Error Control for Network Coding

by

Danilo Silva

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

Error Control for Network Coding

Danilo Silva

Doctor of Philosophy

Graduate Department of Electrical and Computer Engineering

University of Toronto

2009

Network coding has emerged as a new paradigm for communication in networks, allowing packets to be algebraically combined at internal nodes, rather than simply routed or replicated. The very nature of packet-mixing, however, makes the system highly sensitive to error propagation. Classical error correction approaches are therefore insufficient to solve the problem, which calls for novel techniques and insights.

The main portion of this work is devoted to the problem of error control assuming an adversarial or worst-case error model. We start by proposing a general coding theory for adversarial channels, whose aim is to characterize the correction capability of a code. We then specialize this theory to the cases of coherent and noncoherent network coding. For coherent network coding, we show that the correction capability is given by the rank metric, while for noncoherent network coding, it is given by a new metric, called the injection metric. For both cases, optimal or near-optimal coding schemes are proposed based on rank-metric codes. In addition, we show how existing decoding algorithms for rank-metric codes can be conveniently adapted to work over a network coding channel. We also present several speed improvements that make these algorithms the fastest known to date.

The second part of this work investigates a probabilistic error model. Upper and lower bounds on capacity are obtained for any channel parameters, and asymptotic expressions are provided in the limit of long packet length and/or large field size. A simple coding

scheme is presented that achieves capacity in both limiting cases. The scheme has fairly low decoding complexity and a probability of failure that decreases exponentially both in the packet length and in the field size in bits. Extensions of the scheme are provided for several variations of the channel.

A final contribution of this work is to apply rank-metric codes to a closely related problem: securing a network coding system against an eavesdropper. We show that the maximum possible rate can be achieved with a coset coding scheme based on rank-metric codes. Unlike previous schemes, our scheme has the distinctive property of being universal: it can be applied on top of any communication network without requiring knowledge of or any modifications on the underlying network code. In addition, the scheme can be easily combined with a rank-metric-based error control scheme to provide both security and reliability.

To Amanda

Acknowledgements

I am most grateful to Prof. Frank Kschischang for being no less than the best supervisor I could possibly imagine. In fact, Frank has continually surprised me by exceeding all my expectations. He is not just a brilliant researcher and a wonderful person. He is a real coach: encouraging, supportive, understanding, honest, and kind. He not only was constantly helping me identify and overcome my weaknesses, but seemed to always have a well-thought-out plan to develop my potential. He has also been very generous in giving advice—about life, the academia and everything—which has been priceless to me. Moreover, he is a scientist in the strongest sense of the word. I truly admire his focus on fundamental questions, his interest in practically relevant problems, his attention to textual clarity and logical precision, and his taste for beautiful mathematics, all of which have exerted an invaluable influence on me. To top it off, Frank is an excellent teacher and communicator, and makes every effort to pass on his abilities to his students. His wisdom and his dedication in giving careful feedback have greatly benefited me (and, I hope, the audience of my talks). In summary, Frank is not just a brilliant researcher, he also cares a lot about his students. I will always be grateful to him for making my graduate experience so significant, rewarding, and fun.

I am also grateful to Prof. Ralf Kötter for his generosity, patience and confidence in collaborating with me. Ralf was the pioneer of this work and a constant source of inspiration to me. His ingenious insights and invaluable suggestions greatly contributed to this work.

I wish to thank the members of the examination committee, Prof. Baochun Li, Prof. Ben Liang, Prof. Wei Yu, and Dr. Emina Soljanin from Alcatel-Lucent, for their useful suggestions to improve this thesis.

I would like to thank the University of Toronto for their excellent resources and work environment and their friendly and helpful staff. In addition, I would like to thank the CAPES Foundation, Brazil, for their generous financial support.

Thank you to all my friends and colleagues from the communications group: Ben Smith, Hayssam Dahrouj, Nevena Lazic, Jim Huang, Azadeh Khaleghi, Da Wang, Weifei Zeng, Chen Feng, and Siyu Liu; as well as my Brazilian friends who are also in the academia: Tiago Falk, Eric Bouton, Marcos Vasconcelos, and Tiago Vinhoza.

Last but not least, I would like to thank my family, for their love and support, and especially my sweet and loving wife Amanda, for placing our relationship above anything else, and for being so strong, understanding and supportive.

Contents

Chapter 1

Introduction

1.1 Network Coding

The traditional way of operating communication networks is to treat data as a commodity. Packets originating at a source node are routed through the network until they reach a destination; in this process, each packet is kept essentially intact. The underlying assumption is that the very packets that are produced at the source node must be *delivered* to a destination node. Indeed, the theory and practice of network communications has evolved together with the theory of commodity flows in networks. As an analogy, we may think of the commodity flow problem as that of cars traveling on interconnected roads. It is evident that, if a car is to travel to another city, all its parts (either together or disassembled) must be physically transported.

This assumption that information flows could be treated as commodity flows remained unquestioned for several decades, until very recently. The revolutionary insight of *network coding* is that *bits are not like cars* [1]. Bits can be *coded*, i.e., undergo mathematical operations, something no physical commodity can. The analogy breaks down because, in contrast to the parts of a car, there is no need to deliver to a destination the *actual* packets produced by the source node: instead, mere *evidence* about these packets suffices. In this

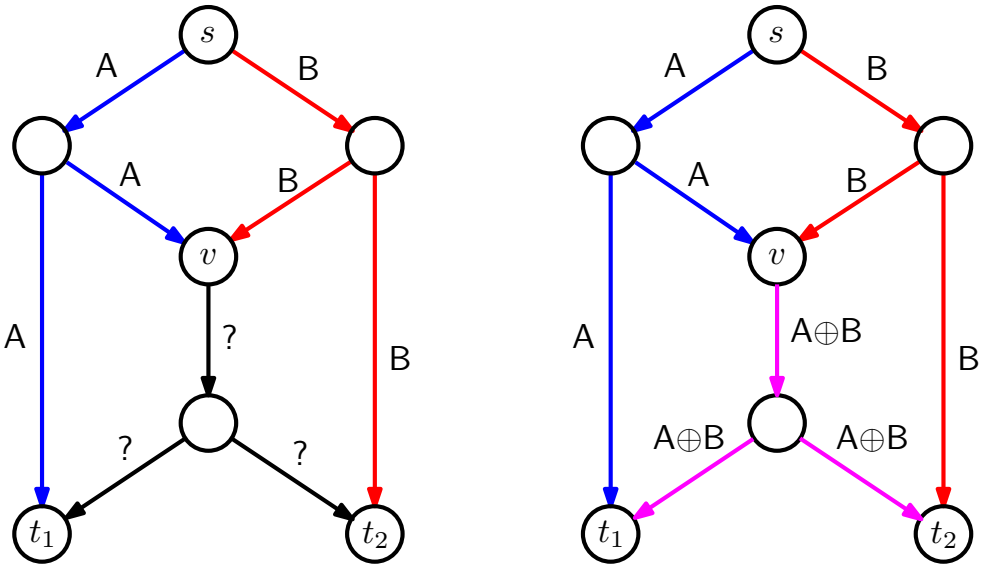

Figure 1.1: The butterfly network. (a) With routing alone, there is a bottleneck at node v, as only one packet can be transmitted at a time. (b) Network coding.

context, coding is essentially a way of producing evidence, transforming a transportation problem into an inference problem. What does this insight buy us? The groundbreaking result of [2] is that, in many useful cases, network coding allows communication at a *higher rate* than when no coding is allowed.

The simplest example that can illustrate the point has come to be known as the *butterfly network* [2], depicted in Fig. 1.1. There is a single source s, which produces bits A and B, and two destinations t_1 and t_2, both of which request both bits A and B. As the same data must be transmitted to multiple destinations, this is a problem known as that of *multicasting* information. Each link in the network has capacity to transmit a single bit at a time. If no coding is allowed, then, as we can see from Fig. 1.1a, there will be a "bottleneck" (congestion) at node v, as only one packet of either A or B can be transmitted through its outgoing link ℓ. Packets would then be placed in a queue, to be transmitted sequentially as new opportunities arise. In order for both destinations to recover both packets, two transmissions over link ℓ are required.

On the other hand, if coding is allowed, then node v can simply transmit the XOR of bits A and B, as illustrated in Fig. 1.1b. This allows destination t_1 to recover $B = A \oplus (A \oplus B)$ and destination t_2 to recover $A = B \oplus (A \oplus B)$. Thus, by overcoming congestion, network coding allows to increase the throughput of a communication network. More precisely stated, to achieve the capacity of a multicast network in general, it is strictly necessary to use network coding.

In contrast to the simple example of Fig. 1.1, real networks can be hugely complex. If network coding is to be widely used, efficient coding (and decoding) operations must be devised so that the benefits of network coding can be obtained without increasing the implementation costs to a prohibitive level. In this context, two contributions can be said to have raised network coding from a theoretical curiosity to a potential engineering application: these are *linear network coding* [3,4] and *random linear network coding* [5,6]. With linear network coding, all coding operations at nodes are constrained to be linear combinations of packets over a finite field. The importance of linear operations is that they are arguably the simplest possible to perform in practice, and the fundamental contribution of [3] is to show that no loss in multicast capacity is incurred if the field size is sufficiently large. What exactly to choose as the coefficients of these linear combinations, however, remains a problem. Must the network code be designed by a central authority and informed individually to each node before transmission? The contribution of [6] shows that a distributed design is sufficient in most cases. More precisely, if nodes choose coefficients uniformly at random and independently from each other, then no capacity is lost with high probability if the field size is sufficiently large. The actual coefficients used can be easily recorded in the packet headers. As decoding corresponds to performing the well-known algorithm of Gaussian elimination, random linear network coding becomes a practical approach that can potentially be implemented over virtually any network.

