded by*

$$\sum_{i=1}^{s}(n-\tau_i)(\mu_i-\tau_i)-\log_q 4^s\prod_{i=0}^{\tau_s-1}(1-q^{i-n})<C_{AMC}<\sum_{i=1}^{s}(n-\tau_i)(\mu_i-\tau_i)-\log_q\prod_{i=0}^{\tau_s-1}(1-q^{i-n}).$$

Proof. It follows immediately from Theorem 5.6 and (5.4), (5.9). □

We next turn to the asymptotic behavior of the AMC.

Theorem 5.7. *The asymptotic capacity $\bar{C}_{AMC}$ is given by*

$$\bar{C}_{AMC}=\frac{\sum_{i=1}^{s}(\bar{n}-\bar{\tau}_i)(\bar{\mu}_i-\bar{\tau}_i)}{\bar{n}|\bar{\mu}|}. \tag{5.14}$$

Proof. It follows from Corollary 5.2 and the fact that

$$\frac{1}{m^2}\log_q 4^s\prod_{i=0}^{\tau_s-1}(1-q^{i-n})\to 0,\ \text{as } m\to\infty.$$

□

5.7.2 Coding Scheme

We focus on a special case when $\tau=t$, and present a coding scheme based on the idea of error-trapping in [78]. This scheme achieves the asymptotic capacity for this special case.

Encoding

Set $v\geq t$. The input matrix X is constructed as

$$X=\begin{bmatrix}0 & 0\\ 0 & U\end{bmatrix},$$

where the size of U is $(n-v)\times(m-v)$, and the sizes of other zero matrices are chosen to make X an $n\times m$ matrix. Here, U is chosen from the set $R^{(n-v)\times(\mu-v)}$ by using the construction in Section 5.4 (as illustrated in Fig. 5.5). Clearly, the encoding rate of the scheme is $R_{\text{AMC}}=\sum_{i=1}^{s}(n-v)(\mu_i-v)$.

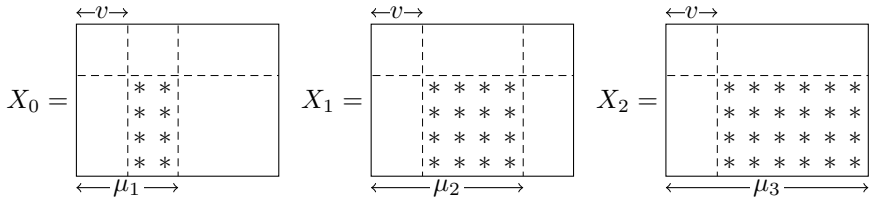


Figure 5.5: Illustration of the AMC encoding scheme for $s = 3$, $n = 6$, $\mu = (4, 6, 8)$, and $v = 2$.

Decoding

Following [78], we write the noise matrix Z as

$$Z = BE = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{bmatrix} E_1 & E_2 \end{bmatrix},$$

where $B_1 \in R^{v \times t}$, $B_2 \in R^{(n-v) \times t}$, $E_1 \in R^{t \times v}$ and $E_2 \in R^{t \times (m-v)}$. The received matrix Y is then given by

$$Y = X + Z = \begin{bmatrix} B_1 E_1 & B_1 E_2 \\ B_2 E_1 & U + B_2 E_2 \end{bmatrix}.$$

Similar to [78], we define that the error trapping is successful if $\operatorname{shape} B_1 E_1 = t$. Assume that this is the case. Then by Proposition 2.1.3, we have $\operatorname{shape} B_1 = \operatorname{shape} E_1 = t$. Consider the submatrix consisting of the first v columns of Y. Since $\operatorname{shape} B_1 E_1 = t$, the rows of $B_2 E_1$ are completely spanned by the rows of $B_1 E_1$. That is, $\operatorname{row} B_2 E_1 \subseteq \operatorname{row} B_1 E_1$. Thus, there exists some matrix $\bar{T}$ such that $B_2 E_1 = \bar{T} B_1 E_1$. Since E_1 is full row rank, by Lemma 2.1, $B_2 E_1 = \bar{T} B_1 E_1$ implies $B_2 = \bar{T} B_1$. It follows that

$$T \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = \begin{bmatrix} B_1 \\ 0 \end{bmatrix}, \text{ where } T = \begin{bmatrix} I & 0 \\ -\bar{T} & I \end{bmatrix}.$$

Note also that $TX = X$. Thus,

$$TY = TX + TZ = \begin{bmatrix} B_1 E_1 & B_1 E_2 \\ 0 & U \end{bmatrix},$$

from which the data matrix U is readily obtained.

The decoding is summarized as follows. The decoder observes $B_1 E_1$, $B_1 E_2$, and $B_2 E_1$ thanks to the error traps. The decoder then checks the condition $\operatorname{shape} B_1 E_1 = t$. If the condition does not hold, the decoder declares a failure. Otherwise, the decoder finds a matrix $\bar{T}$ such that $B_2 E_1 = \bar{T} B_1 E_1$ (which

means $B_2 = \bar{T}B_1$). Since $B_2 = \bar{T}B_1$, the decoder can recover B_2E_2 by using the relation $B_2E_2 = \bar{T}B_1E_2$. Clearly, the error probability of the scheme is zero. The failure probability of the scheme is

$$P_f = \Pr[\text{shape}\, B_1E_1 \neq t].$$

Lemma 5.3. *The failure probability P_f of the above scheme is upper-bounded by $P_f < \frac{2t}{q^{1+v-t}}$.*

Proof. If B_1 and E_1 are full rank, then $\text{shape}\, B_1E_1 = t$. Hence, by the union bound, the failure probability

$$P_f \leq \Pr[E_1 \text{ is not full rank}] + \Pr[B_1 \text{ is not full rank}].$$

Now consider the probability that E_1 is full rank. Recall that $E \in R^{t\times\mu}$ is a full-rank matrix chosen uniformly at random. An equivalent way of generating E is to first generate the entries of a matrix $E' \in R^{t\times\mu}$ uniformly at random, and then discard E' if it is not full rank. This suggests that

$$\begin{aligned}\Pr[E_1 \text{ is full rank}] &= \Pr[E_1' \text{ is full rank} \mid E' \text{ is full rank}] \\ &> \Pr[E_1' \text{ is full rank}],\end{aligned}$$

where E_1' consists of the first v columns of E'. Thus,

$$\begin{aligned}\Pr[E_1 \text{ is full rank}] &> |\mathcal{T}_t(R^{t\times v})|/|R^{t\times v}| \\ &= q^{stv}\prod_{i=0}^{t-1}(1-q^{i-v})/q^{stv} \\ &= \prod_{i=0}^{t-1}(1-q^{i-v}) \\ &> 1 - \frac{t}{q^{1+v-t}}.\end{aligned}$$

Similarly, we can show that

$$\Pr[B_1 \text{ is full rank}] > 1 - \frac{t}{q^{1+v-t}}.$$

Therefore, the failure probability $P_f < \frac{2t}{q^{1+v-t}}$. □

Recall that the encoding rate of the scheme is $R_{\text{AMC}} = \sum_{i=1}^{s}(n-v)(\mu_i - v)$. Thus, if we set v such that

$$v - t \to \infty, \text{ and } \frac{v-t}{m} \to 0,$$

as $m \to \infty$, then we have $P_f \to 0$ and $\bar{R}_{\text{AMC}} = \frac{R_{\text{AMC}}}{n|\mu|} \to \bar{C}_{\text{AMC}}$. Therefore, we have the following

theorem.

Theorem 5.8. *The coding scheme described above can achieve the capacity expression (5.14) for the special case when $\tau = t$.*

Remark: The general case can also be handled by combining the above scheme with the successive cancellation technique.

5.8 The Multiplicative-Additive Matrix Channel

In this section, we consider the *multiplicative-additive matrix channel (MAMC)* defined by the law

$$Y = AX + Z,$$

where $A \in \mathcal{T}_n(R^{N\times n})$ and $Z \in \mathcal{T}_\tau(R^{N\times\mu})$ are uniformly distributed and independent from any other variables.

5.8.1 Capacity Bounds

Since A is uniform over $\mathcal{T}_n(R^{N\times n})$, A is statistically equivalent to $P\begin{bmatrix}0\\ I_n\end{bmatrix}$, where $P \in R^{N\times N}$ is uniform over $\mathrm{GL}_N(R)$, $I_n \in R^{n\times n}$ is an identity matrix, and $0 \in R^{(N-n)\times n}$ is a zero matrix. Hence, we have

$$Y = P\begin{bmatrix}0\\ I_n\end{bmatrix}X + Z = P\begin{bmatrix}0\\ X\end{bmatrix} + Z = P\left(\begin{bmatrix}0\\ X\end{bmatrix} + W\right),$$

where $W = P^{-1}Z$ is uniform over $\mathcal{T}_\tau(R^{N\times\mu})$ and independent of X.

Theorem 5.9. *The capacity of the MAMC, in q-ary symbols per channel use, is upper-bounded by*

$$C_{AMMC} \le \log_q \sum_{\lambda\preceq N, n+\tau, \mu} \begin{bmatrix}\mu\\ \lambda\end{bmatrix}_q - \log_q |\mathcal{T}_\tau(R^{N\times\mu})| + \log_q \sum_{\tau'\preceq\tau} |\mathcal{T}_{\tau'}(R^{N\times\min\{n+\tau_s, N\}})|. \tag{5.15}$$

Proof. Let $U = \begin{bmatrix}0\\ X\end{bmatrix} + W$. Then $Y = PU$, and X, U, Y form a Markov chain. Hence, $I(X;Y|U) = 0$.

Using the chain rules, we have

$$\begin{aligned} I(X;Y) &= I(U;Y) - I(U;Y|X) + \underbrace{I(X;Y|U)}_{=0} \\ &= I(U;Y) - H(U|X) + H(U|X,Y) \\ &= I(U;Y) - H(W) + H(W|X,Y) \\ &= I(U;Y) - \log_q |\mathcal{T}_\tau(R^{N\times\mu})| + H(W|X,Y) \end{aligned}$$

Next, we upper bound the terms $I(U;Y)$ and $H(W|X,Y)$. Since $\operatorname{shape} U \preceq N, n+\tau$, the row span $\operatorname{row} U$ has at most $\sum_{\lambda\preceq N,n+\tau,\mu} \begin{bmatrix}\mu\\ \lambda\end{bmatrix}_q$ choices. Hence, $I(U;Y) \le \log_q \sum_{\lambda\preceq N,n+\tau,\mu} \begin{bmatrix}\mu\\ \lambda\end{bmatrix}_q$.

Let $\kappa = \operatorname{shape} Y$. Let S be the Smith normal form of Y. Then S contains κ_s nonzero diagonal entries. Thus, Y can be expressed as

$$Y = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} S_{11} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} = P_1 S_{11} Q_1,$$

where $P_1 \in R^{N\times\kappa_s}$, $Q_1 \in R^{\kappa_s\times m}$, and $S_{11} \in R^{\kappa_s\times\kappa_s}$.

Note that

$$\begin{bmatrix} 0 \\ X \end{bmatrix} + W = P^{-1}Y = P^* Q_1,$$

where $P^* = P^{-1}P_1 S_{11}$. Since Q_1 consists of the first κ_s rows of an invertible matrix Q, Q_1 is a full-rank matrix. In particular, Q_1 contains an invertible $\kappa_s \times \kappa_s$ submatrix. By reordering columns if necessary, we can assume that the left $\kappa_s \times \kappa_s$ submatrix of Q_1 is invertible. Write $Q_1 = \begin{bmatrix} Q_{11} & Q_{12} \end{bmatrix}$, $X = \begin{bmatrix} X_1 & X_2 \end{bmatrix}$ and $W = \begin{bmatrix} W_1 & W_2 \end{bmatrix}$, where Q_{11}, X_1, and W_1 have κ_s columns. We have

$$\begin{bmatrix} 0 & 0 \\ X_1 & X_2 \end{bmatrix} + \begin{bmatrix} W_1 & W_2 \end{bmatrix} = \begin{bmatrix} P^*Q_{11} & P^*Q_{12} \end{bmatrix}.$$

It follows that

$$P^* = \left(\begin{bmatrix} 0 \\ X_1 \end{bmatrix} + W_1 \right) Q_{11}^{-1} \text{ and } W_2 = P^* Q_{12} - \begin{bmatrix} 0 \\ X_2 \end{bmatrix}.$$

This suggests that W_2 can be computed from W_1 if X and Y are known. Thus,

$$H(W|X,Y) = H(W_1|X,Y) \le H(W_1|\operatorname{shape} Y).$$

Since W_1 is an $N \times \kappa_s$ matrix with $\operatorname{shape} W_1 \preceq \tau$, we have

$$H(W_1|\operatorname{shape} Y = \kappa) \le \log_q \sum_{\tau' \preceq \tau} |\mathcal{T}_{\tau'}(R^{N\times\kappa_s})|,$$

which is maximized when $\kappa_s = \min\{N, n+\tau_s\}$. Hence,

$$H(W_1|\operatorname{shape} Y) \le \log_q \sum_{\tau' \preceq \tau} |\mathcal{T}_{\tau'}(R^{N\times\min\{n+\tau_s,N\}})|.$$

So, $H(W|X,Y) \le \log_q \sum_{\tau'\preceq\tau} |\mathcal{T}_{\tau'}(R^{N\times\min\{n+\tau_s,N\}})|$, which completes the proof. □

Corollary 5.3. *The capacity C_{MAMC} is upper-bounded by*

$$C_{MAMC} \le \sum_{i=1}^{s}(\mu_i-\xi_i)\xi_i+\sum_{i=1}^{s}(\min\{n+\tau_s,N\}-\mu_i)\tau_i+2s\log_q 4+\log_q\tbinom{N+s}{s}+\log_q\tbinom{\tau_s+s}{s}-\log_q\prod_{i=0}^{\tau_s-1}(1-q^{i-N}),$$

where $\xi_i = \min\{N, n+\tau_i, \lfloor \mu_i/2 \rfloor\}$ for all i. In particular, when $\mu \succeq 2N$ and $\tau = t$, the upper bound reduces to

$$C_{MAMC} \le \sum_{i=1}^{s}(\min\{n+t,N\}-t)(\mu_i-\min\{n+t,N\})+2s\log_q 4+\log_q\tbinom{N+s}{s}+\log_q\tbinom{t+s}{s}-\log_q\prod_{i=0}^{t-1}(1-q^{i-N}).$$

Proof. By (5.12), we have

$$\log_q \sum_{\lambda\preceq N,n+\tau,\mu} \begin{bmatrix}\mu\\ \lambda\end{bmatrix}_q \le \sum_{i=1}^{s}(\mu_i-\xi_i)\xi_i + s\log_q 4 + \log_q\binom{N+s}{s}.$$

By (5.9), we have

$$-\log_q|\mathcal{T}_\tau(R^{N\times\mu})| \le -\sum_{i=1}^{s}(N+\mu_i-\tau_i)\tau_i - \log_q\prod_{i=0}^{\tau_s-1}(1-q^{i-N}).$$

Note that

$$|\mathcal{T}_{\tau'}(R^{N\times\min\{n+\tau_s,N\}}| \le |R^{N\times\tau'}| \begin{bmatrix}\min\{n+\tau_s,N\}\\ \tau'\end{bmatrix}_q \le 4^s q^{\sum_{i=1}^{s}(N+\min\{n+\tau_s,N\}-\tau_i')\tau_i'},$$

where the first inequality comes from (5.8), and the second inequality comes from (5.4) and (5.6). Hence,

$$\begin{aligned}\sum_{\tau' \preceq \tau} |\mathcal{T}_{\tau'}(R^{N\times \min\{n+\tau_s,N\}}| &\leq \sum_{\tau' \preceq \tau} 4^s q^{\sum_{i=1}^s (N+\min\{n+\tau_s,N\}-\tau_i')\tau_i'} \\ &\leq \binom{\tau_s+s}{s} 4^s q^{\sum_{i=1}^s (N+\min\{n+\tau_s,N\}-\tau_i)\tau_i}\end{aligned}$$

where the second inequality comes from the fact that τ maximizes the quantity $q^{\sum_{i=1}^s (N+\min\{n+\tau_s,N\}-\tau_i')\tau_i'}$ and the fact that the number of shapes τ' with $\tau' \preceq \tau$ is upper-bounded by $\binom{\tau_s+s}{s}$. Therefore, we have

$$\log_q \sum_{\tau' \preceq \tau} |\mathcal{T}_{\tau'}(R^{N\times \min\{n+\tau_s,N\}})| \leq \sum_{i=1}^s (N+\min\{n+\tau_s,N\}-\tau_i)\tau_i + s\log_q 4 + \log_q \tbinom{\tau_s+s}{s}.$$

Combining all the above results, we have obtained the upper bound. In particular, when $\mu \succeq 2N$ and $\tau = t$, we have $\xi_i = \min\{n+t, N\}$ for all i. Substituting this into the upper bound completes the proof. □

We next study the asymptotic behavior of C_{AMMC}.

Theorem 5.10. *When $\mu \succeq 2N$ and $\tau = t$, the asymptotic capacity $\bar{C}_{MAMC}$ is upper-bounded by*

$$\bar{C}_{MAMC} \leq \begin{cases} \frac{\sum_{i=1}^s \bar{n}(\bar{\mu}_i - \bar{n} - \bar{t})}{\bar{n}|\bar{\mu}|} & \text{if } n+t \leq N \\ \frac{\sum_{i=1}^s (\bar{N}-\bar{t})(\bar{\mu}_i - \bar{N})}{\bar{n}|\bar{\mu}|} & \text{if } n+t > N. \end{cases} \tag{5.16}$$

Proof. This follows directly from Corollary 5.3. □

5.8.2 A Coding Scheme

We again focus on the special case when $\mu \succeq 2N$ and $\tau = t$. We describe a coding scheme that achieves the asymptotic bound in Theorem 5.10.

Encoding

The encoding is a combination of the encoding strategies for the MMC and the AMC. We first consider the case when $n+t > N$. Set $v \geq t$. We construct the input matrix X as

$$X = \begin{bmatrix} 0 & 0 \\ 0 & \bar{X} \end{bmatrix},$$

where the size of $\bar{X}$ is $(N-v)\times(m-v)$, and the sizes of other zero matrices are readily available. Here, $\bar{X}$ is chosen from the set of principal row canonical forms for $\mathcal{T}_\kappa(R^{(N-v)\times(\mu-v)})$ by using the construction in Section 5.4.2, where $\kappa_i = \min\{N-v, \lfloor(\mu_i - v)/2\rfloor\}$ for all i. The encoding is illustrated in Fig. 5.6. Clearly, the encoding rate of the scheme is $R_{\text{MAMC}} = \sum_{i=1}^{s} \kappa_i(\mu_i - v - \kappa_i)$. In particular, when $\mu \succeq 2N$, we have $\lfloor(\mu_i - v)/2\rfloor \geq n - v$ for all i. Thus, $\kappa_i = N - v$ for all i, and the encoding rate is $R_{\text{MAMC}} = \sum_{i=1}^{s}(N-v)(\mu_i - N)$.

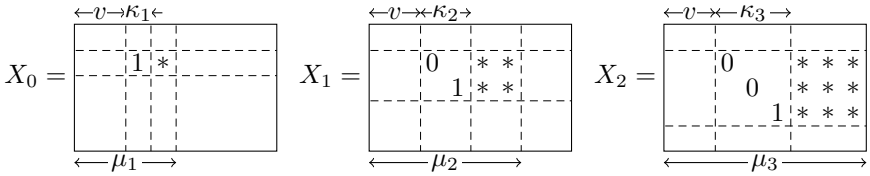


Figure 5.6: Illustration of the MAMC encoding scheme for $s = 3$, $N = 6$, $n = 5$, $v = 2$, $\mu = (4, 6, 8)$, so that $\kappa = (1, 2, 3)$.