An important benefit of random network coding is that, even in the cases where network coding does not increase the throughput, it can greatly simplify the network

operation. The idea is that all "intelligence" is removed from the network (in the form of scheduling and other algorithms) and is instead placed at its endpoints. The internal nodes are as "dumb" as possible, simply performing random linear combinations of packets, with little regard to the conditions of farther nodes. One could argue that this principle is consistent, for instance, with the design principle of the Internet [7], and therefore represents an unavoidable trend.

Indeed, since its original publication in 2000, network coding has quickly emerged into a major research area. The quickly growing list of contributions (see [8]) ranges from theoretical inquiries on fundamental limits to real-world practical applications, and provides sound evidence that network coding may indeed spur a revolution in network communications in the near future.

1.2 Error Control

However elegant and compelling, the principle of network coding is not without its drawbacks. Network coding achieves its benefits, essentially, by making every packet in the network be statistically dependent on (almost) every other packet. However, this dependence creates a problem. What if some of the packets are corrupted? As Fig. 1.2 illustrates, corrupt packets may contaminate other packets when coded at the internal nodes, leading to an error propagation problem. Indeed, even a single corrupt packet has the potential, when linearly combined with legitimate packets, to affect all packets gathered by a destination node. This is in contrast to routing (no coding), where an error in one packet affects only one source-destination path.

Thus, the phenomenon of error propagation in network coding can overwhelm the error correction capability of any classical error-correcting code used to protect the data end-to-end. The word "classical" here means designed for the Hamming metric. In short, we might say that the Hamming metric is not well-suited to the end-to-end channel

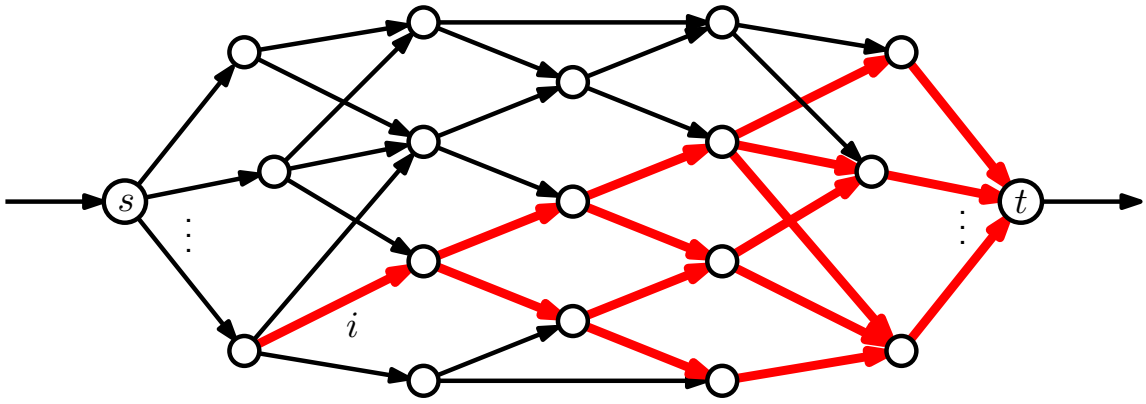

Figure 1.2: The phenomenon of error propagation induced by network coding. An error occurred at the link i, which then propagated to the rightmost links after packet mixing.

induced by network coding. Thus, novel coding techniques are needed to solve this new error correction problem. As we shall see, network coding requires error-correcting codes designed for a different metric.

At this point, one might wonder whether considering this problem is indeed realistic. Aren't corrupt packets detected and immediately rejected by the lower network layers—leading to an *erasure* rather than an undetected error? First, it must be noticed that the physical-layer code may not always be perfect. Very reliable physical-layer codes require very long block lengths, and there are several situations where a very short block length is required for practical reasons (e.g., [9]). Thus, we must be able to cope with eventual packet errors, without having to drop (and request retransmission of) all the received packets.

Another compelling source of errors might be the presence of an *adversary*, i.e., a user who does not comply with the protocol. Since the adversary injects corrupt packets at the application layer, his effects cannot possibly be detected at the physical layer. Adversarial errors represent an even more severe threat than random errors, since they can be specifically designed to defeat the end-to-end error-correcting code.

Research into error-correcting codes for network coding started with [10–14], which

investigated fundamental limits for adversarial error correction under a deterministic (i.e., non-random) network coding setting. Code constructions for random network coding were later proposed in [15, 16]; these constructions rely on using arbitrarily large packet length and field size. In contrast, the framework proposed in [17, 18] is valid for any field or packet size. It can be said that their main philosophical contribution is to "take the network out of the problem:" given any parameters of a random network coding system, an end-to-end error-correcting scheme can be constructed providing any specified error-correcting capability. Thus, the whole network can be seen simply as a (different kind of) channel—in fact, a noncoherent multiple-input multiple-output finite field channel.

The work in [18] has been the main motivation for this thesis. Some of the questions left open in [18], and which we tackle in this thesis, are the following. What are the fundamental limits of this end-to-end approach? Can the approach be extended to the deterministic setting of [11, 12]? And how can we design codes that simultaneously exhibit good performance and yet admit computationally efficient encoding and decoding algorithms? As we shall see, designing such efficient encoders and decoders is a vast research problem on its own.

1.3 Information-Theoretic Security

Another issue brought up by the use of network coding lies in the field of information-theoretic security. The issue has to do with the fact that the linear scrambling of packets induced by network coding may actually be helpful to an eventual eavesdropper.

Consider the problem of securely transmitting a message over a network subject to the presence of a wiretapper. More precisely, assume that the network supports the reliable transmission of n packets from source to destination, and that the wiretapper can intercept only $\mu < n$ packets (anywhere in the network). Suppose that the network uses routing only. This problem is an instance of the wiretap channel II of Ozarow and

Wyner [19]. It can be shown that the maximum rate that can be achieved, such that the wiretapper obtains *no* information about the message, is precisely $n - \mu$ packets per network use. Moreover, a simple coding scheme achieving this rate can be obtained as follows. First, generate μ packets uniformly at random; then, apply a (carefully designed) invertible linear transformation to the collection of these μ random packets together with the $n - \mu$ message packets, in order to produce the n packets that should be transmitted. While the destination can easily invert the linear transformation and obtain the message, the fact that "noise" was added to the transmission has the effect of completely confusing the wiretapper. This happens because the linear transformation has been carefully designed to produce a perfect mixing of message and noise.

Now, consider the case where the network uses linear network coding. It may be the case that the linear scrambling performed by network coding actually "unscrambles" part of the linear transformation applied at the source. In other words, network coding may disrupt the perfect mixing of message and noise, and help the wiretapper obtain non-negligible information about the message. Thus, network coding may significantly impair the use of information-theoretic security.

In order to reconcile network coding and information-theoretic security, several works have proposed a joint design of the network code and the outer linear transformation. In [20, 21], security was achieved by carefully designing the linear transformation taking into account a given network coding. In contrast, [22] proposed to design the network code based on a specific linear transformation (one suitable for the routing-only problem). A difficulty in all of these works is that they require a significantly large field size in order for such a joint design to be possible. In other words, it is not possible under any of these approaches to guarantee perfect information-theoretic security for a large network that uses a small field size. In this spirit of the end-to-end approach of [18], we may formulate the following question: is it possible to design an encoding that is *universally* secure, i.e., regardless of the network code and the field size?

At a first glance, the problem of error-control coding and information-theoretic security seem barely related. Our motivation for addressing the latter problem is that the techniques we propose for error correction are actually very helpful to solve the security problem. We shall see that end-to-end codes designed for adversarial error correction turn out to be closely related to linear transformations that are universally secure. Not only can universal security be achieved: it can be implemented in an efficiently manner using techniques similar to those for error control coding.

A final question that is natural at this point: is it possible to simultaneously provide error control and information-theoretic security in the same random linear network coding system?

As we see in Chapter 8, an affirmative answer can be obtained by appropriately concatenating (layering) our proposed error control and security schemes; in particular, no joint design is required, but simply a common interface.

1.4 Contributions

This thesis aims at providing a theoretical but practically-oriented solution to the problems of error control and security in network coding. The solutions we propose are neither unique nor optimal, but lie in an (arguably) compelling point of the "theoretical performance" versus "practical feasibility" tradeoff.

Our results can be divided into three main areas: correction of adversarial errors; correction of random errors; and security against a wiretapper.

Our results for random error correction are the easiest to explain. This is because a linear network coding channel that admits a probabilistic model can be understood and analyzed using principles from information theory. Given such a probabilistic model, we can define notions of channel capacity and capacity-achieving codes, as well as examine the tradeoff between performance and delay. Our results in this area are the following:

- We compute upper and lower bounds on the capacity of the random linear network coding channel under a probabilistic error model. These bounds match (giving the channel capacity) when either the field size or the packet length are large.
- We present a simple coding scheme that asymptotically achieves the channel capacity when either the field size or the packet length grows.
- For finite channel parameters, our coding scheme has simultaneously a better performance and lower complexity than previous schemes [23].
- We present extensions of our results for several variations of the channel.