We then consider the case when $n + t \leq N$. Similarly, set $v \geq t$. We construct the input matrix X as

$$X = \begin{bmatrix} 0 & \bar{X} \end{bmatrix},$$

where the size of $\bar{X}$ is $n \times (m-v)$. Again, $\bar{X}$ is chosen from the set of principal row canonical forms for $\mathcal{T}_\kappa(R^{n\times(m-v)})$, where $\kappa_i = \min\{n, \lfloor(\mu_i - v)\rfloor\}$ for all i. Clearly, the encoding rate is $R_{\text{MAMC}} = \sum_{i=1}^{s} \kappa_i(\mu_i - v - \kappa_i)$. In particular, when $\mu \succeq 2N$, we have $\kappa_i = n$ for all i, and the encoding rate $R_{\text{MAMC}} = \sum_{i=1}^{s} n(\mu_i - n - v)$.

Decoding

The decoder receives $Y = P\left(\begin{bmatrix} 0 \\ X \end{bmatrix} + W\right)$ and attempts to recover $\bar{X}$ from the row canonical form of Y. We decompose the noise matrix W as

$$W = BE = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{bmatrix} E_1 & E_2 \end{bmatrix},$$

as we did in Section 5.7. Clearly, we have

$$\begin{bmatrix} 0 \\ X \end{bmatrix} + W = \begin{bmatrix} B_1E_1 & B_1E_2 \\ B_2E_1 & \bar{X} + B_2E_2 \end{bmatrix}.$$

Following [78], we define error trapping to be successful if shape $B_1 E_1 = t$. Assume that this is the case. From Section 5.7, there exists some matrix $T \in \mathrm{GL}_N(R)$ such that

$$T\left(\begin{bmatrix} 0 \\ X \end{bmatrix} + W\right) = \begin{bmatrix} B_1 E_1 & B_1 E_2 \\ 0 & \bar{X} \end{bmatrix} = \begin{bmatrix} B_1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} E_1 & E_2 \\ 0 & \bar{X} \end{bmatrix}.$$

Note that

$$\mathrm{RCF}\left(\begin{bmatrix} E_1 & E_2 \\ 0 & \bar{X} \end{bmatrix}\right) = \begin{bmatrix} \tilde{Z}_1 & \tilde{Z}_2 \\ 0 & \bar{X} \end{bmatrix}$$

for some $\tilde{Z}_1 \in R^{t \times v}$ in row canonical form and some $\tilde{Z}_2 \in R^{t \times (m-v)}$. It follows that

$$\begin{aligned} \mathrm{RCF}\left(\begin{bmatrix} 0 \\ X \end{bmatrix} + W\right) &= \mathrm{RCF}\left(\begin{bmatrix} B_1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} E_1 & E_2 \\ 0 & \bar{X} \end{bmatrix}\right) \\ &= \begin{bmatrix} \tilde{Z}_1 & \tilde{Z}_2 \\ 0 & \bar{X} \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

Since P is invertible, we have $\mathrm{RCF}(Y) = \mathrm{RCF}\left(\begin{bmatrix} 0 \\ X \end{bmatrix} + W\right)$, from which $\bar{X}$ can be readily obtained. Hence, decoding amounts to computing the row canonical form, whose complexity is $\mathcal{O}(nm \min\{n, m\})$ basic operations over R.

The decoding can be summarized as follows. First, the decoder computes $\mathrm{RCF}(Y)$. Second, the decoder checks the condition shape $B_1 E_1 = t$. If the condition does not hold, the decoder declares a failure. Otherwise, the decoder outputs $\bar{X}$ from $\mathrm{RCF}(Y)$.

Let $n' = \min\{n+v, N\}$. Let $\hat{Y}$ denote the left-most n' columns of $\mathrm{RCF}(Y)$, i.e., $\hat{Y} = \mathrm{RCF}(Y)[1{:}N, 1{:}n']$. We note that shape $B_1 E_1 = t$ if and only if shape $\hat{Y} = t + \kappa$. Hence, the error probability of the scheme is zero, and the failure probability P_f of the scheme is bounded by $P_f < \frac{2t}{q^{1+v-t}}$ (as shown in Section 5.7).

Finally, if we set v such that $v - t \to \infty$ and $\frac{v-t}{m} \to 0$, as $m \to \infty$, we have $P_f \to 0$, and $\bar{R}_{\mathrm{MAMC}} = \frac{R_{\mathrm{MAMC}}}{n|\mu|}$ approaches the upper bound of the asymptotic capacity in Theorem 5.10.

Theorem 5.11. *When $\tau = t$ and $\mu \succeq 2N$, the coding scheme described above can achieve the upper bound (5.16).*

5.9 Extensions

Previously, we assume that the transfer matrix $A \in R^{N\times n}$ is uniform over all full-rank matrices, and the noise matrix $Z \in R^{N\times m}$ is uniform over all rank-t matrices. In this section, we discuss possible extensions of our previous channel models.

5.9.1 Non-Uniform Transfer Matrices

We note that the uniformness assumption on A leads to a "worst-case" scenario. To see this, let us consider a model identical to the MAMC except for the fact that the transfer matrix A is chosen according to an arbitrary probability distribution on all full-rank matrices in $R^{N\times n}$. It should be clear that the capacity of this channel cannot be smaller than that of the MAMC. This is because our coding scheme does not rely on any particular distribution of A (as long as A is full-column-rank and Z is uniform over all rank-t matrices), and therefore still works for non-uniform distributions. Hence, we have the following lower bound on the asymptotic capacity $\bar{C}$:

$$\bar{C} \geq \begin{cases} \frac{\sum_{i=1}^{s} \bar{n}(\bar{\mu}_i - \bar{n} - \bar{t})}{\bar{n}|\bar{\mu}|} & \text{if } n + t \leq N \\ \frac{\sum_{i=1}^{s} (\bar{N} - \bar{t})(\bar{\mu}_i - \bar{N})}{\bar{n}|\bar{\mu}|} & \text{if } n + t > N. \end{cases} \tag{5.17}$$

On the other hand, the capacity of the channel $Y = AX + Z$ can be upper-bounded by assuming that the transfer matrix A is known at the receiver. One can show that the asymptotic capacity is upper-bounded by

$$\bar{C} \leq \begin{cases} \frac{\sum_{i=1}^{s} \bar{n}(\bar{\mu}_i - \bar{t})}{\bar{n}|\bar{\mu}|} & \text{if } n + t \leq N \\ \frac{\sum_{i=1}^{s} (\bar{N} - \bar{t})(\bar{\mu}_i - \bar{t})}{\bar{n}|\bar{\mu}|} & \text{if } n + t > N. \end{cases} \tag{5.18}$$

Note that when μ_1 is much larger than N, the difference between the lower bound (5.17) and the upper bound (5.18) is small. In this case, our coding scheme is close to the capacity.

5.9.2 Noise Matrix with Variable Rank

We consider a more general case where the number of error packets is allowed to vary, while still bounded by t. More precisely, we assume that Z is chosen uniform at random from rank-T matrices, where $T \in \{0, \ldots, t\}$ is a random variable with an arbitrary probability distribution $\Pr[T = k] = p_k$.

Note that

$$\begin{aligned} H(Z) = H(Z,T) &= H(T) + H(Z|T) \\ &= H(T) + \sum_k p_k H(Z|T=k) \\ &= H(T) + \sum_k p_k \log_q |\mathcal{T}_k(R^{N\times\mu})| \\ &\leq H(T) + \log_q |\mathcal{T}_t(R^{N\times\mu})|. \end{aligned}$$

Hence, the capacity may be reduced by at most $H(T) \leq \log_q(t+1)$ compared to the MAMC. This loss is asymptotically negligible for large n and N.

The coding scheme remains the same. The only difference is that now decoding errors may occur, because the condition shape $B_1E_1 = t$ becomes shape $B_1E_1 = T$, which is, in general, impossible to check. Yet, the analysis of decoding is still applicable, and the error probability is bounded by $P_e < \frac{2t}{q^{1+v-t}}$, which goes to 0 as $v - t \to \infty$.

5.9.3 Non-uniform Noise Matrices

We note that the uniformness assumption on Z again gives a "worst-case" scenario. To see this, consider a model identical to the MAMC except for the fact that the noise matrix Z is chosen according to some non-uniform probability distribution on $\mathcal{T}_t(R^{N\times m})$. It should be clear that the capacity can only increase, since the entropy $H(Z)$ always decreases.

To apply our coding scheme in this more general case, we need some transformation. At the transmitter side, let $X = X'Q$, where $Q \in R^{m\times m}$ is chosen uniformly at random (and independent of any other variables) from the set of matrices of the form

$$Q = \begin{bmatrix} Q'_{\mu_1\times\mu_1} & 0 \\ 0 & I_{m-\mu_1} \end{bmatrix}.$$

Here, Q' is an invertible matrix (of size $\mu_1 \times \mu_1$) and I is an identify matrix (of size $(m-\mu_1)\times(m-\mu_1)$). Clearly, Q is invertible by construction. At the receiver side, let $Y' = PYQ^{-1}$, where $P \in R^{N\times N}$ is chosen uniformly at random (and independent of any other variables) from all invertible matrices. Then

$$\begin{aligned} Y' = PYQ^{-1} &= P(AX'Q + Z)Q^{-1} \\ &= (PA)X' + PZQ^{-1}. \end{aligned}$$

After this transformation, our coding scheme can be applied directly. Moreover, our error analysis still holds, and the failure probability is again bounded by $P_f < \frac{2t}{q^{1+v-t}}$.

5.10 Summary

In this chapter, we have studied the matrix channel $Y = AX + BE$ where the packets are from the ambient space Ω of form (5.2). Under the assumption that A is uniform over all full-rank matrices and BE is uniform over all rank-t matrices, we have derived tight capacity results and provided polynomial-complexity capacity-achieving coding schemes, which naturally extend the work of [78] from finite fields to certain finite rings. Our extension is based on several new enumeration results and construction methods, for matrices over finite chain rings, which may be of independent interest.

We believe that there is still much work to be done in this area. One direction would be to further relax the assumptions on A and BE. Following this direction, we have explored a particular case when A can be any matrix and $BE = 0$ in [73]. Another direction would be to find other applications of the algebraic tools developed in this chapter, especially the row canonical form.

Chapter 6

Application to Random-Access Wireless Networks

This chapter studies the effect of C&F on stability and delay of slotted-ALOHA-based random-access systems. It turns out that this problem is closely related to the stability condition of slotted ALOHA with multi-packet reception, which is, however, largely open except for two special cases: the two-user case and the fully-symmetric case. In this chapter, an *approximate* stability condition is proposed, which not only recovers existing results, but also is provably *exact* when the number of users grows large. Further, it is shown that the approximate stability condition is very accurate even for systems with a small number of users (such as two or three). Finally, this stability condition is used to characterize the benefit of C&F in terms of throughput and delay.

6.1 Introduction

Previous chapters incorporate four important practical constraints into the theory of C&F, providing a theoretical foundation for its application to wireless networks. Still, the models we discussed do not capture some other key aspects of real-world wireless networks, including bursty data traffic and decentralized network operations. In addition, in many networking applications, delay is a primary concern, which has not been analyzed in previous models.

As a starting point to capture these aspects, we consider slotted-ALOHA-based random-access protocols. Such protocols have been used in various systems, ranging from satellite communications networks to wireless local area networks. In particular, we are interested in the benefit of C&F on such systems

in terms of throughput and delay.

It turns out that the characterization of the benefit of C&F is closely related to the stability condition of slotted ALOHA with multi-packet reception. However, such stability condition is largely open except for two special cases: the two-user case and the fully-symmetric case [90]. In order to make progress on this open problem, we propose an approximate stability condition and show that it is *asymptotically exact* when the number of users grows large. Furthermore, this approximate stability condition recovers all existing results, and is extremely accurate even for small systems with three users (the approximation is exact for two users).

We then apply the stability condition to derive the throughput and delay performance of slotted ALOHA with C&F, and compare it with standard ALOHA systems. We show that C&F significantly improves the throughput and delay performance.

6.2 System Model

Consider a system where N users contend to transmit packets to a central processor through K relays. Such a system model represents enterprise WiFi where multiple access points (APs) are connected to a central controller. Assume that time is slotted so that packet transmissions begin only at the start of a slot. Packets are of constant length whose transmission takes one time slot.

Each user is equipped with an infinite buffer for storing packets in a FIFO manner. Packets arrive into user i's buffer according to a Bernoulli process $\{A_i(t)|t = 0, 1, \ldots\}$ with mean λ_i, i.e., at each time slot, a new packet arrives into the buffer of user i with probability λ_i. The arrival processes are assumed to be independent across users.

At the beginning of each slot, if user i's buffer is nonempty, it transmits a packet with probability p_i. Let $\mathbf{p} = (p_1, \ldots, p_N)$ denote the vector of fixed transmission probabilities. At the end of each slot, each relay decodes as many linear combinations as possible, and then sends the corresponding coefficients to the central processor, based on which the central processor is able to decide whether it can recover all the transmitted packets. If yes, the central processor will request certain (linear independent) linear combinations from the relays. Otherwise, the central processor declares a failure.

Suppose that L ($L \leq N$) users are sending packets in a same slot. If the central processor is able to recover all the L transmitted packets, it acknowledges this round of communication (via the help of relays). Once an ACK arrives at a user at the end of a slot, the user removes the corresponding packet from its buffer.

Denote by q_L the conditional probability of recovering these L packets at the central possessor. Here,

the conditional recovery probabilities $\{q_L\}$, which abstract the C&F operation at the physical layer, are assumed to be constant across different slots. Denote by $X_i(t)$ the number of packets in the buffer of user i at the beginning of slot t. The state of the system at slot t is given by $\mathbf{X}(t) = (X_1(t), \ldots, X_N(t))$. Clearly, $\{\mathbf{X}(t)|t = 0, 1, \ldots\}$ is a discrete-time Markov chain. If the Markov chain $\{\mathbf{X}(t)\}$ has a stationary distribution for packet arrival rates $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)$, we say that the system is *stable* for $\boldsymbol{\lambda}$. The stability region Λ^N is defined as the set of vectors $\boldsymbol{\lambda}$ such that the system is stable for $\boldsymbol{\lambda}$.

Although the Markov chain $\mathbf{X}(t)$ is easy to describe, it is, in general, very difficult to analyze due to the complex interactions among the queues. Indeed, only two special cases are well understood in the literature in terms of the stability region. They are the two-user case (i.e., $N = 2$) and the symmetric case (i.e., $\lambda_1 = \cdots = \lambda_N$ and $p_1 = \cdots = p_N$). However, users in a practical system typically have different arrival rates and transmission probabilities.

6.3 Main Result

We provide an approximate expression of the stability region for a system containing an arbitrary number of users with different transmission probabilities. We prove that this approximation is *exact* when the number of users grows large.

For ease of presentation, we impose a restriction on the conditional probabilities $\{q_L\}$: $q_L = 0$ for $L > 2$. This restriction says that the relays give up the decoding opportunity when there are more than two active users. This restriction is made to keep the expression of the stability region simple. In Sec. 6.6, we will explain how to relax this restriction.

Let $\partial_j[0, 1]^N$ be the set of $\rho \in \mathbb{R}_+^N$ such that $\forall i$, $\rho_i \leq 1$, and $\rho_j = 1$. Under the above condition, the approximate stability region can be expressed as follows.

Definition 6.1. *The approximate stability region $\hat{\Lambda}^N$ is the region lying below one of N boundaries $\partial_j\hat{\Lambda}^N$ defined by*

$$\partial_j\hat{\Lambda}^N = \left\{\lambda \mid \exists\rho \in \partial_j[0, 1]^N : \forall i, \lambda_i = P_i(\rho)\right\},$$

where

$$P_i(\rho) = \rho_i p_i \prod_{j\neq i}(1 - \rho_j p_j)\left(q_1 + q_2 \sum_{j\neq i} \frac{\rho_j p_j}{1 - \rho_j p_j}\right).$$

First, we notice that the above approximate stability region is *exact* for the two-user case and the symmetric case. As such, it recovers the existing results in the literature. For the two-user case, the

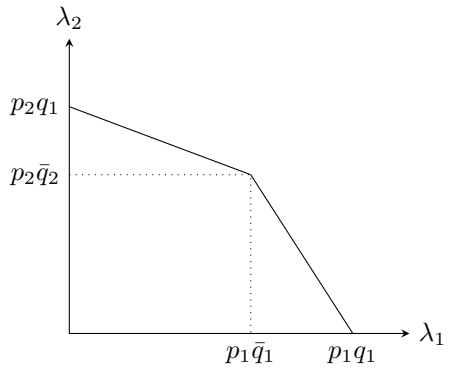


Figure 6.1: Stability region for a fixed transmission probability vector (p_1, p_2).

boundaries $\partial_j \hat{\Lambda}^2$ are line segments given by

$$\partial_1 \hat{\Lambda}^2 = \{(q_1 p_1, 0) + \rho p_2((q_2 - q_1)p_1, Q_2) \mid \rho \in [0,1]\},$$
$$\partial_2 \hat{\Lambda}^2 = \{(0, q_1 p_1) + \rho p_1(Q_1, (q_2 - q_1)p_2) \mid \rho \in [0,1]\},$$

where $\bar{q}_1 \triangleq q_1(1-p_2) + q_2 p_2$ and $\bar{q}_2 \triangleq q_1(1-p_1) + q_2 p_1$. Hence, the approximate stability region $\hat{\Lambda}^2$ is a simple polygon with four vertices $(0,0)$, $(p_1 q_1, 0)$, $(p_1\bar{q}_1, p_2\bar{q}_2)$, and $(0, p_2 q_1)$, as illustrated in Fig. 6.1. This agrees with the stability region derived in [90]. For the symmetric case, the intersection of the N boundaries is the unique point $(\lambda^*, \ldots, \lambda^*)$, where

$$\lambda^* = \frac{1}{N} \sum_{k=1}^{2} k q_k \binom{N}{k} p^k (1-p)^{N-k}.$$

This matches the maximum stable throughput derived in [90].

Our main result states that the approximate stability region $\hat{\Lambda}^N$ is very close to the actual stability region Λ^N when N is large. Define $1^N \triangleq (1/N, \ldots, 1/N)$. Further, assume that

$$\sum_{i=1}^{N} p_i < \frac{\sqrt{q_1^2 + 4q_2^2} - q_1 + 2q_2}{2q_2}.$$

This assumption simplifies the expression of the stability region, and can be relaxed in the analysis.