In contrast to information theory, a unified framework is unavailable for treating adversarial channels. The kind of adversarial channel discussed in this thesis (where the adversary can inject at most t error packets) is most closely analogous to a binary vector channel where an adversary may arbitrarily flip at most t bits—a channel for which classical coding theory provides a complete and elegant solution. However, classical coding theory alone is not enough to provide guidance for network error correction. For instance, classical coding theory often requires a distance metric relating channel input and output words; however, for the network coding channel, the input and output alphabets may not even be the same (thus making it impossible to define any such metric).

To fill this gap, we propose a coding theory for adversarial channels that is suitable for the channels at hand. Such a theory is then specialized to coherent and noncoherent network coding—the distinction is in whether or not the receiver has information about the channel state. As in classical coding theory, the core notion of our theory is a "distance function" that precisely describes the error correction capability of a code. The close relationship between such distance functions (for both coherent and noncoherent network coding) and the *rank metric* allows us to construct optimal or near-optimal codes based on rank-metric codes. Moreover, decoding of our codes is shown to be

closely related to the decoding of rank-metric codes.

Our results for adversarial error correction comprise the main part of this thesis and can be summarized as follows:

- We define the concept of Δ-distance and compute this distance for both coherent and noncoherent network coding. We then prove an "if and only if" statement relating the minimum Δ-distance of a code and its error correction capability. A consequence of this result is that—just as with classical coding theory—one can ignore the channel model and focus solely on the combinatorial problem of finding the largest code with a given minimum Δ-distance.
- We show that a class of rank-metric codes called maximum-rank-distance (MRD) codes are optimal for coherent network coding and can be slightly modified to provide near-optimal codes for noncoherent network coding.
- We show that the decoding problems for both coherent and noncoherent network coding are mathematically equivalent and can be reformulated as a generalized decoding problem for rank-metric codes.
- We propose two algorithms for generalized rank-metric decoding: one is based on a well-known "time-domain" algorithm and the other is based on a novel transform-domain approach. We also propose two encoding algorithms, one for high-rate systematic codes and another for nonsystematic codes.
- The encoding and decoding algorithms we propose make use of optimal (or low-complexity) normal bases to improve their speed. We show that these algorithms are faster than any previous algorithms.

With respect to information-theoretic security, our framework follows closely that of [22] (which in turn follows [19]). The novelty lies in the fact that we investigate more general transformations to be applied at the source. More precisely, we show how one can exploit the properties of rank-metric codes to achieve universal security. This is all

developed in the succinct and elegant algebraic language of finite field extensions, which proves useful to investigate variations and generalizations of the problem. Our results in this area are as follows:

- We propose a MRD-based coding scheme that can provide universal security. We show that the scheme is optimal in the sense of requiring the smallest packet length possible for universal security.
- We investigate the notion of weak security proposed in [24], as well as extensions and variations of the problem, and show that *universal* weak security amounts to the existence of matrices with certain properties. We then show that such matrices always exist if the packet length (not the field size) is sufficiently large. In some special cases we present coding schemes that are optimal in terms of packet length.
- We address the case where wiretapper is also a jammer (who can introduce error packets) and show that universal strong/weak security and error control can be simultaneously achieved using a simple tandem scheme based on MRD codes.

The results in this thesis originated the following publications: [25–30] for adversarial error correction, [31,32] for random error correction, and [33–35] for information-theoretic security.

1.5 Outline

The remainder of the thesis is organized as follows. Chapter 2 reviews mathematical preliminaries on linear algebra and finite field theory, as well as the theory of rank-metric codes. Chapter 3 presents the basic model of a linear network coding channel subject to errors. Chapters 4, 5 and 6 treat adversarial error correction, Chapter 7 treats random error correction, and Chapter 8 treats information-theoretic security. Chapter 9 provides our conclusions and suggestions for future work.

Due to the variety of subjects covered, we anticipate that this thesis may be read by readers with distinct interests and backgrounds. To their benefit, this thesis can be read in a modular fashion after Chapters 2 and 3. Readers who are interested only in fundamental limits (i.e., capacity results) for network coding may read only Chapters 4, 7 and 8. Readers who are interested only in results for rank-metric codes may read only Chapter 6 (possibly also Section 5.2). Readers who are only interested in error control aspects of network coding may read Chapters 4, 5, 6 and 7 and skip Chapter 8, while readers who are only interested in security issues may read only Chapter 8.

Chapter 2

Preliminaries

This chapter reviews necessary mathematical background and establishes the notation used in this thesis. Sections 2.1 and 2.2 review linear algebra and finite-field algebra. Section 2.3 reviews the basic theory of rank-metric codes. Section 2.4 reviews linearized polynomials, which are used in Chapter 6 in the decoding of rank-metric codes. Section 2.5 discusses the complexity of operations in normal bases; this section may safely be skipped by readers who are not interested in implementation details.

We start with some miscellaneous notation. Let $\mathbb{N} = \{0, 1, 2, \ldots\}$. Define $[x]^+ = \max\{x, 0\}$. Let $\mathbf{1}[P]$ represent the Iverson notation for a proposition P, i.e., $\mathbf{1}[P]$ is equal to 1 if P is true and is equal to 0 otherwise.

2.1 Matrices and Subspaces

Let $q \geq 2$ be a power of a prime, and let $\mathbb{F}_q$ denote the finite field with q elements. Let $\mathbb{F}_q^{n\times m}$ denote the set of all $n \times m$ matrices over $\mathbb{F}_q$, and set $\mathbb{F}_q^n = \mathbb{F}_q^{n\times 1}$. In particular, $v \in \mathbb{F}_q^n$ is a column vector and $v \in \mathbb{F}_q^{1\times m}$ is a row vector.

If v is a vector, then the symbol v_i denotes the ith entry of v. If A is a matrix, then the symbol A_i may denote either the ith row or the ith column of A; the distinction will always be clear from the way in which A is defined. In either case, the symbol A_{ij} always

refers to the entry in the ith row and jth column of A.

For clarity, the $n \times m$ all-zero matrix and the $n \times n$ identity matrix are denoted by $0_{n\times m}$ and $I_{n\times n}$, respectively, where the subscripts may be omitted when there is no risk of confusion. If $I = I_{n\times n}$, then our convention is that I_i denotes the ith *column* of I. More generally, if $\mathcal{U} \subseteq \{1, \ldots, n\}$, then $I_{\mathcal{U}} = [I_i,\, i \in \mathcal{U}]$ denotes the sub-matrix of I consisting of the columns indexed by $\mathcal{U}$.

The row (column) rank of a matrix $A \in \mathbb{F}_q^{n\times m}$ is the maximum number of rows (columns) of A that are linearly independent. The row rank and the column rank are always equal and are simply called the rank of A, denoted by $\mathsf{rank}\, A$. If $\mathsf{rank}\, A = n$ ($\mathsf{rank}\, A = m$), then A is said to have full row rank (column rank); otherwise, it is said to have a row-rank (column-rank) deficiency of $n - \mathsf{rank}\, A$ ($m - \mathsf{rank}\, A$). The maximum possible rank for A is $\min\{n, m\}$, in which case A is said to be a full-rank matrix.

Let $\mathsf{wt}(X)$ denote the number of nonzero rows of a matrix X. Clearly, $\mathsf{rank}\, X \leq \mathsf{wt}(X)$.

Let $\mathcal{T}_{n\times m,t}(\mathbb{F}_q)$ denote the set of all $n \times m$ matrices of rank t over $\mathbb{F}_q$. We shall write simply $\mathcal{T}_{n\times m,t} = \mathcal{T}_{n\times m,t}(\mathbb{F}_q)$ when the field $\mathbb{F}_q$ is clear from the context. We also use the notation $\mathcal{T}_{n\times m} = \mathcal{T}_{n\times m,\min\{n,m\}}$ for the set of all full-rank $n \times m$ matrices.

The rank of a matrix $X \in \mathbb{F}_q^{n\times m}$ is the smallest r for which there exist matrices $P \in \mathbb{F}_q^{n\times r}$ and $Q \in \mathbb{F}_q^{r\times m}$ such that $X = PQ$, i.e.,

$$\mathsf{rank}\, X = \min_{\substack{r\in\mathbb{N}, P\in\mathbb{F}_q^{n\times r}, Q\in\mathbb{F}_q^{r\times m}: \\ X=PQ}} r. \tag{2.1}$$

Note that both matrices obtained in the decomposition are full-rank; accordingly, such a decomposition is called a full-rank decomposition [36]. In this case, note that, by partitioning P and Q, the matrix X can be further expanded as

$$M = PQ = \begin{bmatrix} P' & P'' \end{bmatrix} \begin{bmatrix} Q' \\ Q'' \end{bmatrix} = P'Q' + P''Q''$$

where $\mathsf{rank}\,(P'Q') + \mathsf{rank}\,(P''Q'') = r$. In particular, X can be expanded as a sum of outer products

$$X = PQ = \begin{bmatrix} P_1 & \cdots & P_r \end{bmatrix} \begin{bmatrix} Q_1 \\ \vdots \\ Q_r \end{bmatrix} = \sum_{i=1}^{r} P_i Q_i$$

where $P_1, \ldots, P_r \in \mathbb{F}_q^{n\times 1}$ and $Q_1, \ldots, Q_r \in \mathbb{F}_q^{1\times m}$.

Two other useful properties of the rank function are given below [36]: for any $X, Y \in \mathbb{F}_q^{n\times m}$, we have

$$\mathsf{rank}\,(X+Y) \leq \mathsf{rank}\; X + \mathsf{rank}\; Y \tag{2.4}$$

and, for $X \in \mathbb{F}_q^{n\times m}$ and $A \in \mathbb{F}_q^{N\times n}$, we have

$$\mathsf{rank}\; A + \mathsf{rank}\; X - n \leq \mathsf{rank}\; AX \leq \min\{\mathsf{rank}\; A, \mathsf{rank}\; X\}. \tag{2.5}$$

Let $\mathsf{dim}\; \mathcal{V}$ denote the dimension of a vector space $\mathcal{V}$. Let $\langle v_1, \ldots, v_k \rangle$ denote the linear span of a set of vectors $v_1, \ldots, v_k$, and let $\langle X \rangle$ denote the row space of a matrix X. Recall that $\mathbb{F}_q^{1\times m}$ (or $\mathbb{F}_q^m$) is an m-dimensional vector space over $\mathbb{F}_q$. The row space of $X \in \mathbb{F}_q^{n\times m}$ is a subspace of $\mathbb{F}_q^{1\times m}$. Moreover, $\mathsf{dim}\; \langle X \rangle = \mathsf{rank}\; X$.

Let $\mathcal{U}$ and $\mathcal{V}$ be subspaces of some fixed vector space. Recall that the sum

$$\mathcal{U} + \mathcal{V} = \{u + v \colon u \in \mathcal{U},\, v \in \mathcal{V}\}$$

is the smallest vector space that contains both $\mathcal{U}$ and $\mathcal{V}$. The intersection $\mathcal{U} \cap \mathcal{V}$ is the largest vector space that is contained in both $\mathcal{U}$ and $\mathcal{V}$. Recall also that

$$\mathsf{dim}\,(\mathcal{U} + \mathcal{V}) = \mathsf{dim}\; \mathcal{U} + \mathsf{dim}\; \mathcal{V} - \mathsf{dim}\,(\mathcal{U} \cap \mathcal{V}). \tag{2.7}$$

If $X \in \mathbb{F}_q^{n\times m}$ and $Y \in \mathbb{F}_q^{N\times m}$ are matrices, then a very useful fact about their row spaces is that

$$\left\langle \begin{bmatrix} X \\ Y \end{bmatrix} \right\rangle = \langle X \rangle + \langle Y \rangle\,. \tag{2.8}$$

Therefore,

$$\begin{aligned}\mathsf{rank}\begin{bmatrix}X\\Y\end{bmatrix} &= \mathsf{dim}\left(\langle X\rangle + \langle Y\rangle\right)\\ &= \mathsf{rank}\ X + \mathsf{rank}\ Y - \mathsf{dim}\left(\langle X\rangle \cap \langle Y\rangle\right).\end{aligned} \tag{2.9}$$

Let $\mathcal{P}(\mathbb{F}_q^m)$ denote the set of all subspaces of $\mathbb{F}_q^m$. Assume $n \leq m$. The *Grassmannian* $\mathcal{P}_n(\mathbb{F}_q^m)$ is the set of all n-dimensional subspaces of $\mathbb{F}_q^m$. Define

$$\mathcal{P}_n^{\max}(\mathbb{F}_q^m) = \bigcup_{k=0}^{n} \mathcal{P}_k(\mathbb{F}_q^m)$$

as the set of all subspaces of $\mathbb{F}_q^m$ with dimension up to n. Let $\mathsf{RRE}\left(X\right)$ denote the reduced row echelon (RRE) form of a matrix X. Note that there exists a bijection between $\mathcal{P}_n^{\max}(\mathbb{F}_q^m)$ and the subset of $\mathbb{F}_q^{n\times m}$ consisting of matrices in RRE form. In other words, every subspace $\mathcal{V} \in \mathcal{P}_n^{\max}(\mathbb{F}_q^m)$ can be uniquely represented by a matrix $X \in \mathbb{F}_q^{n\times m}$ in RRE form such that $\mathcal{V} = \langle X\rangle$. In particular, every subspace in $\mathcal{P}_n(\mathbb{F}_q^m)$ is associated with a matrix in $\mathcal{T}_{n\times m}$ in RRE form.

The size of the Grassmannian $\mathcal{P}_n(\mathbb{F}_q^m)$ is given by the *Gaussian coefficient*

$$\begin{bmatrix}m\\n\end{bmatrix}_q = \prod_{i=0}^{n-1} \frac{(q^m - q^i)}{(q^n - q^i)}.$$

Two useful properties of the Gaussian coefficient are [18, Lemma 5]

$$q^{n(m-n)} < \begin{bmatrix}m\\n\end{bmatrix}_q < 4q^{n(m-n)} \tag{2.12}$$

and [37]

$$\begin{bmatrix}m\\n\end{bmatrix}_q \begin{bmatrix}n\\t\end{bmatrix}_q = \begin{bmatrix}m\\t\end{bmatrix}_q \begin{bmatrix}m-t\\n-t\end{bmatrix}_q, \quad t \leq n \leq m. \tag{2.13}$$

The Gaussian coefficient also arises when computing the size of $\mathcal{T}_{n\times m,t}$. It can be

shown that the number of $n \times m$ matrices of rank t is given by [38, p. 455]

$$|\mathcal{T}_{n\times m,t}| = \frac{|\mathcal{T}_{n\times t}||\mathcal{T}_{t\times m}|}{|\mathcal{T}_{t\times t}|} = |\mathcal{T}_{n\times t}| \begin{bmatrix} m \\ t \end{bmatrix}_q \tag{2.14}$$

$$= q^{(n+m-t)t} \prod_{i=0}^{t-1} \frac{(1-q^{i-n})(1-q^{i-m})}{(1-q^{i-t})}. \tag{2.15}$$

A method of computing the RRE form of a matrix is the well-known Gaussian (or rather, Gauss-Jordan) elimination [39]. It is a straightforward exercise to show that the number of operations needed to convert an $n \times m$ matrix of rank r to RRE form (assuming $n \leq m$) is no more that r divisions, $\frac{1}{2}rn(2m-r-1)$ multiplications, and $\frac{1}{2}r(n-1)(2m-r-1)$ additions in $\mathbb{F}_q$.

2.2 Bases over Finite Fields

For convenience, when dealing with bases over finite fields, we assume that the entries of all bases, vectors and matrices are indexed starting from 0.

Let $\mathcal{V}$ be n-dimensional vector space over $\mathbb{F}_q$ with an ordered basis $\mathcal{A} = \{\alpha_0, \ldots, \alpha_{n-1}\}$. For $v \in \mathcal{V}$, we denote by $\left[v\right]_{\mathcal{A}}$ the coordinate vector of v relative to $\mathcal{A}$; that is, $\left[v\right]_{\mathcal{A}} = \begin{bmatrix} v_0 & \cdots & v_{n-1} \end{bmatrix}$, where $v_0, \ldots, v_{n-1}$ are the unique elements in $\mathbb{F}_q$ such that $v = \sum_{i=0}^{n-1} v_i \alpha_i$. When the basis $\mathcal{A}$ is clear from context, we shall use the simplified notation $\underline{v} \triangleq \left[v\right]_{\mathcal{A}}$.

Let $\mathcal{W}$ be an m-dimensional vector space over $\mathbb{F}_q$ with ordered basis $\mathcal{B} = \{\beta_0, \ldots, \beta_{m-1}\}$, and let T be a linear transformation from $\mathcal{V}$ to $\mathcal{W}$. We denote by $\left[T\right]_{\mathcal{A}}^{\mathcal{B}}$ the matrix representation of T in the bases $\mathcal{A}$ and $\mathcal{B}$; that is, $\left[T\right]_{\mathcal{A}}^{\mathcal{B}}$ is the unique $n \times m$ matrix over $\mathbb{F}_q$ such that

$$T(\alpha_i) = \sum_{j=0}^{m-1} \left(\left[T\right]_{\mathcal{A}}^{\mathcal{B}} \right)_{ij} \beta_j, \quad i = 0, \ldots, n-1. \tag{2.16}$$

With these notations, we have, for $v \in \mathcal{V}$,

$$\Big[T(v)\Big]_{\mathcal{B}} = \Big[v\Big]_{\mathcal{A}} \Big[T\Big]_{\mathcal{A}}^{\mathcal{B}}.$$

Let $\mathcal{U}$ be a k-dimensional vector space over $\mathbb{F}_q$ with ordered basis $\Theta = \{\theta_0, \ldots, \theta_{k-1}\}$, and let S be a linear transformation from $\mathcal{W}$ to $\mathcal{U}$. Recall that [39]

$$\Big[S \circ T\Big]_{\mathcal{A}}^{\Theta} = \Big[T\Big]_{\mathcal{A}}^{\mathcal{B}} \Big[S\Big]_{\mathcal{B}}^{\Theta}. \tag{2.18}$$

Let $\mathbb{F}_{q^m}$ be an extension field of $\mathbb{F}_q$. Recall that every extension field can be regarded as a vector space over the base field. Let $\mathcal{A} = \{\alpha_0, \ldots, \alpha_{m-1}\}$ be a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. If $\mathcal{A}$ is of the form $\mathcal{A} = \{\alpha^0, \alpha^1, \ldots, \alpha^{m-1}\}$, then $\mathcal{A}$ is called a *polynomial* (or *standard*) *basis*. If $\mathcal{A}$ is of the form $\mathcal{A} = \left\{\alpha^{q^0}, \alpha^{q^1}, \ldots, \alpha^{q^{m-1}}\right\}$, then $\mathcal{A}$ is called a *normal basis*, and α is called a *normal element* [38].