Theorem 6.1. *For any $\epsilon > 0$ small enough, there exists $N(\epsilon)$ such that for all $N > N(\epsilon)$:*

1. *if $\lambda^N + \epsilon \cdot 1^N \in \hat{\Lambda}^N$, then $\lambda^N \in \Lambda^N$;*
2. *if $\lambda^N - \epsilon \cdot 1^N \notin \hat{\Lambda}^N$, then $\lambda^N \notin \Lambda^N$.*

Theorem 6.1 provides new bounds and asymptotic results for the stability regions. Theorem 6.1 is

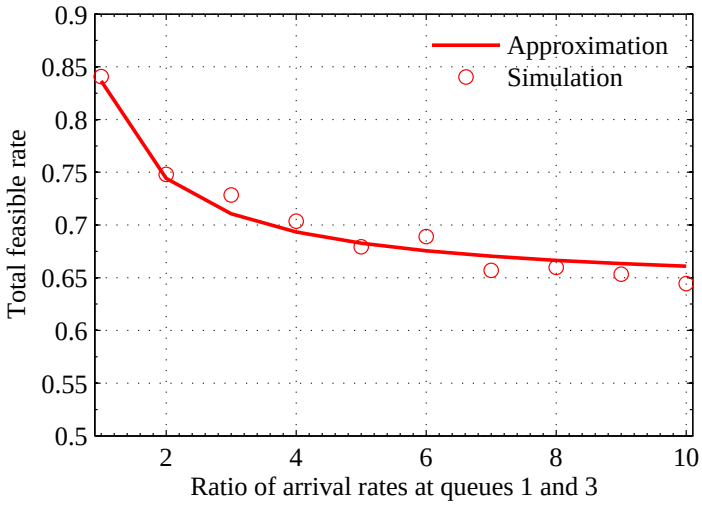


Figure 6.2: Stability condition for a system of three users with same transmission probability.

proven Appendix K. The main steps of the proof are as follows. First, we study the stability of the limiting system (where N, the number of users, tends to infinity) in which the evolutions of the queues of the various users are independent. Then, we relate the stability of the limiting system to the stability of a finite system when N grows large.

6.4 Accuracy of $\hat{\Lambda}^N$

Theorem 6.1 says that the gap between the approximate region $\hat{\Lambda}^N$ and the actual region Λ^N tends to 0 when N grows large. But we notice that the approximate region $\hat{\Lambda}^N$ is very accurate even for small N, as illustrated in the numerical experiments provided below.

Our simulation set-up is described as follows. The system consists of N users and 2 relays. The channel between each active user and each relay is a Rayleigh fading channel, independent of each other. The average received signal-to-noise ratio (SNR) is set to 15 dB. The code rate is set to 2 bits per channel use, corresponding to a scheme using 16-QAM together with a rate 1/2 channel code (e.g., a convolutional code or an LDPC code). Under the above setup, it is easy to obtain the conditional probabilities $\{q_L\}$ through simulations. They are $q_1 = 99.15\%$ and $q_2 = 89.07\%$.

Example 1: We consider the case of $N = 3$ users, each transmitting with probability 1/3. We vary the relative values of the arrival rates at the users: $\lambda_1 = \lambda$, $\lambda_2 = \lambda$, $\lambda_3 = \lambda/x$. We vary x from 1 to 10.

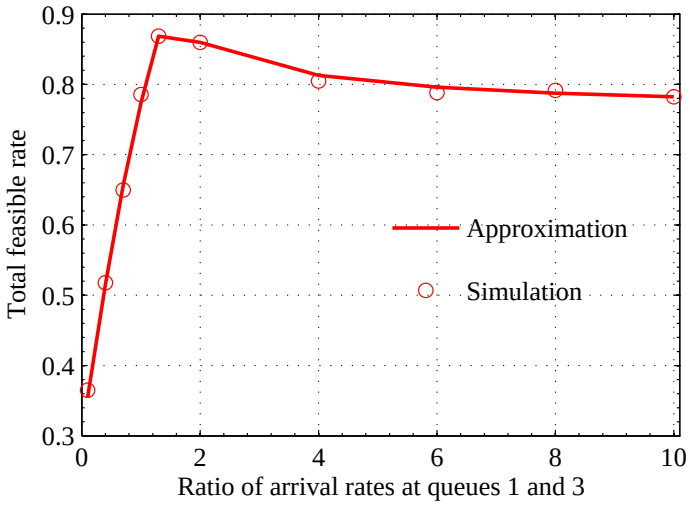


Figure 6.3: Stability condition for a system of three users with different transmission probabilities.

It can be shown that the approximate stability condition is

$$\sum_{i=1}^{3} \lambda_i < \frac{4}{9}q_1 + \frac{2}{9}q_2 + \frac{2}{9}q_2\rho_3,$$

where

$$\rho_3 = \frac{3(q_1 + q_2/2)}{2x(q_1 + q_2) + (q_1 - q_2/2)}.$$

Fig. 6.2 compares this limit to the actual stability limit found by simulation. In order to obtain steady-state performance, we first run the simulation for $100,000$ slots, and then collect data for another $10,000$ slots. As observed from Fig. 6.2, our analytical results match the simulation quite well.

Example 2: We consider a similar numerical example where p_1, p_2, p_3 are equal to 0.4, 0.4, 0.3, respectively. The arrival rates are set in the same way as in Example 1. We vary x from 0.1 to 10. Let

$$x_0 \triangleq \frac{p_1(1-p_3)\left(q_1 + q_2\left(\frac{p_1}{1-p_1} + \frac{p_3}{1-p_3}\right)\right)}{(1-p_1)p_3\left(q_1 + q_2\frac{2p_1}{1-p_1}\right)}.$$

For $x < x_0$, at the boundary of $\hat{\Lambda}^3$, user 3 is saturated; whereas for $x \geq x_0$, user 1 and user 2 are saturated. The approximate stability condition is:

$$\sum_{i=1}^{3} \lambda_i < \begin{cases} \rho_1 p_1(1-\rho_1 p_1)(1-p_3)\left(q_1 + q_2\left(\frac{\rho_1 p_1}{1-\rho_1 p_1} + \frac{p_3}{1-p_3}\right)\right)(2+1/x), & \text{if } x < x_0 \\ p_1(1-p_1)(1-\rho_3 p_3)\left(q_1 + q_2\left(\frac{p_1}{1-p_1} + \frac{\rho_3 p_3}{1-\rho_3 p_3}\right)\right)(2+1/x), & \text{otherwise} \end{cases}$$

where ρ_1 is the unique solution of

$$\frac{\frac{\rho_1 p_1}{1-\rho_1 p_1}\left(q_1 + q_2\left(\frac{\rho_1 p_1}{1-\rho_1 p_1} + \frac{p_3}{1-p_3}\right)\right)}{\frac{p_3}{1-p_3}\left(q_1 + q_2\frac{2\rho_1 p_1}{1-\rho_1 p_1}\right)} = x$$

in $(0,1)$, and ρ_3 is the unique solution of

$$\frac{\frac{p_1}{1-p_1}\left(q_1 + q_2\left(\frac{p_1}{1-p_1} + \frac{\rho_3 p_3}{1-\rho_3 p_3}\right)\right)}{\frac{\rho_3 p_3}{1-\rho_3 p_3}\left(q_1 + q_2\frac{2p_1}{1-p_1}\right)} = x$$

in $(0,1)$. Fig. 6.3 illustrates the accuracy of $\hat{\Lambda}^3$. Again as expected, our approximation results closely match the simulation.

6.5 Applications

In this section, we apply the approximate stability region to analyze the throughput and delay performance of slotted-ALOHA systems with C&F, based on which we are able to characterize the benefit of C&F.

6.5.1 Throughput and Delay Performance

We begin with the symmetric case and then proceed to the two-class case where the users in each class have the same transmission probability and arrival rate. Extension to the general case is straightforward.

The symmetric case

Recall that for the symmetric case, the approximate stability region states that

$$\lambda \leq \frac{1}{N}\sum_{k=1}^{2} k q_k \binom{N}{k} p^k (1-p)^{N-k}.$$

Hence, the network throughput τ is given by

$$\tau \approx \begin{cases} N\lambda, & \text{if } \lambda < f_N(p) \\ N f_N(p), & \text{otherwise} \end{cases} \tag{6.1}$$

where $f_N(p) = (1/N)\sum_{k=1}^{2} k q_k \binom{N}{k} p^k (1-p)^{N-k}$.

Next, we validate the accuracy of the above result by comparing it with simulation as illustrated

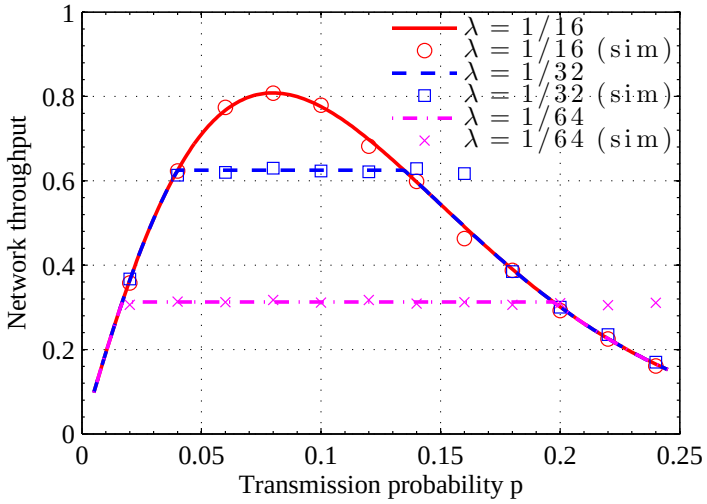


Figure 6.4: Network throughput of ALOHA-C&F versus transmission probability for different arrival rates.

in Fig. 6.4. The set-up is the same as that in Sec. 6.4 except that the number of users is set to 20. We conduct a number of experiments by varying the transmission probability p and the arrival rate λ. As expected, when the traffic load is low, the network throughput is determined by the arrival rate. Otherwise, it is well approximated by $Nf_N(p)$.

Next we turn to the delay performance. There are two kinds of delays of particular interest, namely, the service delay and the total delay. The *service delay* of a packet is defined as the time it takes for this packet to be decoded (at the central processor) after it reaches the head of the queue. The *total delay* of a packet is defined as the time it takes for the packet to be decoded after it arrives the queue. Clearly, the total delay equals to the service delay plus the queuing delay.

As we will see shortly, the delay analysis can be obtained by interpreting the parameter ρ as the "active probability." Suppose that the equation

$$\lambda = \frac{1}{N}\sum_{k=1}^{2} kq_k \binom{N}{k} (\rho p)^k (1-\rho p)^{N-k} \tag{6.2}$$

has a unique solution ρ^* in $(0,1)$[1]. Then ρ^* approximates the probability that a user is active[2]. Hence, for a particular user with a non-empty buffer, the success probability p_s can be approximated as

$$p_s = \frac{1}{N}\sum_{k=1}^{2} kpq_k \binom{N}{k} (\rho^* p)^{k-1} (1-\rho^* p)^{N-k}. \tag{6.3}$$

[1]Otherwise, the system might not be stable.
[2]This approximation becomes exact when queues are independent, as suggested in the proof of Theorem 6.1.

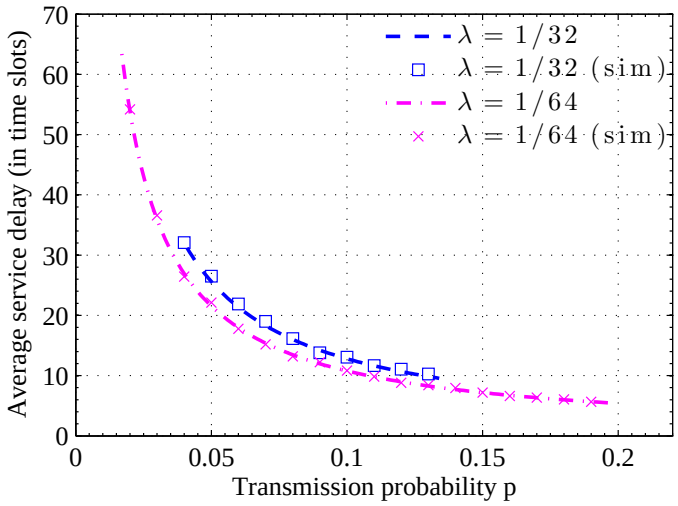


Figure 6.5: Average service delay of ALOHA-C&F versus transmission probability for different arrival rates.

Comparing (6.2) and (6.3), we have $\lambda = \rho^* p_s$.

Note that the service delay depends solely on the success probability p_s under the above approximation. In fact, it can be modeled as a geometric random variable with parameter p_s. Thus, the average service delay D_s can be expressed as

$$D_s \approx 1/p_s = \rho^*/\lambda. \tag{6.4}$$

Similarly, the average total delay D_t can be expressed as

$$D_t \approx \frac{1}{p_s\left(1 - \frac{\lambda(1-p_s)}{(1-\lambda)p_s}\right)}. \tag{6.5}$$

Fig. 6.5 compares our approximation results with simulation with regard to the average service delay. Once again, our approximation results are very accurate. In addition, it is observed that it is beneficial to increase the transmission probability as long as the system is stable.

The two-class case

We proceed to the throughput and delay analysis for the two-class case where the users in class-i have the same transmission probability p_i and the same arrival rate λ_i. Denote by N_i the number of class-i users.

Without loss of generality, assume that $\lambda_2 = \lambda_1/x$ for some $x > 0$. Then for any given x, we can

compute the maximum feasible rates $(\lambda_1^*, \lambda_2^*)$ (with respect to x) as follows. Let

$$x_0 \triangleq \frac{p_1(1-p_2)\left(q_1 + q_2\left((N_1-1)\frac{p_1}{1-p_1} + N_2\frac{p_2}{1-p_2}\right)\right)}{(1-p_1)p_2\left(q_1 + q_2\left(N_1\frac{p_1}{1-p_1} + (N_2-1)\frac{p_2}{1-p_2}\right)\right)}.$$

When $x < x_0$, class-2 users are first saturated as we increase the arrival rates (λ_1, λ_2). In this case, the maximum feasible rates $(\lambda_1^*, \lambda_2^*)$ are given by

$$\lambda_1^* = \frac{\rho_1 p_1}{1-\rho_1 p_1}(1-\rho_1 p_1)^{N_1}(1-p_2)^{N_2}\left(q_1 + q_2\left((N_1-1)\frac{\rho_1 p_1}{1-\rho_1 p_1} + N_2\frac{p_2}{1-p_2}\right)\right)$$
$$\lambda_2^* = \lambda_1^*/x$$

where ρ_1 is the unique solution of

$$\frac{\rho_1 p_1(1-p_2)\left(q_1 + q_2\left((N_1-1)\frac{\rho_1 p_1}{1-\rho_1 p_1} + N_2\frac{p_2}{1-p_2}\right)\right)}{(1-\rho_1 p_1)p_2\left(q_1 + q_2\left(N_1\frac{\rho_1 p_1}{1-\rho_1 p_1} + (N_2-1)\frac{p_2}{1-p_2}\right)\right)} = x$$

in $(0,1)$. Similarly, $x \geq x_0$, then class-1 users are first saturated with $(\lambda_1^*, \lambda_2^*)$ given by

$$\lambda_1^* = \frac{p_1}{1-p_1}(1-p_1)^{N_1}(1-\rho_2 p_2)^{N_2}\left(q_1 + q_2\left((N_1-1)\frac{p_1}{1-p_1} + N_2\frac{\rho_2 p_2}{1-\rho_2 p_2}\right)\right)$$
$$\lambda_2^* = \lambda_1^*/x$$

where ρ_2 is the unique solution of

$$\frac{p_1(1-\rho_2 p_2)\left(q_1 + q_2\left((N_1-1)\frac{p_1}{1-p_1} + N_2\frac{\rho_2 p_2}{1-\rho_2 p_2}\right)\right)}{(1-p_1)\rho_2 p_2\left(q_1 + q_2\left(N_1\frac{p_1}{1-p_1} + (N_2-1)\frac{\rho_2 p_2}{1-\rho_2 p_2}\right)\right)} = x$$

in $(0,1)$.

Fig. 6.6 depicts the region of feasible rates whose boundary corresponds to the collection of maximum feasible rates. We validate its accuracy by comparing the boundary with several simulated maximum feasible rates. As expected, our analytical result matches the simulation very well.

We next turn to the delay performance. As we will soon see, the analysis here is parallel to the delay analysis for the symmetric case. Suppose that the equation

$$\begin{bmatrix}\lambda_1\\ \lambda_2\end{bmatrix} = \begin{bmatrix}\frac{\rho_1 p_1}{1-\rho_1 p_1}(1-\rho_1 p_1)^{N_1}(1-\rho_2 p_2)^{N_2}\left(q_1 + q_2\left((N_1-1)\frac{\rho_1 p_1}{1-\rho_1 p_1} + N_2\frac{\rho_2 p_2}{1-\rho_2 p_2}\right)\right)\\ \frac{\rho_2 p_2}{1-\rho_2 p_2}(1-\rho_1 p_1)^{N_1}(1-\rho_2 p_2)^{N_2}\left(q_1 + q_2\left(N_1\frac{\rho_1 p_1}{1-\rho_1 p_1} + (N_2-1)\frac{\rho_2 p_2}{1-\rho_2 p_2}\right)\right)\end{bmatrix}$$

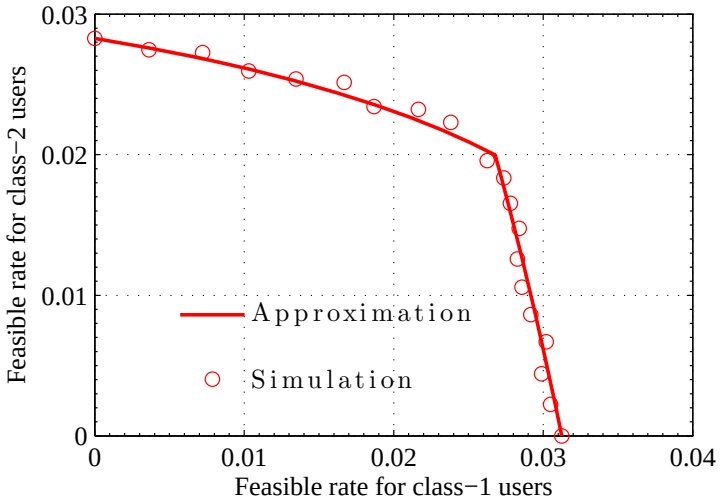


Figure 6.6: Feasible rates for transmission probabilities $p_1 = 0.04$ and $p_2 = 0.03$ with $N_1 = 20$ and $N_2 = 10$.

has a unique solution (ρ_1^*, ρ_2^*) in $(0,1) \times (0,1)$. Then ρ_i^* approximates the active probability for class-i users. Similarly, the success probability $p_s^{(i)}$ for a class-i user can be approximated as $p_s^{(i)} = \lambda_i/\rho_i^*$. The corresponding average service delay $D_s^{(i)}$ and the average total delay $D_t^{(i)}$ can be expressed as

$$D_s^{(i)} \approx \rho_i^*/\lambda_i$$

and

$$D_t^{(i)} \approx \frac{1}{p_s^{(i)}\left(1 - \frac{\lambda_i(1-p_s^{(i)})}{(1-\lambda_i)p_s^{(i)}}\right)},$$

respectively.

6.5.2 Benefit of C&F

We characterize the benefit of C&F by comparing its throughput and delay performance derived as above with the performance of slotted ALOHA systems. Note that our analytical results also apply to slotted ALOHA, if we set $q_L = 0$ for all $L > 1$.

For the symmetric case, we use the same setup as that in Sec. 6.5.1. Fig. 6.7 compares the network throughput for different transmission probabilities. It is observed that C&F significant improves the throughput for a wide range of transmission probabilities. In particular, the maximum network throughput achieved with C&F doubles that without C&F. Fig. 6.8 compares the average service delay for different transmission probabilities when the arrival rate $\lambda = 1/64$. As expected, C&F greatly reduces

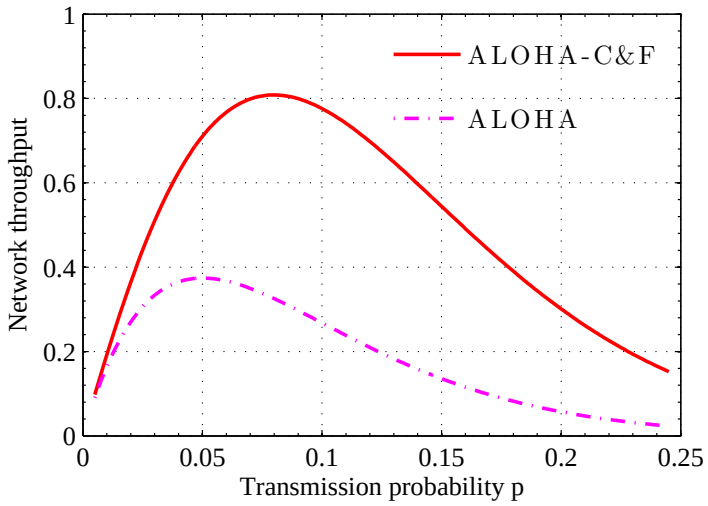


Figure 6.7: Throughput improvement for the symmetric case.

the average service delay.

For the two-class case, we apply the same setup as that in Sec. 6.5.1. Fig. 6.9 compares the region of feasible rates achieved with and without C&F. As seen from Fig. 6.9, C&F significantly enlarges the feasible region.

6.6 Extensions

In this section, we relax the restriction on the conditional probabilities $\{q_L\}$. Instead of assuming that $q_L = 0$ for all $L > 2$, we assume that $q_L = 0$ for all $L > K$ (where $K \geq 2$) and that the polynomial

$$g(x) \triangleq -q_K x^K + (Kq_K - q_{K-1})x^{K-1} + \cdots + (2q_2 - q_1)x + q_1$$

has exactly one positive root. Clearly, this new assumption is much more general. For instance, if we set $K = 2$, then $g(x) = -q_2x^2 + (2q_2 - q_1)x + q_1$, which has precisely one positive root, thereby satisfying our new assumption. In fact, by applying the Descartes' rule of signs, we can obtain a sufficient condition for our new assumption: the number of sign differences between consecutive nonzero coefficients is equal to one. To understand this sufficient condition, let us set $K = 3$, and in this case we have

$$g(x) = -q_3x^3 + (3q_3 - q_2)x^2 + (2q_2 - q_1)x + q_1$$

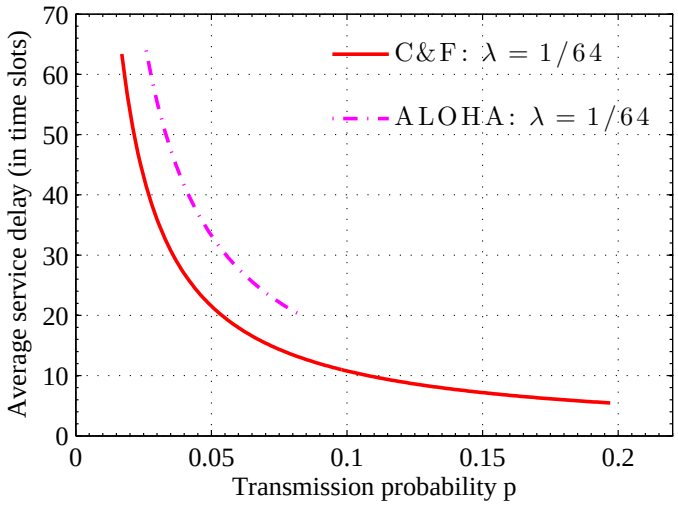


Figure 6.8: Delay improvement for the symmetric case with arrival rate $\lambda = 1/64$.

When $3q_3 > q_2$ and $2q_2 > q_1$, $g(x)$ has one sign change between the first and second coefficients (the sequence of pairs of successive signs is $-+$, $++$, $++$), thereby satisfying the sufficient condition. Similarly, when $3q_3 < q_2$ and $2q_2 < q_1$, or, $3q_3 < q_2$ and $2q_2 > q_1$, $g(x)$ has one sign change between consecutive coefficients.

This new assumption seems to be mild for our set-up. Recall that there are K APs in the system. Loosely speaking, the central processor can recover up to K transmitted messages. This justifies the condition that $q_L = 0$ for all $L > K$. Furthermore, if we set the coding rate appropriately, then the decoding performance will degrade gracefully, as the number of active users increases. This justifies the condition $q_L > q_{L-1}/L$ for $L \leq K$, which ensures that $g(x)$ has exactly one positive root.

Under the new assumption, we can prove that the following approximate stability region is asymptotically exact when the number of users grows.

Let $\mathcal{J}_k(i)$ denote the set of k-tuples $(j_1, \ldots, j_k)$ where $j_1, \ldots, j_k$ are different and none of them is equal to i. Then the approximate stability region $\hat{\Lambda}^N$ is the region lying below one of N boundaries $\partial_j \hat{\Lambda}^N$ defined by

$$\partial_j \hat{\Lambda}^N = \left\{ \lambda \mid \exists \rho \in \partial_j [0,1]^N : \forall i, \lambda_i = P_i(\rho) \right\},$$

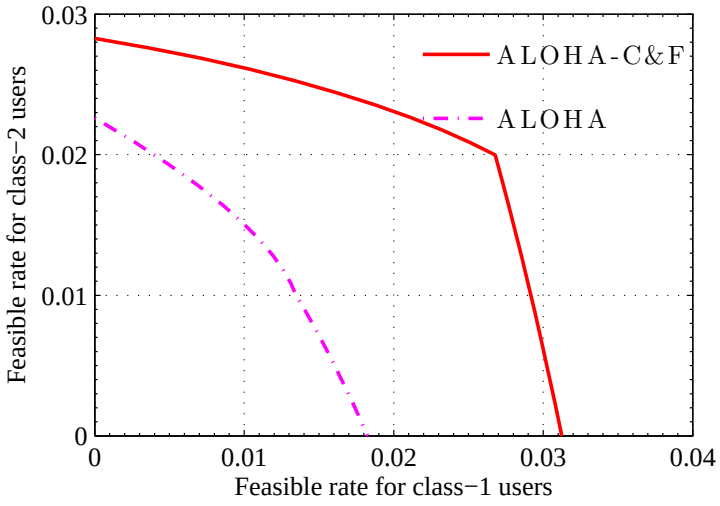


Figure 6.9: Throughput improvement for the two-class case.

where

$$P_i(\rho) = \rho_i p_i \prod_{j\in\mathcal{J}_1(i)} (1-\rho_j p_j) \left(q_1 + q_2 \sum_{j\in\mathcal{J}_1(i)} \frac{\rho_j p_j}{1-\rho_j p_j} + \cdots + q_K \sum_{(j_1,\ldots,j_{K-1})\in\mathcal{J}_{K-1}(i)} \frac{\rho_{j_1} p_{j_1}}{1-\rho_{j_1} p_{j_1}} \times \cdots \times \frac{\rho_{j_{K-1}} p_{j_{K-1}}}{1-\rho_{j_{K-1}} p_{j_{K-1}}} \right).$$

The proof here is parallel to the proof given in Appendix K. The only difference is that we need to prove that the function $f(x) \triangleq (q_1 + q_2 x^2 + \cdots + q_K x^K)e^{-x}$ first increases and then decreases. To show this, note that the derivative of $f(x)$ is given by

$$f'(x) = e^{-x}\left(-q_K x^K + (Kq_K - q_{(K-1)})x^{K-1} + \cdots + (2q_2 - q_1)x + q_1\right) = e^{-x} g(x).$$

Since $g(x)$ has exactly one positive root (as guaranteed by our new assumption), so does $f'(x)$. Hence, $f(x)$ indeed first increases and then decreases.

6.7 Summary

This chapter studies the application of C&F to slotted-ALOHA systems, with a particular focus on its stability and delay performance. The main result of this chapter is an approximate stability region, which is asymptotically exact as the number of users grows large. Furthermore, this stability region is very accurate even for small systems with two or three users. Based on this stability region, we are able

to characterize the benefit of C&F with regard to the throughput and delay, showing that C&F achieves significantly better performance compared to standard slotted-ALOHA systems.

Chapter 7

Conclusion

This thesis develops new theoretical tools for the analysis and design of C&F schemes. These theoretical tools take into account a number of important practical constraints, including short block length, low decoding complexity, imperfect channel state information (CSI), unreliable relays, bursty data traffic, and decentralized network operations. We conclude this thesis by revisiting our contributions and outlining the remaining challenges as well as future work.

7.1 Contributions

In this thesis, we make the following contributions:

- *An Algebraic Framework:* An algebraic framework is developed that allows us to systematically design C&F schemes with controlled block length and decoding complexity. Based on this framework, explicit design criteria, together with concrete design examples, are provided. These results have a number of interesting applications. First, prior to this work, it is a common belief that only a special family of nested lattice codes is compatible with C&F. This work shows that *any* nested lattice code can be used for C&F, provided that the message space is allowed to be a module over certain finite rings. This result greatly enlarges the design space of C&F, allowing for the use of advanced lattice constructions (such as Construction D) in C&F. Following this result, several very recent papers [46, 47, 91, 92] have constructed high-performance C&F schemes with unique advantages. In particular, some of these newly developed schemes achieve better computation rates than the original C&F schemes in certain scenarios. Second, this work demonstrates that C&F schemes can be constructed by using *off-the-shelf* components. In particular, Example 3.7 illustrates how to do this with existing convolutional codes widely used in GSM and 802.11a.

- *Blind Compute-and-Forward:* Prior to this work, it is a common assumption that CSI is perfectly available at each relay. However, as shown by a recent work [27] (and some of our own simulation results in Sec. 4.2.1), C&F is sensitive to channel estimation errors, especially when the number of users is large. This challenges the common assumption and calls for novel solutions. In this work, we propose a new C&F decoding algorithm that enables us to eliminate the need for CSI. It is shown that this new algorithm achieves almost the same performance as its CSI-enabled counterpart with a modest increase in decoding complexity. The key ideas behind our blind C&F scheme include the use of error detection codes and a novel application of the Smoothing Lemma. Based on these two key ideas, we are able to control the complexity of our blind scheme, making it only twice the complexity of the original C&F scheme (with perfect CSI) in the high-throughput region.

- *End-to-End Error Control:* An end-to-end error control mechanism is designed that allows us to effectively tolerate unreliable relays. In particular, this problem is modeled as the multiplicative-additive matrix channel (MAMC). In prior work, such a model was considered for the case when the matrices were over finite fields, and the capacity bounds and capacity-approaching coding schemes were derived. With C&F, the matrices are over finite chain rings, and hence a generalization from the finite-field case to the finite-chain-ring case is needed. To achieve such a generalization, we develop several new linear algebra results for matrices over finite chain rings, such as the matrix canonical form and some new enumeration results, which may be of independent interest. Our generalization sheds some light on the performance of C&F in large wireless relay networks in the presence of unreliable relays. Our main result shows that there exists a polynomial-time coding scheme that approaches the capacity of MAMC as long as the network transfer matrix is full rank. This suggests the network designer to control the network topology so that the network transfer matrix can be made full rank (with high probability), because such a configuration leads to simple, capacity-approaching, coding schemes.

- *Application to Random-Access Wireless Networks:* The application of C&F to random-access wireless networks is investigated, with a particular focus on slotted-ALOHA systems. In particular, an approximate stability region is provided, based on which the benefit of C&F on slotted-ALOHA systems is characterized with regard to the network throughput and delay. It is shown that the use of C&F is able to significantly improve the throughput and delay performance of such systems. This work makes the first step towards incorporating networking aspects into the C&F framework, which is generally missing in prior work. The results in this work provide a theoretical foundation

for understanding the interplay between C&F and higher layer protocols.

7.2 Remaining Challenges and Future Work

The new approaches proposed in this thesis addressed several major challenges involved in bringing C&F theory to practical wireless networks. Below, we discuss a few challenging issues that the thesis does not solve. As we will see, most of these issues are not critical, and some of them can be solved by using our proposed framework in the near future.

Our algebraic framework developed in Chapter 3 leads to a general message space in the form of

$$W = T/\langle \pi_1 \rangle \times T/\langle \pi_2 \rangle \times \cdots \times T/\langle \pi_k \rangle.$$

This opens up the opportunity for new constructions of nested lattice codes for C&F. One such new construction is called product construction, proposed by Huang, Narayanan, and Tunali [92], which has several unique advantages. We expect to see more work along this direction.

Our design criteria developed in Chapter 3 assumes hypercube shaping. Although this assumption holds in many practical wireless systems, it is still worth relaxing this assumption in the hope of achieving better shaping performance. An initial attempt along this direction has been reported in [46].

Also, our design criteria assumes that each relay is interested in linear combinations of the transmitted messages rather than integer combinations of the transmitted codewords. While this is certainly true for C&F, it may not be the case for other closely related problems such as integer-forcing [93] and interference channels [22], in which each relay shall decode integer combinations of the transmitted codewords instead. For these cases, similar design criteria can be easily derived. In particular, under hypercube shaping, one only needs to replace $d(\Lambda/\Lambda')$ with $d(\Lambda)$ and $K(\Lambda/\Lambda')$ with $K(\Lambda)$ in Theorem 3.4.

Our blind C&F scheme proposed in Chapter 4 does not require any CSI, which clearly serves as an extreme case. In reality, some partial channel knowledge can be obtained from the use of pilots. Hence, it would be interesting to combine channel estimation (perhaps in a coarse-resolution) with our searching algorithms, developing some efficient *semi-blind* C&F scheme.

Our end-to-end error control scheme presented in Chapter 5 assumes that the network transfer matrix is full rank. This is a valid assumption if we have certain control over the network topology. However, when we have no such control, it is unclear what is the capacity and how we can approach it with simple coding schemes. Some preliminary results towards this direction can be found in [73].

Our system model in Chapter 6 focuses on the slotted ALOHA case. We believe that it is quite

possible to move beyond the slotted ALOHA case to other cases, such as CSMA/CA and IEEE 802.11. Such an extension may involve the use of semi-Markov processes and Markov renewal processes. On the other hand, there is a lot of recent interest in slotted ALOHA along with interference cancellation (e.g., [94–96]). Although these schemes can achieve substantial increase in throughput, they generally suffer from unbounded delays. On the contrary, our C&F-based random-access schemes improve both the throughput and delay performance. So, perhaps a better strategy is to combine the above two types of random-access schemes.

Finally, this thesis—along with many other work on C&F—assumes a scalar linear Gaussian channel $\mathbf{y} = \sum_\ell h_\ell \mathbf{x}_\ell + \mathbf{z}$. Although such a channel has been widely used in the literature of wireless communications, a more realistic model is still needed to better understand the performance of C&F in real-world wireless channels.

7.3 Looking Forward

In short, this thesis is about bringing C&F closer to wireless practice. More broadly, the thesis advocates designing future wireless networks based on theoretical advances from information theory. Although information theory is very successful in mastering point-to-point communications, it has not yet made a comparable mark in the area of wireless networking—an area that has witnessed a tremendous growth in the past decade and has continued to change people's lives through diverse applications. We believe that, to change this situation, information theorists should show that their proposed schemes have great potential to be implemented in future wireless networks. While this thesis takes the first few steps towards this direction, bridging the gap between theoretical advances and wireless practice remains an exciting challenge in the near future. The success in doing so will help future wireless networks to reach their full potential.

Appendices

Appendix A

Proof of Theorem 3.4

We upper bound the error probability $\Pr[\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\mathbf{n}) \notin \Lambda']$. Consider the (non-lattice) set $\{\Lambda \setminus \Lambda'\} \cup \{\mathbf{0}\}$, i.e., the set difference $\Lambda \setminus \Lambda'$ adjoined with the zero vector. Let $\mathcal{R}_V(\mathbf{0})$ be the Voronoi region of $\mathbf{0}$ in the set $\{\Lambda \setminus \Lambda'\} \cup \{\mathbf{0}\}$, i.e.,

$$\mathcal{R}_V(\mathbf{0}) = \{\mathbf{x} \in \mathbb{C}^n : \forall \boldsymbol{\lambda} \in \Lambda \setminus \Lambda' \left(\|\mathbf{x} - \mathbf{0}\| \leq \|\mathbf{x} - \boldsymbol{\lambda}\|\right)\}.$$

We have the following upper bound for $\Pr[\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\mathbf{n}) \notin \Lambda']$.

Lemma A.1. $\Pr[\mathcal{Q}_{\Lambda}^{NN}(\mathbf{n}) \notin \Lambda'] \leq \Pr[\mathbf{n} \notin \mathcal{R}_V(\mathbf{0})]$.