Every basis over a finite field admits a *dual basis*. The dual basis of $\mathcal{A}$ is the unique basis $\mathcal{A}' = \{\alpha'_0, \ldots, \alpha'_{m-1}\}$ such that

$$\mathsf{Tr}(\alpha_i \alpha'_j) = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

where $\mathsf{Tr}(\beta) = \sum_{k=0}^{m-1} \beta^{q^k}$ denotes the *trace* of an element $\beta \in \mathbb{F}_{q^m}$ [38]. Note that $\beta \mapsto \beta^q$ is an $\mathbb{F}_q$-linear operator. It follows that if $[\beta]_{\mathcal{A}} = \Big[\beta_0 \quad \cdots \quad \beta_{n-1}\Big]$, then $\beta_j = \mathsf{Tr}(\beta\alpha'_j)$, for all j. An important property of the trace is that it always returns elements in the base field, i.e., $\mathsf{Tr}(\beta) \in \mathbb{F}_q$, for all $\beta \in \mathbb{F}_{q^m}$. Moreover,

$$\mathsf{Tr}(\beta^q) = \mathsf{Tr}(\beta), \quad \text{for all } \beta \in \mathbb{F}_{q^m} \tag{2.20}$$

due to the fact that $\beta^{q^m} = \beta$ in $\mathbb{F}_{q^m}$.

A basis is *self-dual* if it is equal (in the same order) to its dual basis.

2.3 Rank-Metric Codes

Let $X, Y \in \mathbb{F}_q^{n\times m}$ be matrices. The *rank distance* between X and Y is defined as

$$d_{\mathrm{R}}(X, Y) \triangleq \mathsf{rank}\,(Y - X).$$

As observed in [37, 40], the rank distance is indeed a *metric*. In particular, the triangle inequality follows directly from (2.4). Thus, $\mathbb{F}_q^{n\times m}$ is a metric space.

A *rank-metric code* $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ is a matrix code (i.e., a nonempty set of matrices) used in the context of the rank metric. The *minimum rank distance* of $\mathcal{C}$, denoted $d_{\mathrm{R}}(\mathcal{C})$, is the minimum rank distance between all pairs of distinct codewords of $\mathcal{C}$.

Rank-metric codes are typically used as error-correcting codes. A minimum distance decoder for a rank-metric code $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ takes a word $r \in \mathbb{F}_q^{n\times m}$ and returns a codeword $\hat{x} \in \mathcal{C}$ that is closest to r in rank distance, that is,

$$\hat{x} = \operatorname*{argmin}_{x\in\mathcal{C}} \mathsf{rank}\,(r - x). \tag{2.22}$$

Note that if $d_{\mathrm{R}}(x, r) < d_{\mathrm{R}}(\mathcal{C})/2$, then a minimum distance decoder is guaranteed to return $\hat{x} = x$. Throughout this text, we refer to problem (2.22) as the *conventional* rank-metric decoding problem.

There is a rich coding theory for rank-metric codes that is analogous to the classical coding theory in the Hamming metric. In particular, the Singleton bound for the rank metric [37, 41] (see also [27, 42, 43]) states that every rank-metric code $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ with minimum rank distance d must satisfy

$$|\mathcal{C}| \leq q^{\max\{n,m\}(\min\{n,m\}-d+1)}. \tag{2.23}$$

Codes that achieve this bound are called *maximum-rank-distance* (MRD) codes and they are known to exist for all choices of parameters q, n, m and $d \leq \min\{n, m\}$ [37].

In order to construct rank-metric codes, it is useful to endow $\mathbb{F}_q^{n\times m}$ with additional algebraic properties. Specifically, it is useful to regard $\mathbb{F}_q^{1\times m}$ as the finite field $\mathbb{F}_{q^m}$ (i.e.,

endowing it with a multiplication operation). More precisely, let $\mathcal{A}$ be a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. Then we can extend the bijection $[\cdot]_{\mathcal{A}} \colon \mathbb{F}_{q^m} \to \mathbb{F}_q^{1\times m}$ to a bijection $[\cdot]_{\mathcal{A}} \colon \mathbb{F}_{q^m}^n \to \mathbb{F}_q^{n\times m}$ given by

$$\left[x\right]_{\mathcal{A}} = \begin{bmatrix} \left[x_0\right]_{\mathcal{A}} \\ \vdots \\ \left[x_{n-1}\right]_{\mathcal{A}} \end{bmatrix}$$

where $x \in \mathbb{F}_{q^m}^n$. As before, we shall use the simplified notation $\underline{x} \triangleq \left[x\right]_{\mathcal{A}}$ when $\mathcal{A}$ is fixed. Note that $x \in \mathbb{F}_{q^m}^n$ is a length-n (column) *vector* over the field $\mathbb{F}_{q^m}$, while $\underline{x}$ is an $n \times m$ matrix over $\mathbb{F}_q$. Thus, concepts such as the rank of a vector $x \in \mathbb{F}_{q^m}^n$ or the rank distance between vectors $x, y \in \mathbb{F}_{q^m}^n$ are naturally defined through their matrix counterparts. In this context, a rank-metric code $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$ is simply a block code of length n over $\mathbb{F}_{q^m}$. (Note that, differently from classical coding theory, here we treat each codeword as a *column* vector.)

It is particularly useful to consider *linear* block codes over $\mathbb{F}_{q^m}$. For linear (n, k) codes over $\mathbb{F}_{q^m}$ with minimum rank distance d, the Singleton bound becomes

$$d \leq \min\left\{1, \frac{m}{n}\right\}(n-k)+1. \tag{2.25}$$

Note that the classical Singleton bound $d \leq n-k+1$ can be achieved only if $m \geq n$; that is, a code has $d = n-k+1$ if and only if it is MRD *and* $m \geq n$. The similarity with the classical Singleton bound is not accidental: every MRD code with $m \geq n$ is also MDS as a block code over $\mathbb{F}_{q^m}$.

Such a class of linear MRD codes can be characterized by the following theorem [37].

Theorem 2.1: Let $\mathcal{C}$ be a linear (n, k) code over $\mathbb{F}_{q^m}$ with parity-check matrix $H \in \mathbb{F}_{q^m}^{(n-k)\times n}$. Then $\mathcal{C}$ is an MRD code with $m \geq n$ if and only if the matrix HT is nonsingular for any full-rank matrix $T \in \mathbb{F}_q^{n\times(n-k)}$.

For $m \geq n$, an important class of rank-metric codes was proposed by Gabidulin [37]. For convenience, let $[i]$ denote q^i. A *Gabidulin code* is a linear (n, k) code over $\mathbb{F}_{q^m}$ defined

by the parity-check matrix

$$H = \begin{bmatrix} h_0^{[0]} & h_1^{[0]} & \cdots & h_{n-1}^{[0]} \\ h_0^{[1]} & h_1^{[1]} & \cdots & h_{n-1}^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ h_0^{[n-k-1]} & h_1^{[n-k-1]} & \cdots & h_{n-1}^{[n-k-1]} \end{bmatrix} \tag{2.26}$$

where the elements $h_0, \ldots, h_{n-1} \in \mathbb{F}_{q^m}$ are linearly independent over $\mathbb{F}_q$. It can be shown that the minimum rank distance of a Gabidulin code is $d = n - k + 1$, so it is an MRD code [37]. Encoding and decoding of Gabidulin codes can be performed efficiently; this topic is covered in more detail in Chapter 6.

Besides the matrix version and the block version, yet a third (and most general) definition of rank-metric codes is possible. This definition arises when we take the code "alphabet" to be, rather than $\mathbb{F}_q^{1\times m}$ or $\mathbb{F}_{q^m}$, a general vector space $\mathcal{V}$ with dimension m over $\mathbb{F}_q$.

Consider the set $\mathcal{V}^n$ of n-tuples with components from $\mathcal{V}$. Recall, once again, that we may establish a bijection between $\mathcal{V}^n$ and $\mathbb{F}_q^{n\times m}$ once a basis is fixed for $\mathcal{V}$ over $\mathbb{F}_q$. Under this bijection, the rank of a tuple $x \in \mathcal{V}^n$ can be defined as the rank of the corresponding matrix, and similarly for the rank distance. This turns $\mathcal{V}^n$ into a metric space. A rank-metric code is simply a subset of $\mathcal{V}^n$.