Proof.

$$\begin{aligned}\Pr[\mathbf{n} \in \mathcal{R}_V(\mathbf{0})] &= \Pr[\forall \boldsymbol{\lambda} \in \Lambda \setminus \Lambda' \left(\|\mathbf{n} - \mathbf{0}\| \leq \|\mathbf{n} - \boldsymbol{\lambda}\|\right)] \\ &= \Pr[\forall \boldsymbol{\lambda} \in \Lambda \setminus \Lambda' \left(\|\mathbf{n} - \mathbf{0}\| < \|\mathbf{n} - \boldsymbol{\lambda}\|\right)].\end{aligned}$$

Note that if $\|\mathbf{n} - \mathbf{0}\| < \|\mathbf{n} - \boldsymbol{\lambda}\|$ for all $\boldsymbol{\lambda} \in \Lambda \setminus \Lambda'$, then $\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\mathbf{n}) \notin \Lambda \setminus \Lambda'$, as $\mathbf{0}$ is closer to $\mathbf{n}$ than any element in $\Lambda \setminus \Lambda'$. Thus,

$$\Pr[\mathbf{n} \in \mathcal{R}_V(\mathbf{0})] \leq \Pr[\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\mathbf{n}) \notin \Lambda \setminus \Lambda'] = \Pr[\mathcal{Q}_{\Lambda}^{\mathrm{NN}}(\mathbf{n}) \in \Lambda'].$$

□

We further upper bound the probability $\Pr[\mathbf{n} \notin \mathcal{R}_V(\mathbf{0})]$. Let $\mathrm{Nbr}(\Lambda \setminus \Lambda') \subseteq \Lambda \setminus \Lambda'$ denote the set of neighbors of $\mathbf{0}$ in $\Lambda \setminus \Lambda'$, i.e., $\mathrm{Nbr}(\Lambda \setminus \Lambda')$ is the smallest subset of $\Lambda \setminus \Lambda'$ such that $\mathcal{R}_V(\mathbf{0})$ is precisely

the set

$$\{\mathbf{x} \in \mathbb{C}^n : \forall \boldsymbol{\lambda} \in \mathrm{Nbr}(\Lambda \setminus \Lambda') \, (\|\mathbf{x} - \mathbf{0}\| \leq \|\mathbf{x} - \boldsymbol{\lambda}\|)\}.$$

Then, for any $\nu > 0$, we have

$$\begin{aligned}
&P[\mathbf{n} \notin \mathcal{R}_V(\mathbf{0})] \\
&= P\left[\|\mathbf{n}\|^2 \geq \|\mathbf{n} - \boldsymbol{\lambda}\|^2, \text{ some } \boldsymbol{\lambda} \in \mathrm{Nbr}(\Lambda \setminus \Lambda')\right] \\
&= P\left[\mathrm{Re}\{\boldsymbol{\lambda}^{\mathsf{H}}\mathbf{n}\} \geq \|\boldsymbol{\lambda}\|^2/2, \text{ some } \boldsymbol{\lambda} \in \mathrm{Nbr}(\Lambda \setminus \Lambda')\right] \\
&\leq \sum_{\boldsymbol{\lambda} \in \mathrm{Nbr}(\Lambda \setminus \Lambda')} P\left[\mathrm{Re}\{\boldsymbol{\lambda}^{\mathsf{H}}\mathbf{n}\} \geq \|\boldsymbol{\lambda}\|^2/2\right] \qquad \text{(A.1)} \\
&\leq \sum_{\boldsymbol{\lambda} \in \mathrm{Nbr}(\Lambda \setminus \Lambda')} \exp(-\nu\|\boldsymbol{\lambda}\|^2/2) E\left[\exp(\nu \mathrm{Re}\{\boldsymbol{\lambda}^{\mathsf{H}}\mathbf{n}\})\right], \qquad \text{(A.2)}
\end{aligned}$$

where (A.1) follows from the union bound and (A.2) follows from the Chernoff bound. Since $\mathbf{n} = \sum_\ell (\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha\mathbf{z}$, we have

$$\begin{aligned}
&E\left[\exp\left(\nu \mathrm{Re}\{\boldsymbol{\lambda}^{\mathsf{H}}\mathbf{n}\}\right)\right] \\
&= E\left[\exp\left(\nu \mathrm{Re}\left\{\boldsymbol{\lambda}^{\mathsf{H}}\left(\sum_\ell (\alpha h_\ell - a_\ell)\mathbf{x}_\ell + \alpha\mathbf{z}\right)\right\}\right)\right] \\
&= E\left[\exp(\nu \mathrm{Re}\{\boldsymbol{\lambda}^{\mathsf{H}}\alpha\mathbf{z}\})\right] \\
&\quad \cdot \prod_\ell E\left[\exp(\nu \mathrm{Re}\{\boldsymbol{\lambda}^{\mathsf{H}}(\alpha h_\ell - a_\ell)\mathbf{x}_\ell\})\right] \qquad \text{(A.3)} \\
&= \exp\left(\frac{1}{4}\nu^2\|\boldsymbol{\lambda}\|^2|\alpha|^2 N_0\right) \\
&\quad \cdot \prod_\ell E\left[\exp(\nu \mathrm{Re}\{\boldsymbol{\lambda}^{\mathsf{H}}(\alpha h_\ell - a_\ell)\mathbf{x}_\ell\})\right] \qquad \text{(A.4)}
\end{aligned}$$

where (A.3) follows from the independence of $\mathbf{x}_1, \ldots, \mathbf{x}_L, \mathbf{z}$ and (A.4) follows from the moment-generating function of a circularly symmetric complex Gaussian random vector.

Lemma A.2. *Let $\mathbf{x} \in \mathbb{C}^n$ be a complex random vector uniformly distributed over a hypercube $\gamma\mathbf{U}\mathcal{H}_n$ for some $\gamma > 0$ and some $n \times n$ unitary matrix. Then*

$$E\left[\exp(Re\{\mathbf{v}^{\mathsf{H}}\mathbf{x}\})\right] \leq \exp(\|\mathbf{v}\|^2\gamma^2/24).$$

Proof. First, we consider a special case where the unitary matrix $\mathbf{U} = \mathbf{I}_n$. In this case, we have

$$\begin{aligned}
&E\left[\exp(\mathrm{Re}\{\mathbf{v}^{\mathsf{H}}\mathbf{x}\})\right] \\
&= E\left[\exp(\mathrm{Re}\{\mathbf{v}\}^T\mathrm{Re}\{\mathbf{x}\} + \mathrm{Im}\{\mathbf{v}\}^T\mathrm{Im}\{\mathbf{x}\})\right] \\
&= E\left[\exp\left(\sum_{i=1}^{n}(\mathrm{Re}\{\mathbf{v}_i\}\mathrm{Re}\{\mathbf{x}_i\} + \mathrm{Im}\{\mathbf{v}_i\}\mathrm{Im}\{\mathbf{x}_i\})\right)\right] \\
&= \prod_{i=1}^{n} E\left[\exp(\mathrm{Re}\{\mathbf{v}_i\}\mathrm{Re}\{\mathbf{x}_i\}\right] E\left[\exp \mathrm{Im}\{\mathbf{v}_i\}\mathrm{Im}\{\mathbf{x}_i\})\right] && \text{(A.5)} \\
&= \prod_{i=1}^{n} \frac{\sinh(\mathrm{Re}\{\mathbf{v}_i\}\gamma/2)}{\mathrm{Re}\{\mathbf{v}_i\}\gamma/2} \frac{\sinh(\mathrm{Im}\{\mathbf{v}_i\}\gamma/2)}{\mathrm{Im}\{\mathbf{v}_i\}\gamma/2} && \text{(A.6)} \\
&\leq \prod_{i=1}^{n} \exp\left(\frac{(\mathrm{Re}\{\mathbf{v}_i\}\gamma)^2}{24}\right) \exp\left(\frac{(\mathrm{Im}\{\mathbf{v}_i\}\gamma)^2}{24}\right) && \text{(A.7)} \\
&= \exp\left(\frac{\gamma^2}{24}\|\mathbf{v}\|^2\right)
\end{aligned}$$

where (A.5) follows from the independence among each real/imaginary component, (A.6) follows from the moment-generating function of a uniform random variable (note that both $\mathrm{Re}\{\mathbf{x}_i\}$ and $\mathrm{Im}\{\mathbf{x}_i\}$ are uniformly distributed over $[-\gamma/2, \gamma/2]$), and (A.7) follows from $\sinh(x)/x \leq \exp(x^2/6)$ (which can be obtained by simple Taylor expansion).

Then we consider a general unitary matrix $\mathbf{U}$. In this case, we have $\mathbf{x} = \mathbf{U}\mathbf{x}'$, where $\mathbf{x}' \in \gamma[-1/2, 1/2]^{2n}$, i.e., both $\mathrm{Re}\{\mathbf{x}'_i\}$ and $\mathrm{Im}\{\mathbf{x}'_i\}$ are uniformly distributed over $[-\gamma/2, \gamma/2]$. Hence,

$$\begin{aligned}
E\left[\exp(\mathrm{Re}\{\mathbf{v}^{\mathsf{H}}\mathbf{x}\})\right] &= E\left[\exp(\mathrm{Re}\{\mathbf{v}^{\mathsf{H}}\mathbf{U}\mathbf{x}'\})\right] \\
&= E\left[\exp(\mathrm{Re}\{(\mathbf{U}^{\mathsf{H}}\mathbf{v})^{\mathsf{H}}\mathbf{x}'\})\right] \\
&\leq \exp\left(\frac{\gamma^2}{24}\|\mathbf{U}^{\mathsf{H}}\mathbf{v}\|^2\right) \\
&= \exp\left(\frac{\gamma^2}{24}\|\mathbf{v}\|^2\right).
\end{aligned}$$

□

Note that $P = \frac{1}{n}E[\|\mathbf{x}_\ell\|^2] = \gamma^2/6$. Thus, we have

$$\begin{aligned}
&E\left[\exp(\nu \mathrm{Re}\{\boldsymbol{\lambda}^{\mathsf{H}}\mathbf{n}\})\right] \\
&\leq \exp\left(\frac{1}{4}\nu^2\|\boldsymbol{\lambda}\|^2|\alpha|^2 N_0\right)\prod_{\ell}\exp(\|\nu\boldsymbol{\lambda}(\alpha h_\ell - a_\ell)\|^2 P/4) \\
&= \exp\left(\frac{1}{4}\nu^2\|\boldsymbol{\lambda}\|^2|\alpha|^2 N_0 + \|\nu\boldsymbol{\lambda}\|^2\|\alpha\mathbf{h}-\mathbf{a}\|^2 P/4\right) \\
&= \exp\left(\frac{1}{4}\|\boldsymbol{\lambda}\|^2\nu^2 N_0 Q(\mathbf{a},\alpha)\right),
\end{aligned}$$

where the quantity $Q(\mathbf{a},\alpha)$ is given by

$$Q(\mathbf{a},\alpha) = |\alpha|^2 + \mathsf{SNR}\,\|\alpha\mathbf{h}-\mathbf{a}\|^2$$

and $\mathsf{SNR} = P/N_0$.

It follows that, for all $\nu > 0$,

$$\begin{aligned}
&\Pr[\mathbf{n} \notin \mathcal{R}_V(\mathbf{0})] \\
&\leq \sum_{\boldsymbol{\lambda}\in \mathrm{Nbr}(\Lambda\setminus\Lambda')} \exp\left(-\nu\|\boldsymbol{\lambda}\|^2/2 + \frac{1}{4}\|\boldsymbol{\lambda}\|^2\nu^2 N_0 Q(\mathbf{a},\alpha)\right).
\end{aligned}$$

Choosing $\nu = 1/(N_0 Q(\mathbf{a},\alpha))$, we have

$$\begin{aligned}
\Pr[\mathbf{n} \notin \mathcal{R}_V(\mathbf{0})] &\leq \sum_{\boldsymbol{\lambda}\in \mathrm{Nbr}(\Lambda\setminus\Lambda')} \exp\left(-\frac{\|\boldsymbol{\lambda}\|^2}{4N_0 Q(\mathbf{a},\alpha)}\right) \\
&\approx K(\Lambda/\Lambda')\exp\left(-\frac{d^2(\Lambda/\Lambda')}{4N_0 Q(\mathbf{a},\alpha)}\right)
\end{aligned}$$

for high signal-to-noise ratios. Therefore, we have

$$\begin{aligned}
\Pr[\mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{n}) \notin \Lambda'] &\leq \Pr[\mathbf{n} \notin \mathcal{R}_V(\mathbf{0})] \\
&\lessapprox K(\Lambda/\Lambda')\exp\left(-\frac{d^2(\Lambda/\Lambda')}{4N_0 Q(\mathbf{a},\alpha)}\right).
\end{aligned}$$

Since α can be carefully chosen, we have

$$\Pr[\mathcal{Q}_\Lambda^{\mathrm{NN}}(\mathbf{n}) \notin \Lambda'] \lessapprox \min_{\alpha\in\mathbb{C}} K(\Lambda/\Lambda')\exp\left(-\frac{d^2(\Lambda/\Lambda')}{4N_0 Q(\mathbf{a},\alpha)}\right),$$

completing the proof for the first part of Theorem 3.4. The second part of Theorem 3.4 follows imme-

diately when the optimal value of α is substituted.

Appendix B

Proof of Proposition 3.2

Recall that $d(\Lambda_r/\Lambda_r')$ is the length of the shortest vectors in the set difference $\Lambda_r \setminus \Lambda_r'$. Hence, we have

$$d(\Lambda_r/\Lambda_r') = \min_{\mathbf{c}\neq\mathbf{0}} \|\sigma^*(\mathbf{c})\|;$$

equivalently, $d^2(\Lambda_r/\Lambda_r') = \min_{\mathbf{c}\neq\mathbf{0}} \|\sigma^*(\mathbf{c})\|^2 = w_E^{\min}(\mathcal{C})$. Recall that $\Lambda = \Lambda_r + i\Lambda_r$. That is, $\Lambda = \Lambda_r \times \Lambda_r$. Hence, we have

$$d^2(\Lambda/\Lambda') = d^2(\Lambda_r/\Lambda_r') = w_E^{\min}(\mathcal{C}).$$

Note that $V(\Lambda') = p^{2n}$ and $V(\Lambda')/V(\Lambda) = p^{2k}$. Hence, we have $V(\Lambda) = p^{2(n-k)}$. Combining the above two results, we have

$$\gamma_c(\Lambda/\Lambda') = w_E^{\min}(\mathcal{C})/p^{2(1-k/n)}.$$

We then turn to $K(\Lambda_r/\Lambda_r')$ and $K(\Lambda/\Lambda')$. When $p = 2$, the minimum Euclidean weight $w_E^{\min}(\mathcal{C})$ of $\mathcal{C}$ is precisely the minimum Hamming weight of $\mathcal{C}$. In this case, $K(\Lambda_r/\Lambda_r') = \left(w_E^{\min}(\mathcal{C})\right) 2^{w_E^{\min}(\mathcal{C})}$, as shown in [36]. When $p > 2$, the set different $\Lambda_r \setminus \Lambda_r'$ can be expressed as

$$\Lambda_r \setminus \Lambda_r' = \bigcup_{\mathbf{c}\neq\mathbf{0}} \{\sigma^*(\mathbf{c}) + \Lambda_r'\}.$$

In this case, $\sigma^*(\mathbf{c})$ is the unique coset leader for the coset $\sigma^*(\mathbf{c}) + \Lambda_r'$. Thus, the number $K(\Lambda_r/\Lambda_r')$ of the shortest vectors in $\Lambda_r \setminus \Lambda_r'$ is precisely the number $A\left(w_E^{\min}(\mathcal{C})\right)$ of coset leaders with $\|\sigma^*(\mathbf{c})\|^2 = w_E^{\min}(\mathcal{C})$. Hence, we have

$$K(\Lambda_r/\Lambda_r') = \begin{cases} A\left(w_E^{\min}(\mathcal{C})\right) 2^{w_E^{\min}(\mathcal{C})}, & \text{when } p = 2, \\ A\left(w_E^{\min}(\mathcal{C})\right), & \text{when } p > 2. \end{cases}$$

Recall that $\Lambda' = \Lambda'_r + i\Lambda'_r$. That is, $\Lambda' = \Lambda'_r \times \Lambda'_r$. It follows that $K(\Lambda/\Lambda') = 2K(\Lambda_r/\Lambda'_r)$, completing the proof.

Appendix C

Proof of Proposition 3.3

The proof is analogous to that of Proposition 3.2 with two differences. First, p is replaced by $|\pi|$ in the expression of $\gamma_c(\Lambda/\Lambda')$. This difference comes from the fact that $V(\Lambda') = |\pi|^{2n}$ and $V(\Lambda')/V(\Lambda) = |\pi|^{2k}$. Second, the case of $|\pi| = 2$ gives an expression of $A\left(w_E^{\min}(\mathcal{C})\right) 4^{w_E^{\min}(\mathcal{C})}$ for $K(\Lambda/\Lambda')$. This is because if the coset $\mathbf{c} + \Lambda'$ contains one shortest vector in $\Lambda \setminus \Lambda'$, then a total of $4^{w_E^{\min}(\mathcal{C})}$ shortest vectors can be found in the coset $\mathbf{c} + \Lambda'$. Suppose that $(c_1, \ldots, c_n)$ is one such shortest vector in $\mathbf{c} + \Lambda'$. Then, $(c_1, \ldots, c_n)$ has precisely $w_E^{\min}(\mathcal{C})$ nonzero elements. Moreover, for each nonzero element, say c_j, if we change it to one of $\{-c_j, i \times c_j, (-i) \times c_j\}$, then the new vector has the same Euclidean norm and is still in the coset $\mathbf{c} + \Lambda'$. Therefore, the number of shortest vectors in $\mathbf{c} + \Lambda'$ is $4^{w_E^{\min}(\mathcal{C})}$.

Appendix D

Λ_r in (3.11) is a Lattice

Let $\tilde{\mathbf{g}}_j = \tilde{\sigma}(\mathbf{g}_j)$, for $j = 1, \ldots, k_s$. It is easy to check that $\boldsymbol{\lambda} \in \Lambda_r$ if and only if $\boldsymbol{\lambda} = p^s\mathbf{r} + \sum_{j=1}^{k_s} c_j\tilde{\mathbf{g}}_j$ for some $\mathbf{r} \in \mathbb{Z}^n$ and $c_j \in \{0, \ldots, p^s - 1\}$ satisfying the division condition: when $k_t < j \leq k_{t+1}$, $p^t \mid c_j$ (where $t = 1, \ldots, s-1$).

Let $\boldsymbol{\lambda}_i = p^s\mathbf{r}_i + \sum_{j=1}^{k_s} c_{ij}\tilde{\mathbf{g}}_j$ $(i = 1, 2)$ be two vectors from Λ_r. Then we have $\mathbf{r}_1, \mathbf{r}_2 \in \mathbb{Z}^n$, and $c_{1j}, c_{2j} \in \{0, \ldots, p^s - 1\}$ satisfy the division condition. Now consider the difference

$$\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 = p^s(\mathbf{r}_1 - \mathbf{r}_2) + \sum_{j=1}^{k_s}(c_{1j} - c_{2j})\tilde{\mathbf{g}}_j.$$

We will show that $\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 \in \Lambda_r$. We need the following lemma from elementary arithmetic.

Lemma D.1. *Let $a, d \in \mathbb{Z}$ with $d \neq 0$. Then there exist unique $q, r \in \mathbb{Z}$ such that $a = qd + r$ and $0 \leq r < |d|$.*

Using the above lemma, we have $c_{1j} - c_{2j} = q_j p^s + r_j$ for some $q_j \in \mathbb{Z}$ and $r_j \in \{0, \ldots, p^s - 1\}$. Furthermore, if p^t divides $c_{1j} - c_{2j}$, then p^t divides r_j, where $t = 1, \ldots, s-1$. Thus, $\{r_j\}$ satisfy the division condition. Note that

$$\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 = p^s(\mathbf{r}_1 - \mathbf{r}_2 + \sum_j q_j\tilde{\mathbf{g}}_j) + \sum_j r_j\tilde{\mathbf{g}}_j.$$

Thus, $\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2 \in \Lambda_r$, which implies that Λ_r is indeed a lattice.

Next, we will construct a generator matrix for Λ_r. Let $\tilde{\mathbf{G}}$ denote the matrix with rows $\tilde{\mathbf{g}}_1, \ldots, \tilde{\mathbf{g}}_n$. Clearly, we have $\det(\tilde{\mathbf{G}}) = 1$ due to the way $\{\mathbf{g}_i\}$ are constructed. This implies that $\tilde{\mathbf{g}}_1, \ldots, \tilde{\mathbf{g}}_n$ span $\mathbb{Z}^n$ over $\mathbb{Z}$. That is, any vector $\mathbf{r} \in \mathbb{Z}^n$ can be expressed as an integer combination of $\tilde{\mathbf{g}}_1, \ldots, \tilde{\mathbf{g}}_n$.

Consider the set of all integer combinations of the following vectors: $\tilde{\mathbf{g}}_1, \ldots, \tilde{\mathbf{g}}_{k_1}$, $p\tilde{\mathbf{g}}_{k_1+1}, \ldots, p\tilde{\mathbf{g}}_{k_2}$, ..., $p^s\tilde{\mathbf{g}}_{k_s+1}, \ldots, p^s\tilde{\mathbf{g}}_n$. On the one hand, it is easy to see that any integer combination of these vectors is a lattice point in Λ_r. On the other hand, let $\boldsymbol{\lambda} = p^s\mathbf{r} + \sum_{j=1}^{k_s} c_j\tilde{\mathbf{g}}_j$ be a lattice point in Λ_r, where $\mathbf{r} \in \mathbb{Z}^n$ and $\{c_j\}$ satisfy the division condition. Recall that $\mathbf{r} = \sum_{j=1}^{n} b_j\tilde{\mathbf{g}}_j$ for some $b_j \in \mathbb{Z}$. Thus, we have

$$\boldsymbol{\lambda} = \sum_{i=1}^{k_s}(c_i + p^s b_i)\tilde{\mathbf{g}}_i + \sum_{j=k_s+1}^{n} p^s b_j\tilde{\mathbf{g}}_j.$$

Since $p^t \mid c_i$, when $k_t < i \leq k_{t+1}$, we have $p^t \mid c_i + p^t b_i$, when $k_t < i \leq k_{t+1}$. Hence, $\boldsymbol{\lambda}$ is indeed an integer combination of the above vectors. Let $\mathbf{G}_{\Lambda_r}$ be the matrix formed by these vectors. Then $\mathbf{G}_{\Lambda_r}$ is a generator matrix for Λ_r.