This abstract definition is particularly useful when $\mathcal{V}$ is also a vector space over a larger field, say, over $\mathbb{F}_{q^\ell}$. In this situation, we may explicitly refer to a rank-metric code over $\mathcal{V}/\mathbb{F}_q$, in order to emphasize the field $\mathbb{F}_q$ with respect to which a bijection is established. Note that this definition encompasses the previous two, as we can take $\mathcal{V} = \mathbb{F}_q^{1\times m}$ or $\mathcal{V} = \mathbb{F}_{q^m}$.

For convenience, define a "matrix-by-tuple" multiplication in the natural way: for $A = [A_{ij}] \in \mathbb{F}_q^{k\times n}$ and $x = [x_i] \in \mathcal{V}^n$, denote by Ax the tuple $y = [y_i] \in \mathcal{V}^k$ given by $y_i = \sum_{j=0}^{n-1} A_{ij}x_j$, $i = 0, \ldots, k-1$.

Note that, when $m \geq n$, the Singleton bound (2.25) exhibits no dependency on m.

Thus, if $n|m$, any linear (n,k) MRD code over $\mathbb{F}_{q^n}/\mathbb{F}_q$ can be used to construct an MRD code over $\mathcal{V}/\mathbb{F}_q$, as shown in the following theorem.

Theorem 2.2: Let $\mathcal{V}$ be a vector space over $\mathbb{F}_{q^n}$. Let $\mathcal{C}_1$ be a linear (n,k) MRD code over $\mathbb{F}_{q^n}$ with generator and parity-check matrices $G \in \mathbb{F}_{q^n}^{k\times n}$ and $H \in \mathbb{F}_{q^n}^{(n-k)\times n}$, respectively. Then $\mathcal{C} \subseteq \mathcal{V}^n$ defined by

$$\mathcal{C} = \{G^T U,\, U \in \mathcal{V}^k\} = \{X \in \mathcal{V}^n \colon HX = 0\}$$

is an MRD code over $\mathcal{V}/\mathbb{F}_q$.

Proof: Let r be the dimension of $\mathcal{V}$ as a vector space over $\mathbb{F}_{q^n}$. Then $\mathcal{C}$ is isomorphic to an r-fold Cartesian product of $\mathcal{C}_1$ with itself and thus $d_{\mathrm{R}}(\mathcal{C}) = d_{\mathrm{R}}(\mathcal{C}_1)$. ∎

From a matrix perspective (i.e., by expanding $\mathcal{V}$ as a vector space over $\mathbb{F}_{q^n}$), the code $\mathcal{C}$ in Theorem 2.2 is the set of all $n \times r$ matrices over $\mathbb{F}_{q^n}$ obtained by "gluing together" any $r = m/n$ codewords of $\mathcal{C}_1$.

2.4 Linearized Polynomials

Linear transformations from $\mathbb{F}_{q^m}$ to itself can be elegantly described in terms of linearized polynomials. A *linearized polynomial* or *q-polynomial* over $\mathbb{F}_{q^m}$ [38] is a polynomial of the form

$$f(x) = \sum_{i=0}^{n} f_i x^{[i]}$$

where $f_i \in \mathbb{F}_{q^m}$. If $f_n \neq 0$, we call n the *q-degree* of $f(x)$. It is easy to see that evaluation of a linearized polynomial is indeed an $\mathbb{F}_q$-linear transformation from $\mathbb{F}_{q^m}$ to itself. In particular, the set of roots in $\mathbb{F}_{q^m}$ of a linearized polynomial is the kernel of the associated map (and therefore a subspace of $\mathbb{F}_{q^m}$).

It is well-known that the set of linearized polynomials over $\mathbb{F}_{q^m}$ forms a noncommutative ring (actually, an algebra over $\mathbb{F}_q$) under addition and composition (evaluation). The latter operation is usually called *symbolic multiplication* in this context and denoted by $f(x) \otimes g(x) = f(g(x))$. Note that if n and k are the q-degrees of $f(x)$ and $g(x)$, respectively, then $P(x) = f(x) \otimes g(x)$ has q-degree equal to $t = n + k$. Moreover, the coefficients of $P(x)$ can be computed as

$$P_\ell = \sum_{i=\max\{0,\ell-k\}}^{\min\{\ell,n\}} f_i g_{\ell-i}^{[i]} = \sum_{j=\max\{0,\ell-n\}}^{\min\{\ell,k\}} f_{\ell-j} g_j^{[\ell-j]}, \quad \ell = 0, \ldots, t.$$

In particular, if $n \leq k$, then

$$P_\ell = \sum_{i=0}^{n} f_i g_{\ell-i}^{[i]}, \qquad n \leq \ell \leq k \tag{2.30}$$

while if $k \leq n$, then

$$P_\ell = \sum_{j=0}^{k} f_{\ell-j} g_j^{[\ell-j]}, \qquad k \leq \ell \leq n. \tag{2.31}$$

One of the most convenient properties of linearized polynomials is to provide canonical maps for specified kernels. There is a unique linearized polynomial $M_{\mathcal{S}}(x) = \sum_{i=0}^{t} M_i x^{[i]}$ with smallest q-degree and $M_0 = 1$ whose root space contains a specified set $\mathcal{S} \subseteq \mathbb{F}_{q^m}$. This polynomial is called the *minimal q-polynomial* of $\mathcal{S}$. The q-degree of $M_{\mathcal{S}}(x)$ is precisely equal to the dimension of the space spanned by $\mathcal{S}$, and is also equal to the nullity of $M_{\mathcal{S}}(x)$ as a linear map. The polynomial $M_{\mathcal{S}}(x)$ can be computed recursively. Let $\{s_1, \ldots, s_t\}$ be a basis for the space spanned by $\mathcal{S}$. Then we can find $M_{\mathcal{S}}(x)$ by computing $M_{\{s_1\}}(x) = x - x^{[1]} s_1 / s_1^{[1]}$ and, for $i = 2, \ldots, t$, $z_i = M_{\{s_1,\ldots,s_{i-1}\}}(s_i)$ and $M_{\{s_1,\ldots,s_i\}}(x) = M_{z_i}(x) \otimes M_{\{s_1,\ldots,s_{i-1}\}}(x)$.

It is useful to define two notions of *reverse* linearized polynomials. In the first definition, we are given a linearized polynomial $f(x) = \sum_{i=0}^{t} f_i x^{[i]}$ whose q-degree does not exceed a specified number t. Then the *partial q-reverse* of $f(x)$ is the polynomial $\tilde{f}(x) = \sum_{i=0}^{t} \tilde{f}_i x^{[i]}$ given by $\tilde{f}_i = f_{t-i}^{[i-t]}$, for $i = 0, \ldots, t$. If t is not specified, it is taken as the q-degree of $f(x)$. In the second definition, we are given a linearized polynomial

Table 2.1: Complexity of operations with linearized polynomials. See text for details.

Operation	Number of operations in $\mathbb{F}_{q^m}$			
	Multiplications	Additions	q-Exponentiations	Inversions
$f(x) \otimes g(x)$	$n'k'$	nk	nk'	–
$f(\beta)$	n'	n	n	–
$\mathcal{S} \mapsto M_{\mathcal{S}}(x)$	t^2	$t(t-1)$	t^2	t

$f(x) = \sum_{i=0}^{m-1} f_i x^{[i]}$ of q-degree at most $m-1$. Then the *full q-reverse* of $f(x)$ is the polynomial $\bar{f}(x) = \sum_{i=0}^{m-1} \bar{f}_i x^{[i]}$ given by $\bar{f}_i = f^{[i]}_{-i \bmod m}$, $i = 0, \ldots, m-1$. Note that a full q-reverse can be seen as a partial q-reverse with $t = m$ and indices taken modulo m. As we will see in Chapter 6, a full q-reverse is closely related to the *transpose* of a matrix representing a linear map.

It is worth mentioning that Gabidulin codes can be described in terms of linearized polynomials, just as Reed-Solomon codes can be described in terms of conventional polynomials. It is shown in [37] that the generator matrix of a Gabidulin code is of the form

$$G = \begin{bmatrix} g_0^{[0]} & g_1^{[0]} & \cdots & g_{n-1}^{[0]} \\ g_0^{[1]} & g_1^{[1]} & \cdots & g_{n-1}^{[1]} \\ \vdots & \vdots & \ddots & \vdots \\ g_0^{[k-1]} & g_1^{[k-1]} & \cdots & g_{n-1}^{[k-1]} \end{bmatrix}$$

for some linearly independent $g_0, \ldots, g_{n-1} \in \mathbb{F}_{q^m}$. Thus, a Gabidulin code may be described as the set of all codewords $c \in \mathbb{F}_{q^m}^n$ such that $c_i = f(g_i)$, $i = 0, \ldots, n-1$, for some linearized polynomial $f(x)$ of q-degree at most $k-1$.

Table 2.1 lists the complexity of some useful operations with linearized polynomials. In the table, β is any element of $\mathbb{F}_{q^m}$, t denotes the size of $\mathcal{S} \subseteq \mathbb{F}_{q^m}$, and $f(x)$ and $g(x)$ are linearized polynomials with q-degrees n and k, respectively. Additionally, $n' = n$ if

$f(x)$ is monic and $n' = n + 1$ otherwise; similarly, $k' = k$ if $g(x)$ is monic and $k' = k + 1$ otherwise.

2.5 Operations in Normal Bases

Let $\alpha \in \mathbb{F}_{q^m}$ be a normal element, and fix a basis $\mathcal{A} = \left\{\alpha^{[0]}, \ldots, \alpha^{[m-1]}\right\}$. In a practical implementation, an element $a \in \mathbb{F}_{q^m}$ is usually represented as the vector $\underline{a}$ over the base field $\mathbb{F}_q$. It is useful to review the complexity of some common operations in $\mathbb{F}_{q^m}$ when elements are represented in this form. For convenience, let $[\![\alpha]\!]$ denote the column vector $\begin{bmatrix}\alpha^{[0]} & \cdots & \alpha^{[m-1]}\end{bmatrix}^T$. Then any element $a \in \mathbb{F}_{q^m}$ can be written as $a = \underline{a}\,[\![\alpha]\!]$.

For a vector $\underline{a} = \begin{bmatrix}a_0, \ldots, a_{m-1}\end{bmatrix} \in \mathbb{F}_q^{1\times m}$, let $\underline{a}^{\leftarrow i}$ denote a cyclic shift to the left by i positions, that is, $\underline{a}^{\leftarrow i} = \begin{bmatrix}a_i, \ldots, a_{m-1}, a_0, \ldots, a_{i-1}\end{bmatrix}$. Similarly, let $\underline{a}^{\rightarrow i} = \underline{a}^{\leftarrow m-i}$. In this notation, we have $a^{[i]} = \underline{a}^{\rightarrow i}\,[\![\alpha]\!]$, or $\underline{a^{[i]}} = \underline{a}^{\rightarrow i}$. Thus, q-exponentiation in a normal basis corresponds to a cyclic shift.

Multiplications in normal bases are usually performed in the following way. Let $T = [T_{ij}] \in \mathbb{F}_q^{n\times m}$ be a matrix such that $\alpha\alpha^{[i]} = \sum_{j=0}^{m-1} T_{ij}\alpha^{[j]}$, $i = 0, \ldots, m-1$. The matrix T is called the *multiplication table* of the normal basis. The number of nonzero entries in T is denoted by $C(T)$ and is called the *complexity* of the normal basis [44]. Note that $\alpha\,[\![\alpha]\!] = T\,[\![\alpha]\!]$. It can be shown that, if $a, b \in \mathbb{F}_{q^m}$, then

$$\underline{ab} = \sum_{i=0}^{m-1} b_i \left(\underline{a}^{\leftarrow i} T\right)^{\rightarrow i}.$$

Thus, a general multiplication in a normal basis requires $mC(T) + m^2$ multiplications and $mC(T) - 1$ additions in $\mathbb{F}_q$. Clearly, this is only efficient if T is sparse; otherwise, it is more advantageous to convert back and forth to a polynomial basis to perform multiplication.

It is a well-known result that the complexity of a normal basis is lower bounded by $2m - 1$. Bases that achieve this complexity are called *optimal*. More generally, low-

Table 2.2: Complexity of operations in $\mathbb{F}_{q^m}$ using a normal basis constructed via Gauss periods, for q a power of 2.

Operation in $\mathbb{F}_{q^m}$	Number of operations in $\mathbb{F}_q$		
	Multiplications	Additions	Inversions
Multiplication	m^2	$m(C(T)-1)$	–
Addition	–	m	–
Inversion	$\frac{5}{2}m^2 + O(m)$	$4m^2 + O(m)$	$m+2$

complexity (but not necessarily optimal) normal bases can be constructed using *Gauss periods*, as described in detail in [44]. For $q = 2^s$, such a construction is possible if and only if m satisfies $\gcd(m, s) = 1$ and $8 \nmid m$ [45]. As an example, for $q = 256$, this condition is satisfied for any odd m. Among the odd $m \leq 100$, the normal bases that result are in fact optimal when $m = 3$, 5, 9, 11, 23, 29, 33, 35, 39, 41, 51, 53, 65, 69, 81, 83, 89, 95, 99. For $q = 2^s$ and odd m, all of the normal bases constructed by Gauss periods are self-dual [45].

An interesting fact about a normal basis constructed via Gauss periods is that its multiplication table T lies entirely in the prime field $\mathbb{F}_p$, where p is the characteristic of q. This in turn implies that the minimal polynomial of α has coefficients in $\mathbb{F}_p$ and the conversion matrices from/to the standard basis $\{\alpha^0, \alpha^1, \ldots, \alpha^{m-1}\}$ have entries also in $\mathbb{F}_p$.

In this thesis, we are mostly interested in the case $p = 2$. In this case, multiplication by T can be done simply by using XORs. In Table 2.2, we give the complexity of each operation in $\mathbb{F}_{q^m}$ assuming that $p = 2$. We also assume that q-exponentiations are free. Note that inversion can be performed using the extended Euclidean algorithm on a standard basis, which takes at most $\frac{5}{2}m^2 + O(m)$ multiplications, $2m^2 + O(m)$ additions and $m + 2$ inversions in $\mathbb{F}_q$ [46]. Converting between standard and normal basis takes at

most $m(m-1)$ additions since the conversion matrices lie in $\mathbb{F}_2$.

Chapter 3

The Linear Network Coding Channel

This chapter describes a channel model for communication over a network subject to packet errors. This model is an extension of the linear network coding model, which is reviewed in Section 3.1.

Network coding was proposed in [2] by Ahlswede, Cai, Li, and Yeung. Linear network coding was proposed in [3] by Li et al., and later received an algebraic formulation by Koetter and Médard [4]. The idea of random linear network coding was proposed by Ho et al. [5, 6] and subsequently implemented by Chou et al. [47]. The additive matrix model for network coding with errors, which we review in this chapter, was first mentioned in [13] and [15], although it had been implicit from [4]. Subsequent work that is based on this model includes [48] and [18].

3.1 Linear Network Coding

Consider a communication network represented by a directed multigraph with unit capacity edges. The network is used for single-source *multicasting*: there is a single source node, which produces a message, and multiple destination nodes, all of which demand

the message. All the remaining nodes are called internal nodes. Each link in the network is assumed to transport, free of errors, a packet (i.e., a vector) of m symbols from a finite field $\mathbb{F}_q$. A packet transmitted on a link directed from a node u to a node v is said to be an outgoing packet of u and an incoming packet of v. The message to be transmitted by the source node is a matrix $X \in \mathbb{F}_q^{n \times m}$, where n is a positive integer. The n rows of this matrix, denoted by $X_1, \ldots, X_n \in \mathbb{F}_q^{1 \times m}$, are assumed to be the incoming packets of the source node.

The network runs as follows: at each transmission opportunity, a node computes an outgoing packet as an $\mathbb{F}_q$-linear combination of incoming packets. It is easy to see that a packet P_i transmitted on an edge i must be an $\mathbb{F}_q$-linear combination of the source packets, i.e., $P_i = g_i X$, where $g_i \in \mathbb{F}_q^{1 \times n}$. The vector g_i is called the *global coding vector* of edge i. Consider some specific destination node, and let $Y \in \mathbb{F}_q^{N \times m}$ be a matrix whose rows are the N packets received by this node. It follows that

$$Y = AX \tag{3.1}$$

for some matrix $A \in \mathbb{F}_q^{N \times n}$, which is called the *transfer matrix* of the network (from the source node to that destination node). It is clear that all destination nodes can obtain X if (and only if, in the case that X is a uniform random variable) all the transfer matrices have rank n, in which case the network code is said to be *feasible*.

It is important to mention that (3.1) holds under a variety of situations:

- Any network topology is allowed (in particular, the network may contain cycles);
- The network may be reused for multiple rounds (called generations in [47]), exhibiting possibly different transfer matrices;
- Packet transmissions may contain delays and the overall topology may be time-varying;
- Wireless broadcast transmissions may be modeled by constraining a node to send the same packet on each of its outgoing links;

- Erasure channels may be modeled by assuming that each link is the instantiation of a successful packet transmission.

Let h be the minimum source-destination min-cut among all the destination nodes. In a general network code (not necessarily linear), nodes are allowed to perform any arbitrary operations on packets. It is shown in [2] that a feasible network code exists if and only if $n \leq h$ and the packet size q^m is sufficiently large. Assuming that $n \leq h$, a remarkable result of [3] (see also [4]) is that a feasible *linear* network code always exists if the field size q is sufficiently large. If q is at least the number of destination nodes, such a code can be constructed in polynomial time by a centralized algorithm [49]. Alternatively, the network code can be constructed in a decentralized fashion by having nodes select coding coefficients uniformly at random from $\mathbb{F}_q$. As shown in [6], such a random network code is feasible with high probability if q is sufficiently large. In order to inform the destination nodes about the specific realization of the network code (or rather their corresponding transfer matrices), a typical approach is to prepend to X an identity matrix [6, 47]. In this way, each packet will contain a header that records its global coding vector as the packet traverses the network.

For the remainder of this text, we assume for simplicity that there is a single destination node. For the problems considered here, each destination node will behave in a similar manner, so any generalization to multicast is straightforward.

3.2 Linear Network Coding with Packet Errors

We now extend the model of the previous section to include the possibility of packet errors. That model assumes that, for every link in the network, the following two conditions are satisfied:

- the link is an error-free channel;
- the transmitter node at the link complies with the established protocol.