Appendix E

Proof of Relation (3.12)

The following two observations simplify the proof of the relation (3.12). First, it suffices to consider the case of $s = 2$, since the case of $s > 2$ is essentially the same. Second, it suffices to prove the relation for the pair of nested $\mathbb{Z}$-lattices $\Lambda_r \supseteq \Lambda'_r$, i.e.,

$$\mathbf{G}_{\Lambda'_r} = \mathrm{diag}(\underbrace{p^2, \ldots, p^2}_{k_1}, \underbrace{p, \ldots, p}_{k_2 - k_1}, \underbrace{1, \ldots, 1}_{n - k_2}) \mathbf{G}_{\Lambda_r} \tag{E.1}$$

due to the lifting operation.

Next we will construct two generator matrices $\mathbf{G}_{\Lambda_r}$ and $\mathbf{G}_{\Lambda'_r}$ satisfying the above relation. Let $\tilde{\mathbf{g}}_i$ denote $\tilde{\sigma}(\mathbf{g}_i)$, for $i = 1, \ldots, n$. On the one hand, by Appendix D, there exists a generator matrix $\mathbf{G}_{\Lambda_r}$ of Λ_r consisting of basis vectors $\tilde{\mathbf{g}}_1, \ldots, \tilde{\mathbf{g}}_{k_1}$, $p\tilde{\mathbf{g}}_{k_1+1}, \ldots, p\tilde{\mathbf{g}}_{k_2}$, $p^2\tilde{\mathbf{g}}_{k_2+1}, \ldots, p^2\tilde{\mathbf{g}}_n$. On the other hand, the vectors $\{p^2\tilde{\mathbf{g}}_1, \ldots, p^2\tilde{\mathbf{g}}_n\}$ form a basis of Λ'_r, because $\tilde{\mathbf{g}}_1, \ldots, \tilde{\mathbf{g}}_n$ span $\mathbb{Z}^n$ over $\mathbb{Z}$. By comparing these two bases for Λ_r and Λ'_r, we conclude that there exist two generator matrices $\mathbf{G}_{\Lambda_r}$ and $\mathbf{G}_{\Lambda'_r}$ satisfying Relation (E.1).

Appendix F

Proof of Proposition 3.4

It suffices to consider the case $s = 2$, since the case of $s > 2$ is essentially the same. Consider a lattice point $\boldsymbol{\lambda} \in \Lambda_r \setminus \Lambda_r'$ given by

$$\boldsymbol{\lambda} = p^2\mathbf{r} + \sum_{j=1}^{k_1} \beta_{1j}\tilde{\mathbf{g}}_j + \sum_{j=1}^{k_2} p\beta_{2j}\tilde{\mathbf{g}}_j,$$

where $\beta_{ij} \in \{0, \ldots, p-1\}$. Clearly, some β_{ij} must be nonzero, because otherwise $\boldsymbol{\lambda} = p^2\mathbf{r} \in \Lambda_r'$. We consider the following two cases.

Case 1: some β_{1j} is nonzero. In this case, we construct a new lattice $\Lambda_{r1} = \{p\mathbf{r} + \sum_{j=1}^{k_1} \beta_j\tilde{\mathbf{g}}_j : \mathbf{r} \in \mathbb{Z}^n, \beta_j \in \{0, \ldots, p-1\}\}$ and a new sublattice $\Lambda_{r1}' = \{p\mathbf{r} : \mathbf{r} \in \mathbb{Z}^n\}$. Clearly, we have $\boldsymbol{\lambda} \in \Lambda_{r1}$ and $\boldsymbol{\lambda} \notin \Lambda_{r1}'$. Thus, $\boldsymbol{\lambda} \in \Lambda_{r1} \setminus \Lambda_{r1}'$. Note that the nested lattice pair $\Lambda_{r1} \supseteq \Lambda_{r1}'$ can be obtained from the code $\mathcal{C}_1$ by Construction A. Thus, we have $\|\boldsymbol{\lambda}\|^2 \geq w_E^{\min}(\mathcal{C}_1)$ and the number of lattice points $\boldsymbol{\lambda}$ of the Euclidean weight $w_E^{\min}(\mathcal{C}_1)$ is upper bounded by $K(\Lambda_{r1}/\Lambda_{r1}')$.

Case 2: all β_{1j} are zero, and some β_{2j} is nonzero. In this case, we construct a new lattice $\Lambda_{r2} = \{p\mathbf{r} + \sum_{j=1}^{k_2} \beta_j\tilde{\mathbf{g}}_j : \mathbf{r} \in \mathbb{Z}^n, \beta_j \in \{0, \ldots, p-1\}\}$ and a new sublattice $\Lambda_{r2}' = \{p\mathbf{r} : \mathbf{r} \in \mathbb{Z}^n\}$. Clearly, we have $\boldsymbol{\lambda} = p^2\mathbf{r} + \sum_{j=1}^{k_2} p\beta_{2j}\tilde{\mathbf{g}}_j \in p\Lambda_{r2}$ and $\boldsymbol{\lambda} \notin p\Lambda_{r2}'$. Thus, $\boldsymbol{\lambda} \in p\Lambda_{r2} \setminus p\Lambda_{r2}'$. Similar to Case 1, the nested lattice pair $\Lambda_{r2} \supseteq \Lambda_{r2}'$ can be obtained from the code $\mathcal{C}_2$ by Construction A. Thus, we have $\|\boldsymbol{\lambda}\|^2 \geq p^2 w_E^{\min}(\mathcal{C}_2)$, and the number of lattice points $\boldsymbol{\lambda}$ of the Euclidean weight $w_E^{\min}(\mathcal{C}_2)$ is upper bounded by $K(\Lambda_{r2}/\Lambda_{r2}')$.

Combining the above two cases, we have, for all $\boldsymbol{\lambda} \in \Lambda_r \setminus \Lambda_r'$, that $\|\boldsymbol{\lambda}\|^2 \geq \min\{w_E^{\min}(\mathcal{C}_1), p^2 w_E^{\min}(\mathcal{C}_2)\}$,

which implies that $d^2(\Lambda_r/\Lambda_r') \geq \min\{w_E^{\min}(\mathcal{C}_1), p^2 w_E^{\min}(\mathcal{C}_2)\}$. Recall that $\Lambda = \Lambda_r \times \Lambda_r$. Hence, we have

$$\begin{aligned} d^2(\Lambda/\Lambda') &= d^2(\Lambda_r/\Lambda_r') \\ &\geq \min\{w_E^{\min}(\mathcal{C}_1), p^2 w_E^{\min}(\mathcal{C}_2)\}. \end{aligned}$$

Note that $V(\Lambda') = p^{4n}$ and $V(\Lambda')/V(\Lambda) = p^{2(k_1+k_2)}$, since each $\beta_{ij} \in \{0, \ldots, p-1\}$. Hence, we have $V(\Lambda) = p^{2(2n-k_1-k_2)}$ and

$$\begin{aligned} \gamma_c(\Lambda/\Lambda') &= d^2(\Lambda/\Lambda')/p^{2(2-(k_1+k_2)/n)} \\ &\geq \frac{\min\{w_E^{\min}(\mathcal{C}_1), p^2 w_E^{\min}(\mathcal{C}_2)\}}{p^{2(2-(k_1+k_2)/n)}}. \end{aligned}$$

We also have $K(\Lambda_r/\Lambda_r') \leq K(\Lambda_{r1}/\Lambda_{r1}') + K(\Lambda_{r2}/\Lambda_{r2}')$ and $K(\Lambda/\Lambda') = 2K(\Lambda_r/\Lambda_r')$, completing the proof for the case $s = 2$.

Appendix G

Modified Viterbi Decoder for Example 3.7

We will show that the nearest neighbor quantizer $\mathcal{Q}_\Lambda^{\mathrm{NN}}$ can be implemented through a modified Viterbi decoder.

First, note that $\mathcal{Q}_\Lambda^{\mathrm{NN}}$ solves the following optimization problem

$$\begin{aligned} \text{minimize} \quad & \|\boldsymbol{\lambda} - \alpha \mathbf{y}\| \\ \text{subject to} \quad & \boldsymbol{\lambda} \in \Lambda. \end{aligned} \tag{G.1}$$

Second, note that the problem (G.1) is equivalent to

$$\begin{aligned} \text{minimize} \quad & \|\tilde{\sigma}(\mathbf{c}) + \boldsymbol{\lambda}' - \alpha \mathbf{y}\| && \text{(G.2)} \\ \text{subject to} \quad & \mathbf{c} \in \mathcal{C} && \text{(G.3)} \\ & \boldsymbol{\lambda}' \in \Lambda'. \end{aligned}$$

This is because each lattice point $\boldsymbol{\lambda} \in \Lambda$ can be expressed as $\boldsymbol{\lambda} = \tilde{\sigma}(\mathbf{c}) + \Lambda'$, where $\mathbf{c} = \sigma(\boldsymbol{\lambda})$ and $\boldsymbol{\lambda}' \in \Lambda'$.

Third, note that Problem (G.2) is equivalent to

$$\begin{aligned} \text{minimize} \quad & \|[\tilde{\sigma}(\mathbf{c}) - \alpha \mathbf{y}] \bmod \Lambda'\| \\ \text{subject to} \quad & \mathbf{c} \in \mathcal{C}, \end{aligned} \tag{G.4}$$

where $[\mathbf{x}] \bmod \Lambda'$ is defined as $[\mathbf{x}] \bmod \Lambda' \triangleq \mathbf{x} - \mathcal{Q}_{\Lambda'}^{\mathrm{NN}}(\mathbf{x})$. This is because $\boldsymbol{\lambda}' = -\mathcal{Q}_{\Lambda'}^{\mathrm{NN}}(\tilde{\sigma}(\mathbf{c}) - \alpha\mathbf{y})$ solves Problem (G.2) for any $\mathbf{c} \in \mathcal{C}$.

Now it is easy to see the problem (G.4) can be solved through a modified Viterbi decoder with the metric given by $\|[\cdot] \bmod \Lambda'\|$ instead of $\|\cdot\|$. Therefore, the nearest neighbor quantizer $\mathcal{Q}_{\Lambda}^{\mathrm{NN}}$ can be implemented through a modified Viterbi decoder.

Appendix H

Proof of Theorem 3.5

First, we show the existence of the solution $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ by induction on m.

If $m = 1$, then the vector $\mathbf{a}_1$ can be chosen such that $\mathbf{a}_1\mathbf{L}$ is one of the shortest lattice points. Note that $\mathbf{a}_1$ is not divisible by π; otherwise it will not be one of the shortest lattice points. In other words, $\bar{\mathbf{a}}_1$ is indeed nonzero. Hence, the solution $\mathbf{a}_1$ always exists when $m = 1$.

Now suppose the solution $\{\mathbf{a}_1, \ldots, \mathbf{a}_k\}$ exists when $k < m$. We will show the existence of the vector $\mathbf{a}_{k+1}$.

Consider the following set

$$\mathcal{A} = \{\mathbf{a} \in T^L : \bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_k, \bar{\mathbf{a}} \text{ are linearly independent}\}.$$

Clearly, the set $\mathcal{A}$ is nonempty, since $k < m$. Then the vector $\mathbf{a}_{k+1}$ can be chosen as

$$\mathbf{a}_{k+1} = \arg\min_{\mathbf{a}\in\mathcal{A}} \|\mathbf{a}\mathbf{L}\|.$$

This proves the existence of the vector $\mathbf{a}_{k+1}$, which completes the induction.

Second, we show that the solution $\{\mathbf{a}_1, \ldots, \mathbf{a}_m\}$ is a dominant solution by induction on m.

If $m = 1$, then $\|\mathbf{a}_1\mathbf{L}\| \leq \|\mathbf{b}_1\mathbf{L}\|$ for any feasible solution $\mathbf{b}_1$, since $\mathbf{a}_1\mathbf{L}$ is one of the shortest lattice points.

Now suppose that $\{\mathbf{a}_1, \ldots, \mathbf{a}_k\}$ is a dominant solution when $k < m$. We will show that $\{\mathbf{a}_1, \ldots, \mathbf{a}_k, \mathbf{a}_{k+1}\}$ is also a dominant solution.

Suppose that $\{\mathbf{b}_1, \ldots, \mathbf{b}_k, \mathbf{b}_{k+1}\}$ is a feasible solution with $\|\mathbf{b}_1\mathbf{L}\| \leq \ldots \leq \|\mathbf{b}_{k+1}\mathbf{L}\|$. Since $\bar{\mathbf{b}}_1, \ldots, \bar{\mathbf{b}}_k$

are linearly independent, we have

$$\|\mathbf{a}_i\mathbf{L}\| \leq \|\mathbf{b}_i\mathbf{L}\|,\ i = 1, \ldots, k.$$

It remains to show $\|\mathbf{a}_{k+1}\mathbf{L}\| \leq \|\mathbf{b}_{k+1}\mathbf{L}\|$. We consider the following two cases.

1. If there exists some $\mathbf{b}_i$ $(i = 1, \ldots, k+1)$ such that $\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_k, \bar{\mathbf{b}}_i$ are linearly independent, then by the construction of $\mathbf{a}_{k+1}$, we have

$$\|\mathbf{a}_{k+1}\mathbf{L}\| \leq \|\mathbf{b}_i\mathbf{L}\| \leq \|\mathbf{b}_{k+1}\mathbf{L}\|.$$

2. Otherwise, each $\bar{\mathbf{b}}_i$ can be expressed as a linear combination of $\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_k$. That is,

$$\bar{\mathbf{b}}_i \in \text{Span}\{\bar{\mathbf{a}}_1, \ldots, \bar{\mathbf{a}}_k\}.$$

   This is contrary to the fact that $\bar{\mathbf{b}}_1, \ldots, \bar{\mathbf{b}}_{k+1}$ are linearly independent, since any $k+1$ vectors in a vector space of dimension k are linearly dependent.

Therefore, we have $\|\mathbf{a}_{k+1}\mathbf{L}\| \leq \|\mathbf{b}_{k+1}\mathbf{L}\|$, which completes the induction.

Appendix I

Proofs for Section 5.3

I.1 Proof of Proposition 5.1

We prove the claims one by one.

1. The presence of a pivot p in a column rules out the possibility of another pivot in the same column and below p, since all entries in the same column below p must be zero and hence cannot be pivots.

2. Deleting a row of A does not influence the value or the position of the pivots in the other rows; thus it easy to verify that the modified matrix satisfies the four conditions required for a matrix to be in row canonical form.

3. By definition p_k has degree smaller than or equal to that of any element in its row. If A contained an element in a row below row k of degree smaller than d_k, then the pivot of that row would have degree smaller than d_k, contradicting the property that pivots of smaller degree must occur above pivots of larger degree.

4. By definition p_k is the earliest element having minimum degree in row k, so every element in row k occurring earlier than p_k has degree strictly larger than d_k. We know from 3) that A contains no element in a row below k of degree smaller than d_k. If such a row contains an element of degree equal to d_k, then the pivot of that row must occur later than p_k, which implies that every element occurring in that row occurring in column c_k or earlier has degree strictly larger than d_k.

5. Consider w_j. From 3) we know that p_1 divides every element of A; in particular, p_1 divides every element of column j of A. Since w_j is a linear combination of these elements, it must be that p_1 divides w_j.

6. If $j < c_1$, we know from 4) that every element in column j of A has degree strictly greater than d_1 and so does every linear combination of these elements, in particular w_j.

I.2 Proof of Proposition 5.3

If A is the zero matrix, then its row canonical form must also be the zero matrix, which is therefore unique. Thus let us assume that A is nonzero.

We will proceed by induction on n. For $n = 1$, the proof is obvious. Thus suppose that $n > 1$, and let B and C be two row canonical forms of A. Clearly, $\operatorname{row} B = \operatorname{row} C$, and each row of B and C are elements of $\operatorname{row} A$. Let $B[1, j_1]$ and $C[1, j_2]$ be the pivots in the first row of B and C, respectively. From Proposition 5.1–5 we have that $B[1, j_1] \mid C[1, j_2]$ and $C[1, j_2] \mid B[1, j_1]$; thus $B[1, j_1]$ and $C[1, j_2]$ are associates. However, since pivot elements must take the form π^l for some l, we conclude that $B[1, j_1] = C[1, j_2]$. Suppose $j_1 < j_2$. By Proposition 5.1–6 we have $\deg(B[1, j_1]) > \deg(C[1, j_2])$, contradicting the fact that $B[1, j_1] = C[1, j_2]$. A similar contradiction arises if $j_1 > j_2$. We conclude that $j_1 = j_2$, i.e., both B and C must have exactly the same pivot element in exactly the same position in their first row.

Now let $j_1 = j_2 = j$. Consider the submodule of $\operatorname{row} A$ in which every element has zero in its jth component. Every element a of this submodule is a linear combination

$$a = \sum_{i=1}^{n} b_i B[i, :];$$

for some choice of coefficients $b_1, \ldots, b_n$. However, since $a_j = 0$, and $B[i, j] = 0$ for $i > 2$, we must have $b_1 B[1, j] = 0$. Since $B[1, j]$ is the pivot element of the first row of B, it divides every element of that row; thus if $b_1 B[1, j] = 0$, then $b_1 B[1, :] = 0$, i.e., the first row can only contribute 0 to a. This means that the given submodule is equal to $\operatorname{row} B[2{:}n, 1{:}m]$. Similarly, the given submodule is also equal to $\operatorname{row} C[2{:}n, 1{:}m]$. By Proposition 5.1–2, both $B[2{:}n, 1{:}m]$ and $C[2{:}n, 1{:}m]$ are in row canonical form. Thus by induction, we have $B[2{:}n, 1{:}m] = C[2{:}n, 1{:}m]$. This implies that B and C can differ in their first row only.

Let us assume that $B[1, :] \neq C[1, :]$, i.e., that the first rows of B and C are not equal, so that $\Delta = (\delta_1, \ldots, \delta_m) = B[1, :] - C[1, :]$ is nonzero. Since Δ is an element of $\operatorname{row} A$ with zero in its jth component, we have $\Delta \in \operatorname{row} B[2{:}n, 1{:}m]$, from which it follows that

$$\Delta = \sum_{i=2}^{n} c_i B[i, :],$$

for some $c_2, \ldots, c_n \in R$. If $B[2{:}n, 1{:}m]$ is the zero matrix, then $\Delta = 0$, which is a contradiction. Otherwise, let $B[2, j_3]$ be the pivot of $B[2, :]$. Note, on the one hand, that $B[i, j_3] = 0$ for all $i > 2$; thus $\delta_{j_3} = c_2 B[2, j_3]$, i.e., δ_{j_3} must be a multiple of $B[2, j_3]$. On the other hand, because $B[2, j_3]$ and $C[2, j_3]$ are (identical) pivots, $B[1, j_3]$, $C[1, j_3] \in \mathcal{R}(R, B[2, j_3])$. If $B[1, j_3]$ and $C[1, j_3]$ are distinct, their difference, δ_{j_3}, cannot be a multiple of $B[2, j_3]$. We conclude that $\delta_{j_3} = 0$, i.e., $B[1, j_3]$ and $C[1, j_3]$ are not distinct. Since $B[2, j_3]$ is the pivot of $B[2, :]$ it divides every element of $B[2, :]$; thus if $c_2 B[2, j_3] = 0$, then $c_2 B[2, :] = 0$. Continuing this argument, we have $c_i B[i, :] = 0$ for all $i \geq 2$. Therefore, we have $\Delta = 0$, which is a contradiction. This establishes uniqueness.

Appendix J

Proofs for Section 5.4

J.1 Proof of Lemma 5.1

Let $\mathcal{S}$ denote the set of row canonical forms in $\mathcal{T}_\kappa(R^{n\times\mu})$, and let $\mathcal{G}$ denote the set of submodules of R^μ with shape κ. Let $\phi : \mathcal{S} \to \mathcal{G}$ be the map that takes a matrix $B \in \mathcal{S}$ to its row module $\operatorname{row} B$. We will show that ϕ is a one-to-one correspondence.

If $\phi(B_1) = \phi(B_2)$ then B_1 and B_2 are left-equivalent, and so B_2 is a row canonical form of B_1 and vice-versa. By the uniqueness of the row canonical form, we have $B_1 = B_2$; thus ϕ is injective.

Now let M be a submodule of R^μ with $\operatorname{shape} M = \kappa$, and construct a matrix A such that every element in M is a row of A. Clearly, $\operatorname{row} A = M$ and $\operatorname{shape} A = \kappa$. Since $\kappa \preceq n$, $\mathrm{RCF}(A)$ has at most n nonzero rows. Let B be the submatrix of $\mathrm{RCF}(A)$ consisting of the top n rows. Then we have $\operatorname{shape} B = \operatorname{shape} A = \kappa$. Hence, $B \in \mathcal{T}_\kappa(R^{n\times\mu})$, and the map ϕ is surjective.

J.2 Proof of Proposition 5.4

We will show that (i) every X constructed as above is a principal row canonical form, and (ii) every principal row canonical form has a π-adic decomposition following the above conditions.

We begin with Claim (i). First, we track the diagonal entries in X. Clearly, by construction, the first κ_1 diagonal entries in X are 1; they are contributed by X_0. The next $\kappa_2 - \kappa_1$ diagonal entries in X are π; they are contributed by X_1. Continuing this argument, we conclude that the diagonal entries in X are indeed of the form (5.7).

Second, we show that X satisfies all the four conditions for row canonical forms.

1. By construction, the first κ_s rows of X are the only nonzero rows. Hence, X satisfies Condition 1.

2. It suffices to show that the nonzero diagonal entries are precisely the pivots in X. Suppose that the ith diagonal entry $X[i,i] = \pi^l$. Then by construction, π^l is contributed by X_l and $\kappa_l < i \leq \kappa_{l+1}$. Note that for each auxiliary matrix $X_{l'}$, only the first $\kappa_{l'+1}$ rows are nonzero. Thus, the ith row in $X_{l'}$ is zero for all $l' = 0, \ldots, l-1$. In particular, $X_{l'}[i,j] = 0$, for all $l' = 0, \ldots, l-1$ and for all $j > i$. Therefore, we have, for all $j > i$,

$$\begin{aligned} X[i,j] &= \sum_{l'=0}^{s-1} \pi^{l'} X_{l'}[i,j] \\ &= \sum_{l'=l}^{s-1} \pi^{l'} X_{l'}[i,j] \\ &= \pi^l \sum_{l'=l}^{s-1} \pi^{l'-l} X_{l'}[i,j]. \end{aligned}$$

That is, every $X[i,j]$ is a multiple of π^l whenever $j > i$. On the other hand, by construction, $X[i,j] = 0$ whenever $j < i$. It follows that $X[i,i]$ is indeed the pivot of row i. Hence, X satisfies Condition 2.

3. Since the nonzero diagonal entries are the pivots, X satisfies Condition 3.

4. Suppose that the ith pivot $X[i,i] = \pi^l$. Then, we have $\kappa_l < i \leq \kappa_{l+1}$. Note that for each auxiliary matrix $X_{l'}$, all other entries in column i are zero as long as $l' \geq l$. Thus, we have, for all $j \neq i$,

$$\begin{aligned} X[j,i] &= \sum_{l'=0}^{s-1} \pi^{l'} X_{l'}[j,i] \\ &= \sum_{l'=0}^{l-1} \pi^{l'} X_{l'}[j,i]. \end{aligned}$$

It follows that $X[j,i] \in \mathcal{R}(R, \pi^l)$ for all $j \neq i$. Hence, X satisfies Condition 4.

We turn now to Claim (ii). Let X be a principal row canonical form in $\mathcal{T}_\kappa(R^{n\times\mu})$. Then the diagonal entries in each X_i must satisfy

$$X_i[1,1], \ldots, X_i[\kappa_{i+1}, \kappa_{i+1}] = \underbrace{0, \ldots, 0}_{\kappa_i}, \underbrace{1, \ldots, 1}_{\kappa_{i+1}-\kappa_i}.$$

Moreover, since X satisfies Condition 4, it follows that each X_i satisfies the first condition described above. Since X satisfies Condition 2, it follows that $X_i[\kappa_{i+1}+1{:}n, 1{:}m]$ is a zero matrix. Finally, due to the constraints imposed by μ, $X_i[1{:}n, \mu_{i+1}+1{:}m]$ is a zero matrix for all i. Therefore, each X_i satisfies the second and third conditions. This completes the proof.

J.3 Proof of Theorem 5.1

We need two technical lemmas. The first lemma is a natural extension of the well-known rank decomposition.

Lemma J.1. *Let B be the row canonical form of $A \in R^{n\times m}$. Let $\tilde{B}$ be the submatrix of B consisting of only nonzero rows. Then A can be decomposed as a product $P_1\tilde{B}$ of some full-column-rank matrix P_1 and the matrix $\tilde{B}$. Moreover, the number of P_1 producing such a decomposition is $q^{n\sum_{i=1}^{s-1} i(\kappa_{i+1}-\kappa_i)}$, where $\kappa = \operatorname{shape} A$.*

Proof. Since B is the row canonical form of A, $A = PB$ for some invertible matrix $P \in \mathrm{GL}_n(R)$. Since $\kappa = \operatorname{shape} A = \operatorname{shape} B$, B has κ_s nonzero rows, and $\tilde{B} \in R^{\kappa_s\times m}$. Let $P = \begin{bmatrix} P_1 & P_2 \end{bmatrix}$, where $P_1 \in R^{n\times\kappa_s}$ and $P_2 \in R^{n\times(n-\kappa_s)}$. Then we have

$$A = PB = \begin{bmatrix} P_1 & P_2 \end{bmatrix} \begin{bmatrix} \tilde{B} \\ 0 \end{bmatrix} = P_1\tilde{B}.$$

Since P is invertible, P_1 is full column rank.

Next, we count the number of such decompositions. Consider the matrix equation $X\tilde{B} = P_1\tilde{B}$, in unknown X. Clearly, the number of decompositions of A is equal to the number of solutions to this matrix equation. Let $\tilde{B}[i, j_i]$ be the pivot of the ith row of $\tilde{B}$, for all $i = 1, \ldots, \kappa_s$. Then $\tilde{B}[i, j_i]$ divides the ith row of $\tilde{B}$. It follows that $\tilde{B} = DB'$, where $D = \operatorname{diag}\left(\tilde{B}[1, j_1], \ldots, \tilde{B}[\kappa_s, j_{\kappa_s}]\right)$, and the ith row of B' is equal to the ith row of $\tilde{B}$ divided by $\tilde{B}[i, j_i]$. Clearly, $B'[i, j_i] = 1$ for all $i = 1, \ldots, \kappa_s$. Since $j_1, \ldots, j_{\kappa_s}$ are all distinct, $\operatorname{shape} B' = (\kappa_s, \ldots, \kappa_s)$, which implies that B' is full row rank. By Lemma 2.1, $(XD - P_1D)B' = 0$ if and only if $XD - P_1D = 0$. Hence, $X\tilde{B} = P_1\tilde{B}$ if and only if $XD = P_1D$. Thus, it suffices to count the number of solutions to $XD = P_1D$. Note that $XD = P_1D$ is equivalent to the following system of equations

$$X[i, k]\tilde{B}[k, j_k] = P_1[i, k]\tilde{B}[k, j_k], i = 1, \ldots, n,\ k = 1, \ldots \kappa_s. \tag{J.1}$$

Suppose that $\tilde{B}[k, j_k] = \pi^{l_k}$ for some $0 \le k < s$. Then it is easy to check that the equation $X[i, k]\pi^{l_k} = P_1[i, k]\pi^{l_k}$ has exactly q^{l_k} solutions for $X[i, k]$. It follows that (J.1) has exactly $q^{n(l_1+\cdots+l_{\kappa_s})}$ solutions. Finally, by using the fact that $\sum_{k=1}^{\kappa_s} l_k = \sum_{i=1}^{s-1} i(\kappa_{i+1} - \kappa_i)$, we complete the proof. □

Lemma J.2. *The number of matrices in $R^{n\times\mu}$ having a given row canonical form in $\mathcal{T}_\kappa(R^{n\times\mu})$ is equal*

to

$$|R^{n\times\kappa}| \prod_{i=0}^{\kappa_s-1} (1-q^{i-n}).$$

Proof. Let B be a row canonical form in $\mathcal{T}_\kappa(R^{n\times\mu})$. Let $\tilde{B}$ be the submatrix of B consisting of only nonzero rows. Clearly, $\tilde{B} \in R^{\kappa_s\times\mu}$. We would like to count the number of matrices in $R^{n\times\mu}$ having the row canonical form B.

By Lemma J.1, every matrix A with $\mathrm{RCF}(A) = B$ has $q^{n\sum_{i=1}^{s-1} i(\kappa_{i+1}-\kappa_i)}$ decompositions of the form $A = C\tilde{B}$ for some full-column-rank $C \in R^{n\times\kappa_s}$. Hence, the number of matrices in $R^{n\times\mu}$ having the row canonical form B is equal to the number of full-column-rank matrices of size $n \times \kappa_s$ divided by $q^{n\sum_{i=1}^{s-1} i(\kappa_{i+1}-\kappa_i)}$, which can be simplified to $|R^{n\times\kappa}| \prod_{i=0}^{\kappa_s-1}(1-q^{i-n})$. □

We can partition all the matrices in $\mathcal{T}_\kappa(R^{n\times\mu})$ based on their row canonical forms: two matrices belong to the same class if and only if they have the same row canonical form. By Lemma 5.1, the number of such classes is $\begin{bmatrix}\mu\\ \kappa\end{bmatrix}_q$. By Lemma J.2, the number of matrices in each class is $|R^{n\times\kappa}| \prod_{i=0}^{\kappa_s-1}(1-q^{i-n})$. Combining these two results gives us Theorem 5.1.

Appendix K

Proof of the Stability Region

The proof here extends the proof of [97] by incorporating the physical-layer effect of compute-and-forward. The main steps of the proof are the following. First, we study the evolution of the system when N tends to infinity. As we will see, the behavior of this limiting system can be characterized by a (deterministic) dynamical system. Second, we provide sufficient and necessary conditions for the global stability of the dynamical system. Finally, we connect the stability of the initial system with N users to the stability of the dynamical system.

K.1 Evolution of the Limiting System

Consider a system with N users. We assume that users can be categorized among a finite set $\mathcal{V}$ of $V = |\mathcal{V}|$ classes, each with the same transmission probability and arrival rate. The transmission probability of user i of class v is $p_i = p_v/N$; the arrival rate of user i of class v is $\lambda_i = \lambda_v/N$. We further assume that the proportion of users in class v tends to β_v as $N \to \infty$.

Now consider a class-v user. When it has a packet in its buffer, it transmits with probability p_v/N. Suppose that it is in state $x = (v, k)$, the probabilities of transition for the next slot are given as follows. The state becomes $(v, k+1)$ with probability $F_b^N/N = \lambda_v/N + o(1/N)$, and $(v, k-1)$ with probability

$$F_d^N/N = o(1/N) + 1_{k>0} \frac{p_v/N}{1 - p_v/N} \prod_{v'} (1 - p_{v'}/N)^{\mu_{v'}(t)\beta_{v'}N} \left(q_1 + q_2 \left(\sum_{v'} \mu_{v'}(t)\beta_{v'} N \frac{p_{v'}/N}{1 - p_{v'}/N} \right) - q_2 \frac{p_v/N}{1 - p_v/N} \right)$$

where $\mu_v(t)$ denotes the proportion of users of class v with nonempty buffers at time slot t.

When $N \to \infty$, the functions F_b^N, F_d^N converge to F_b, F_d where

$$F_b = \lambda_v,$$

and

$$F_d = 1_{k>0} p_v \exp\left(-\sum_{v'} \beta_{v'} \mu_{v'}(t) p_{v'}\right)\left(q_1 + q_2 \sum_{v'} \beta_{v'} \mu_{v'}(t) p_{v'}\right).$$

We next rescale the time so that time τ corresponds to the time slot $\lfloor N\tau \rfloor$. Let $Q_{(v,k)}(\tau)$ denote the limiting probability that a user of class v has k packets in its buffer. We have

$$\begin{aligned}\frac{\partial Q_{(v,k)}(\tau)}{\partial \tau} &= \lambda_v \left(1_{k>0} Q_{(v,k-1)}(\tau) - Q_{(v,k)}(\tau)\right) \\ &\quad + p_v \exp(-\gamma(\tau))(q_1 + q_2\gamma(\tau)) \left(Q_{(v,k+1)}(\tau) - 1_{k>0} Q_{(v,k)}(\tau)\right)\end{aligned}$$

with $\gamma(\tau) = \sum_v \beta_v p_v \mu_v(\tau) = \sum_v \beta_v p_v \left(1 - Q_{(v,0)}(\tau)\right)$. Let $W_v(\tau) = \sum_k k Q_{(v,k)}(\tau)$ be the workload of a class-v user. Then the evolution of $W_v(\tau)$ can be expressed as

$$\frac{\partial W_v(\tau)}{\partial \tau} = \lambda_v - p_v \exp(-\gamma(\tau))(q_1 + q_2\gamma(\tau)) \left(1 - Q_{(v,0)}(\tau)\right).$$

Finally, let $W(\tau) = \sum_v \beta_v W_v(\tau)$ denote the total workload. Then the evolution of $W(\tau)$ can be expressed as

$$\frac{\partial W(\tau)}{\partial \tau} = \sum_v \beta_v \lambda_v - \left(q_1\gamma(\tau) + q_2\gamma^2(\tau)\right) \exp(-\gamma(\tau)).$$

K.2 Stability of the Limiting System

Let $\lambda = \sum_v \beta_v \lambda_v$. Assume that $0 < \lambda < \max_x (q_1 x + q_2 x^2) e^{-x}$. First, we will show that the equation

$$(q_1 x + q_2 x^2) e^{-x} = \lambda$$

has exactly two solutions in $(0, \infty)$. Let $f(x) = (q_1 x + q_2 x^2) e^{-x}$. Then the derivative of $f(x)$ is given by

$$f'(x) = e^{-x} \left(q_1 + (2q_2 - q_1)x - q_2 x^2\right).$$

By Descartes' rule of signs, $q_1 + (2q_2 - q_1)x - q_2 x^2$ has exactly one zero in $(0, \infty)$. It follows that $f(x)$ is first increasing and then decreasing over $(0, \infty)$. Hence, the equation $f(x) = \lambda$ has precisely two solutions in $(0, \infty)$. In the following we denote by $\underline{\gamma}(\lambda)$ and $\overline{\gamma}(\lambda)$ these two solutions where $\underline{\gamma}(\lambda) < \overline{\gamma}(\lambda)$.

Let $\zeta = \sum_v \beta_v p_v$. Then the stability of the dynamical system is given by:

1. If $\zeta < \overline{\gamma}(\lambda)$ and
$$\forall v \in \mathcal{V}, \lambda_v < p_v(q_1 + q_2\underline{\gamma}(\lambda))\exp(-\underline{\gamma}(\lambda)),$$
then the dynamical system is globally stable, and $\zeta > \underline{\gamma}(\lambda)$.
2. If $\zeta < \underline{\gamma}(\lambda)$ or if some $\lambda_v > p_v(q_1 + q_2\underline{\gamma}(\lambda))\exp(-\underline{\gamma}(\lambda))$, then the dynamical system is unstable.
3. If $\zeta > \overline{\gamma}(\lambda)$, then the system is not globally stable.

The above result can be obtained following the steps of the proof of Theorem 5 in [97]. Since the proof in [97] only uses the property that $f(x)$ first increases and then decreases, it can be easily adapted to our case here. Essentially, the proof technique is based on the probabilistic interpretation of the dynamical system as a collection of V queues: a queue parameterized by v has Poisson arrivals of rate λ_v, and it is served at rate $p_v(q_1 + q_2\gamma(\tau))\exp(-\gamma(\tau))$ at time τ.

The above result says that the stability region of the limiting system is given by

$$\tilde{\Lambda} = \left\{ \lambda \in \mathbb{R}_+^V : \forall v, \lambda_v = \rho_v p_v \left(q_1 + q_2 \sum_u \beta_u \rho_u p_u \right) e^{-\sum_u \beta_u \rho_u p_u} \right\}.$$

Next, we need to show that it is equivalent to the approximate stability region. Note that $\tilde{\Lambda}$ is precisely the image of the following map:

$$\begin{bmatrix} g_1 \\ \vdots \\ g_V \end{bmatrix} \to e^{-\sum_v \beta_v g_v} \left(q_1 + q_2 \sum_v \beta_v g_v \right) \begin{bmatrix} g_1 \\ \vdots \\ g_V \end{bmatrix},$$

where $g_v \in [0, p_v]$, or written in vector form, $g \in \mathcal{P} \triangleq [0, p_1] \times \cdots \times [0, p_V]$. The Jacobian matrix of this map is given by

$$J = e^{-\langle \beta, g \rangle} \left((q_1 + q_2\langle \beta, g \rangle) I_V - (q_1 - q_2 + q_2\langle \beta, g \rangle) g\beta^T \right)$$

with determinant

$$\det(J) = e^{-V\langle \beta, g \rangle}(q_1 + q_2\langle \beta, g \rangle) \left((q_1 + q_2\langle \beta, g \rangle) - (q_1 - q_2 + q_2\langle \beta, g \rangle)\langle \beta, g \rangle \right).$$

Recall that $q_1 + (2q_2 - q_1)x - q_2x^2$ has exactly one zero in $(0, \infty)$ given by

$$x_0 = \frac{\sqrt{q_1^2 + 4q_2^2} - q_1 + 2q_2}{2q_2}.$$

Hence, if $\langle \beta, g \rangle$ is less than x_0 for all $g \in \mathcal{P}$, the Jacobian determinant $\det(J)$ is always nonzero. By the inverse function theorem, the boundary of $\mathcal{P}$ is mapped to the boundary of $\tilde{\Lambda}$. Therefore, the above stability region is indeed the same as the approximate stability region when $\langle \beta, p \rangle < x_0$.

K.3 Stability of the Finite System

To conclude the proof of Theorem 6.1, we need to relate the stability of the dynamical system to the stability region of the finite system. The proof is a standard stochastic coupling argument. Here, we only prove the sufficient condition, since the necessary condition can be handled in a similar way.

We prove by induction on V. When $V = 1$, this case is the fully-symmetric case. As shown in [90], the system is stable if and only if

$$\lambda_v < q_1 p_v \left(1 - \frac{p_v}{N}\right)^{N-1} + q_2 \frac{N(N-1)}{N^2} p_v^2 \left(1 - \frac{p_v}{N}\right)^{N-2}.$$

Now assume that $\lambda_v < q_1 p_v \left(1 - \frac{p_v}{N}\right)^{N-1} + q_2 \frac{N(N-1)}{N^2} p_v^2 \left(1 - \frac{p_v}{N}\right)^{N-2} - \epsilon$. For a particular queue, its distribution at any time is stochastically bounded by the distribution one would obtain when all the other queues are saturated, which is that of a Markovian queue of load

$$\frac{\lambda_v}{q_1 p_v \left(1 - \frac{p_v}{N}\right)^{N-1} + q_2 \frac{N(N-1)}{N^2} p_v^2 \left(1 - \frac{p_v}{N}\right)^{N-2}} < 1 - \alpha\epsilon$$

for some $\alpha > 0$.

Suppose that the sufficient condition is true when $|\mathcal{V}| \leq V$. We will show that it is also true when $|\mathcal{V}| = V + 1$. Consider the stochastically dominant system where all queues of class v are saturated. We apply the induction result and use a similar argument as above, concluding that for N large enough, the dominant system without queues of class v is stable.

Bibliography

[1] "Cisco visual networking index: Global mobile data traffic forecast update, 2013-2018," *Cisco Inc.*, Feb. 2014.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul. 2000.

[3] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[4] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 782–795, Oct. 2003.

[5] S. Zhang, S.-C. Liew, and P. P. Lam, "Hot topic: Physical layer network coding," in *Proc. of ACM Int. Conf. on Mobile Compu. and Netw.*, Los Angeles, CA, USA, Sep. 24–29, 2006, pp. 358–365.

[6] P. Popovski and H. Yomo, "The anti-packets can increase the achievable throughput of a wireless multi-hop network," in *Proc. of IEEE Int. Conf. on Commun.*, Istanbul, Turkey, Jun. 11–15, 2006, pp. 3885–3890.

[7] B. Nazer and M. Gastpar, "Computing over multiple-access channels with connections to wireless network coding," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Seattle, USA, Jul. 9–14, 2006, pp. 1354–1358.

[8] P. Popovski and T. Koike-Akino, "Coded bidirectional relaying in wireless networks," in *New Directions in Wireless Communications Research*, V. Tarokh, Ed. Springer, 2009, pp. 291–316.

[9] B. Nazer and M. Gastpar, "Reliable physical layer network coding," *Proc. IEEE*, vol. 99, no. 3, pp. 438–460, Mar. 2011.

[10] S.-C. Liew, S. Zhang, and L. Lu, "Physical-layer network coding: Tutorial, survey, and beyond," *Physical Communication*, vol. 6, no. 1, pp. 4–42, Mar. 2013.

[11] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.

[12] K. R. Narayanan, M. P. Wilson, and A. Sprintson, "Joint physical layer coding and network coding for bi-directional relaying," in *Proc. Allerton Conf. Commun. Control Comput.*, Monticello, IL, Sep. 2007.

[13] M. P. Wilson, K. R. Narayanan, H. D. Pfister, and A. Sprintson, "Joint physical layer coding and network coding for bidirectional relaying," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5641–5654, Nov. 2010.

[14] W. Nam, S.-Y. Chung, and Y. H. Lee, "Capacity bounds for two-way relay channels," in *Proc. of Int. Zurich Seminar on commun.*, Zurich, Switzerland, Mar. 12–14, 2008.

[15] ——, "Capacity of the Gaussian two-way relay channel to within 1/2 bit," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5488–5494, Nov. 2010.

[16] M. P. Wilson and K. R. Narayanan, "Power allocation strategies and lattice based coding schemes for bi-directional relaying," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Coex, Seoul, Korea, Jun. 28 – Jul. 3, 2009, pp. 344–348.

[17] A. S. Avestimehr, A. Sezgin, and D. N. C. Tse, "Capacity of the two-way relay channel within a constant gap," *Eur. Trans. Telecomms.*, vol. 21, no. 4, pp. 363–374, Jun. 2010.

[18] U. Niesen and P. Whiting, "The degrees of freedom of compute-and-forward," *IEEE Trans. Inf. Theory*, vol. 58, no. 8, pp. 5214–5232, Aug. 2012.

[19] U. Niesen, B. Nazer, and P. Whiting, "Computation alignment: Capacity approximation without noise accumulation," *IEEE Trans. Inf. Theory*, vol. 59, no. 6, pp. 3811–3832, Jun. 2013.

[20] J. Zhan, B. Nazer, M. Gastpar, and U. Erez, "MIMO compute-and-forward," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Seoul, South Korea, Jun. 28– Jul. 3, 2009, pp. 2848–2852.

[21] S.-N. Hong and G. Caire, "Compute-and-forward strategy for cooperative distributed antenna systems," *IEEE Trans. Inf. Theory*, vol. 59, no. 9, pp. 5227–5243, Sep. 2013.

[22] O. Ordentlich, U. Erez, and B. Nazer, "The approximate sum capacity of the symmetric Gaussian k-user interference channel," *IEEE Trans. Inf. Theory*, vol. 60, no. 6, pp. 3450–3482, Jun. 2014.

[23] J. Zhu and M. Gastpar, "Lattice codes for many-to-one interference channels with and without cognitive messages," *Computing Research Repository (CoRR)*, Apr. 2014, submitted to the IEEE Trans. Inf. Theory. [Online]. Available: http://arxiv.org/pdf/1404.0273.pdf

[24] A. S. Avestimehr, S. N. Diggavi, and D. N. C. Tse, "Wireless network information flow: A deterministic approach," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 1872–1905, Apr. 2011.

[25] S. H. Lim, Y.-H. Kim, A. E. Gamal, and S.-Y. Chung, "Noisy network coding," *IEEE Trans. Inf. Theory*, vol. 57, no. 5, pp. 3132–3152, May 2011.

[26] U. Erez and R. Zamir, "Achieving $\frac{1}{2}\log(1+\mathrm{SNR})$ on the AWGN channel with lattice encoding and decoding," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.

[27] K. N. Pappi, G. K. Karagiannidis, and R. Schober, "How sensitive is compute-and-forward to channel estimation errors," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Istanbul, Turkey, Jul. 7–12, 2013, pp. 3110–3114.

[28] B. Hassibi and B. M. Hochwald, "How much training is needed in multiple-antenna wireless links," *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 951–963, Apr. 2003.

[29] B. R. McDonald, *Finite Rings with Identity*. Marcel Dekker, Inc., 1974.

[30] ——, *Linear Algebra over Commutative Rings*. New York: Marcel Dekker, Inc., 1984.

[31] W. C. Brown, *Matrices over Commutative Rings*. New York: Marcel Dekker, Inc., 1993.

[32] G. H. Norton and A. Sălăgean, "On the structure of linear and cyclic codes over a finite chain ring," *Appl. Algebra Eng. Commun. Comput.*, vol. 10, no. 6, pp. 489–506, 2000.

[33] T. Honold and I. Landjev, "Linear codes over finite chain rings," *The Electronic Journal of Combinatorics*, vol. 7, 2000.

[34] D. S. Dummit and R. M. Foote, *Abstract Algebra*, 3rd ed. John Wiley & Sons, Inc., 2004.

[35] A. A. Nechaev, "Finite rings with applications," in *Handbook of Algebra*, M. Hazewinkel, Ed. North-Holland, 2008, vol. 5, pp. 213–320.

[36] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, 3rd ed. New York: Springer-Verlag, 1999.

[37] R. Zamir, *Lattice Coding for Signals and Networks*. Cambridge University Press, 2014.

[38] A. Özgür and S. N. Diggavi, "Approximately achieving Gaussian relay network capacity with lattice codes," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Austin, Texas, USA, Jun. 13–18, 2010, pp. 669–673.

[39] R. Kötter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3579–3591, Aug. 2008.

[40] J. Blomer, "Closest vectors, successive minima, and dual HKZ bases of lattices," in *Proc. of Int. Colloq. Automata, Languages and Programming*, Geneva, Switzerland, Jul. 2000, pp. 248 – 259.

[41] P. Popovski and H. Yomo, "Physical network coding in two-way wireless relay channels," in *Proc. of IEEE Int. Conf. on Commun.*, Glasgow, Scotland, Jun. 24–28, 2007, pp. 707–712.

[42] T. Koike-Akino, P. Popovski, and V. Tarokh, "Optimized constellations for two-way wireless relaying with physical network coding," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 773–787, Jun. 2009.

[43] J.-C. Belfiore, "Lattice codes for the compute-and-forward protocol: The flatness factor," in *IEEE Inf. Workshop*, Paraty, Brazil, Oct. 16–20, 2011, pp. 1876–1880.

[44] B. Hern and K. R. Narayanan, "Multilevel coding schemes for compute-and-forward," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Saint Petersburg, Russia, Jul. 31– Aug. 5, 2011, pp. 1713–1717.

[45] O. Ordentlich, J. Zhan, U. Erez, M. Gastpar, and B. Nazer, "Practical code design for compute-and-forward," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Saint Petersburg, Russia, Jul. 31– Aug. 5, 2011, pp. 1876–1880.

[46] Q. T. Sun, J. Yuan, T. Huang, and K. W. Shum, "Lattice network codes based on Eisenstein integers," *IEEE Trans. Comput.*, vol. 61, no. 7, pp. 2713–2725, Jul. 2013.

[47] N. E. Tunali, K. R. Narayanan, J. J. Boutros, and Y.-C. Huang, "Lattices over Eisenstein integers for compute-and-forward," in *Proc. 2012 Allerton Conf. Commun., Control, and Comput.*, Monticello, IL, Oct. 2012, pp. 33–40.

[48] C. Feng, D. Silva, and F. R. Kschischang, "An algebraic approach to physical-layer network coding," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Austin, TX, Jun. 13–18, 2010, pp. 1017–1021.

[49] ——, "Design criteria for lattice network coding," in *Proc. Conf. Inform. Sci. and Systems*, Baltimore, MD, Mar. 23–25, 2011, pp. 1–6.

[50] ——, "Lattice network coding over finite rings," in *Proc. Canadian Workshop Inf. Theory*, Kelowna, Canada, May 17–20, 2011, pp. 78–81.

[51] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inf. Theory*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005.

[52] V. Ntranos, V. R. Cadambe, B. Nazer, and G. Caire, "Asymmetric compute-and-forward," in *Proc. 2013 Allerton Conf. Commun., Control, and Comput.*, Monticello, IL, Oct. 2013, pp. 1174–1181.

[53] J. H. Conway and N. J. A. Sloane, "A fast encoding method for lattice codes and quantizers," *IEEE Trans. Inf. Theory*, vol. 29, no. 6, pp. 820–824, Nov. 1983.

[54] G. D. Forney, Jr., "Multidimensional constellations—part II: Voronoi constellations," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 6, pp. 941–958, Aug. 1989.

[55] ——, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inf. Theory*, vol. 16, no. 6, pp. 720–738, Nov. 1970.

[56] N. Sommer, M. Feder, and O. Shalvi, "Shaping methods for low-density lattice codes," in *IEEE Inf. Workshop*, Taormina, Sicily, Italy, Oct. 11-16, 2009, pp. 238–242.

[57] J. A. Rush and N. Sloane, "An improvement to the Minkowski-Hlawka bound for packing superballs," *Mathematika*, vol. 34, pp. 8–18, 1987.

[58] O. Shalvi, N. Sommer, and M. Feder, "Signal codes: Convolutional lattice codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 5203–5226, Aug. 2011.

[59] C. Feng, D. Silva, and F. R. Kschischang, "Lattice network coding via signal codes," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Saint Petersburg, Russia, Jul. 31– Aug. 5, 2011, pp. 2642–2646.

[60] O. Ordentlich and U. Erez, "Achieving the gains promised by integer-forcing equalization with binary codes," in *Proc. of 26th Convention of Electr. and Electron. Eng. in Israel*, Eilat, Israel, Nov. 2010, pp. 703–707.

[61] A. Sakzad, M.-R. Sadeghi, and D. Panario, "Turbo lattices: Construction and performance analysis," *submitted to IEEE Trans. Inf. Theory*, 2010.

[62] N. E. Tunali and K. R. Narayanan, "Concatenated signal codes with applications to compute and forward," in *Proc. of IEEE Global Commun. Conf.*, Houston, TX, Dec. 5–9, 2011, pp. 1–5.

[63] B. Hern and K. R. Narayanan, "Multilevel coding schemes for compute-and-forward with flexible decoding," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7613–7631, Nov. 2013.

[64] B. Nazer, "Successive compute-and-forward," in *Proc. of Int. Zurich Seminar on commun.*, Zurich, Switzerland, Feb. 29– Mar. 2, 2012.

[65] J. W. S. Cassels, *An Introduction to the Geometry of Numbers*. Springer-Verlag, 1971.

[66] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, no. 4, pp. 515–534, 1982.

[67] H. Napias, "A generalized of the LLL-algorithm over Euclidean rings or orders," *J. Théorie des Nombres de Bordeaux*, pp. 387–396, 1996.

[68] Y. H. Gan, C. Ling, and W. H. Mow, "Complex lattice reduction algorithm for low-complexity full-diversity MIMO detection," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2701 – 2710, Jul. 2009.

[69] B. Vallée, "Gauss' algorithm revisited," *J. Algorithms*, vol. 12, pp. 556–572, 1991.

[70] H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems," in *Proc. of IEEE Global Commun. Conf.*, Taipei, Taiwan, R.O.C., Nov. 17–21, 2002, pp. 424–428.

[71] P. Q. Nguyen and D. Stehlé, "Low-dimensional lattice basis reduction revisited," *ACM Trans. Algorithms*, vol. 5, no. 46, pp. 1–48, Oct. 2009.

[72] C. Feng, R. W. Nóbrega, F. R. Kschischang, and D. Silva, "Communication over finite-ring matrix channels," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Istanbul, Turkey, Jul. 7–12, 2013, pp. 2890–2894.

[73] R. W. Nóbrega, C. Feng, D. Silva, and B. F. Uchôa-Filho, "On multiplicative matrix channels over finite chain rings," in *Proc. of IEEE Int. Symp. on Network Coding*, Calgary, Canada, Jun. 7–9, 2013.

[74] C. Feng, D. Silva, and F. R. Kschischang, "An algebraic approach to physical-layer network coding," *IEEE Trans. Inf. Theory*, vol. 59, no. 11, pp. 7576–7596, Nov. 2013.

[75] G. D. Forney, Jr., "Coset codes—part II: Binary lattices and related codes," *IEEE Trans. Inf. Theory*, vol. 34, no. 5, pp. 1152–1187, Sep. 1988.

[76] D. Micciancio and O. Regev, "Worst-case to average-case reductions based on Gaussian measures," *SIAM J. on Computing*, vol. 37, no. 1, pp. 267–302, 2007.

[77] C. Ling, L. Luzzi, J.-C. Belfiore, and D. Stehle, "Semantically secure lattice codes for the Gaussian wiretap channel," *submitted to IEEE Trans. Inf. Theory*, Sep. 2012.

[78] D. Silva, F. R. Kschischang, and R. Kötter, "Communication over finite-field matrix channels," *IEEE Trans. Inf. Theory*, vol. 56, no. 3, pp. 1296–1305, Mar. 2010.

[79] A. Montanari and R. L. Urbanke, "Iterative coding for network coding," *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1563–1572, Mar. 2013.

[80] M. Jafari Siavoshani, S. Mohajer, C. Fragouli, and S. Diggavi, "On the capacity of non-coherent network coding," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 1046–1066, Feb. 2011.

[81] S. Yang, S.-W. Ho, J. Meng, E.-h. Yang, and R. W. Yeung, "Linear operator channels over finite fields," *Computing Research Repository (CoRR)*, Feb. 2010. [Online]. Available: http://arxiv.org/abs/1002.2293

[82] R. W. Nóbrega, D. Silva, and B. F. Uchôa-Filho, "On the capacity of multiplicative finite-field matrix channels," *IEEE Trans. Inf. Theory*, vol. 59, no. 8, pp. 4949–4960, Aug. 2013.

[83] J. A. Howell, "Spans in the module $(\mathbb{Z}_m)^s$," *Linear and Multilinear Algebra*, vol. 19, pp. 67–77, 1986.

[84] A. Storjohann, "Algorithms for matrix canonical forms," Ph.D. dissertation, Swiss Federal Institute of Technology – ETH, 2000.

[85] V. V. Vazirani, H. Saran, and B. S. Rajan, "An efficient algorithm for constructing minimal trellises for codes over finite abelian groups," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1839–1854, Nov. 1996.

[86] M. Kiermaier, "Geometric constructions of linear codes over Galois rings of characteristic 4 of high homogeneous minimum distance," Ph.D. dissertation, Universität Bayreuth, 2012.

[87] L. E. Fuller, "A canonical set for matrices over a principal ideal ring modulo m," *Canad. J. Math.*, pp. 54–59, 1955.

[88] G. Birkhoff, "Subgroups of abelian groups," *Proc. London Math. Soc.*, pp. 385–401, 1934.

[89] B. R. McDonald, "Enumeration of classes of row equivalent matrices over a principal ideal domain modulo p^n," *Duke Math. J.*, vol. 37, no. 1, pp. 163–169, 1970.

[90] V. Naware, G. Mergen, and L. Tong, "Stability and delay of finite-user slotted ALOHA with multipacket reception," *IEEE Trans. Inf. Theory*, vol. 51, no. 7, pp. 2636–2656, Jul. 2005.

[91] A. Vem, Y.-C. Huang, K. R. Narayanan, and H. D. Pfister, "Multilevel lattices based on spatially-coupled ldpc codes with applications," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Honolulu, HI, Jun. 29– Jul. 4, 2014, pp. 2336–2340.

[92] Y.-C. Huang, K. R. Narayanan, and N. E. Tunali, "Multistage compute-and-forward with multilevel lattice codes based on product constructions," *Computing Research Repository (CoRR)*, Jan. 2014. [Online]. Available: http://arxiv.org/abs/1401.2228

[93] J. Zhan, B. Nazer, U. Erez, and M. Gastpar, "Integer-forcing linear receivers," in *Proc. of IEEE Int. Symp. on Inf. Theory*, Austin, Texas, USA, Jun. 13–18, 2010, pp. 1022–1026.

[94] K. R. Narayanan and H. D. Pfister, "Iterative collision resolution for slotted aloha: An optimal uncoordinated transmission policy," in *Int. Symp. Turbo Codes and Iter. Inf. Processing*, Gothenburg, Sweden, Aug. 27–31, 2012, pp. 136–139.

[95] C. Stefanovic and P. Popovski, "Aloha random access that operates as a rateless code," *IEEE Trans. Commun.*, vol. 61, no. 11, pp. 4653–4662, Nov. 2013.

[96] E. Paolini, G. Liva, and M. Chiani, "Coded slotted aloha: A graph-based method for uncoordinated multiple access," *Computing Research Repository (CoRR)*, Jan. 2014. [Online]. Available: http://arxiv.org/abs/1401.1626

[97] C. Bordenave, D. McDonald, and A. Proutiere, "Asymptotic stability region of slotted aloha," *IEEE Trans. Inf. Theory*, vol. 58, no. 9, pp. 5841–5855, Sep. 2012.