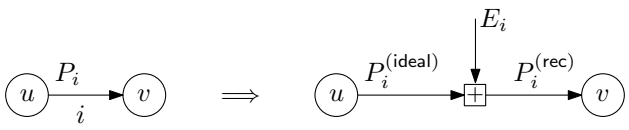

Figure 3.1: Modeling the injection of an error packet.

If any of these two conditions is violated, then a packet error may occur in the link, i.e., the packet received by a node may be different from what we would otherwise expect. This is the scenario that we consider from now on.

Consider a link i, and let u and v be, respectively, the transmitter and receiver nodes at i, as illustrated in Fig. 3.1. Let $P_i^{(\mathsf{rec})}$ denote the packet effectively received by v and let $P_i^{(\mathsf{ideal})}$ denote the packet that *would have been received* by v if both conditions above were satisfied for i. In other words, $P_i^{(\mathsf{ideal})}$ must be a linear combination of $P_j^{(\mathsf{rec})}$ for all links j entering u. We define the error packet at link i as the difference $E_i = P_i^{(\mathsf{rec})} - P_i^{(\mathsf{ideal})}$. As illustrated in Fig. 3.1, we can now view $P_i^{(\mathsf{rec})}$ as the addition of a prescribed packet $P_i^{(\mathsf{ideal})}$ with a possibly corrupting packet E_i. Note that $E_i = 0$ means that no error occurred at link i.

Let $|\mathcal{E}|$ denote the number of edges in the network, and assume that edges are indexed from 1 to $|\mathcal{E}|$. Let E be an $|\mathcal{E}| \times m$ matrix whose rows are the error packets $E_1, \ldots, E_{|\mathcal{E}|}$. By linearity of the network, we can write

$$Y = AX + FE, \tag{3.2}$$

where F is an $N \times |\mathcal{E}|$ matrix corresponding to the overall linear transformation applied to $E_1, \ldots, E_{|\mathcal{E}|}$ on route to the destination. The number of nonzero rows of E, $\mathsf{wt}(E)$, gives the total number of packet errors occurring in the network.

Observe that this model can represent not only the occurrence of random link errors, but also the action of malicious nodes. A malicious node may transmit erroneous packets on some or all of its outgoing links, and may also refuse to transmit some packets. The latter case is modeled by setting $E_i = -P_i^{(\mathsf{ideal})}$, so that $P_i^{(\mathsf{rec})} = 0$. In any case, $\mathsf{wt}(E)$

gives the total number of "packet interventions" performed by all malicious nodes and thus gives a sense of the total adversarial "effort" employed towards jamming the network.

Equation (3.2) is our basic model of an additive error channel induced by linear network coding, and we will refer to it as the *linear network coding channel* (LNCC). The channel input and output alphabets are given by $\mathbb{F}_q^{n\times m}$ and $\mathbb{F}_q^{N\times m}$, respectively. The conditional probability of Y given X is given by

$$\mathsf{Pr}(Y|X) = \mathsf{Pr}(A, F, E|X)\mathbf{1}[Y = AX + FE].$$

Clearly, we still need to specify joint distribution of A, F and E given X in order to fully specify the channel. For generality, we leave this open in the definition of the LNCC and proceed below to discuss a number of special cases.

Definition 3.1: An LNCC is called *coherent* if the transfer matrix A is a constant known to the receiver; otherwise it is called *noncoherent*.

The definition of coherent LNCC is motivated by a scenario where the network code is designed by a central entity and informed to each node in the network before transmission starts; in this case, the matrix F would naturally be known at the receiver. A noncoherent LNCC, on the other hand, is motivated by the use of random network coding, in which case both A and F would be random and unknown.[1] Note that Definition 3.1 also applies also to an LNCC free of errors (as in Section 3.1).

Definition 3.2: An LNCC is said to have a *random* or *probabilistic* error model if the matrix E is a random variable independent from (A, F, X). Otherwise, the error model is said to be *adversarial*.

[1] One might wonder if our definition is appropriate for the case where A is known only at the transmitter but not at the receiver. We believe that this situation is undeserving of a definition for the following reasons: if the network code varies from time to time, then it is unlikely that the transmitter (and only the transmitter) would have knowledge of A; on the other hand, if the network code is fixed, then it should be a simple matter to communicate the transfer matrix to the receiver before message transmission (say, through a control channel as the receiver joins the system, or even by using a single round of noncoherent network coding).

A random error model corresponds to dropping the assumption that the links are error-free, while an adversarial error model corresponds to dropping the assumption that the nodes comply with the protocol, i.e., some nodes may be malicious, as discussed above.

A main assumption made throughout this thesis is that the number of error packets injected in the network is limited, i.e., $\mathsf{wt}(E) \leq t$. This assumption allows us to rewrite (3.2) as

$$Y = AX + DZ, \tag{3.4}$$

where $Z \in \mathbb{F}_q^{t \times m}$ consists of the (potentially) nonzero rows of E, and $D \in \mathbb{F}_q^{N \times t}$ is a submatrix of F. We use expression (3.4) for the most part of this thesis.

In the next three chapters, we address the problem of error correction for an LNCC (either coherent or noncoherent) with an adversarial error model. An LNCC with a probabilistic error model is investigated in Chapter 7.

Chapter 4

Error Control under an Adversarial Error Model

This chapter addresses the problem of error control in linear network coding under an adversarial error model. One main insight of our approach is that both the coherent and the noncoherent LNCCs can be characterized by a parameter that we call *maximum discrepancy*. In Section 4.1, we start by proposing a general theory for adversarial channels that admit such characterization. The theory generalizes classical coding theory in that it provides us with a function—the Δ-*distance*—that perfectly describes the correction capability of a code. We use this approach in Sections 4.2 and 4.3 to study coherent and noncoherent network coding, respectively.

Our results in Sections 4.2 and 4.3 are closely related to those of Yeung et al. [11, 48] and Kötter and Kschischang [18], respectively. Comparisons with such works are made in the corresponding Sections 4.2.2 and 4.3.2. Our main result in Section 4.2 is to show that, under certain mild conditions, MRD codes are optimal for the correction of adversarial errors in coherent network coding. Section 4.3 contains two main results: we provide a precise framework for assessing optimality of error-correcting codes for noncoherent network coding (which was missing in [18]); and we show that a class of

MRD-based codes—which we call *liftings* of MRD codes—are nearly optimal for any practical purposes.

Finally, Section 4.4 discusses the decoding problems that arise in coherent network coding, as well as in noncoherent network coding when the lifting approach is used, and shows that these two problems are mathematically equivalent. The discussion opens grounds for investigating efficient decoding algorithms, which is done in Chapter 5.

In this chapter we use the following notation. Let $\mathcal{X}$ be a set, and let $\mathcal{C} \subseteq \mathcal{X}$. Whenever a function $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{N}$ is defined, denote

$$d(\mathcal{C}) \triangleq \min_{x,x' \in \mathcal{C}\colon\, x \neq x'} d(x, x').$$

If $d(x, x')$ is called a "distance" between x and x', then $d(\mathcal{C})$ is called the *minimum* "distance" of $\mathcal{C}$.

4.1 A General Approach

This section presents a general approach to error correction over adversarial channels. This approach is specialized to coherent and noncoherent network coding in sections 4.2 and 4.3, respectively.

4.1.1 Adversarial Channels

An *adversarial channel* is specified by a finite input alphabet $\mathcal{X}$, a finite output alphabet $\mathcal{Y}$ and a collection of *output sets* $\mathcal{Y}(x) \subseteq \mathcal{Y}$ for all $x \in \mathcal{X}$. For each input x, the output y is constrained to be in $\mathcal{Y}(x)$ but is otherwise arbitrarily chosen by an adversary. The constraint on the output is important: otherwise, the adversary could prevent communication simply by mapping all inputs to the same output. No further restrictions are imposed on the adversary; in particular, the adversary is potentially omniscient and has unlimited computational power.

A code for an adversarial channel is a subset[1] $\mathcal{C} \subseteq \mathcal{X}$. We say that a code is *unambiguous* for a channel if the input codeword can always be uniquely determined from the channel output. More precisely, a code $\mathcal{C}$ is unambiguous if the sets $\mathcal{Y}(x)$, $x \in \mathcal{C}$, are pairwise disjoint. The importance of this concept lies in the fact that, if the code is *not* unambiguous, then there exist codewords x, x' that are *indistinguishable* at the decoder: if $\mathcal{Y}(x) \cap \mathcal{Y}(x') \neq \emptyset$, then the adversary can (and will) exploit this ambiguity by mapping both x and x' to the same output.

A decoder for a code $\mathcal{C}$ is any function $\hat{x}\colon \mathcal{Y} \to \mathcal{C} \cup \{f\}$, where $f \notin \mathcal{C}$ denotes a decoding failure (detected error). When $x \in \mathcal{C}$ is transmitted and $y \in \mathcal{Y}(x)$ is received, a decoder is said to be *successful* if $\hat{x}(y) = x$. We say that a decoder is *infallible* if it is successful for all $y \in \mathcal{Y}(x)$ and all $x \in \mathcal{C}$. Note that the existence of an infallible decoder for $\mathcal{C}$ implies that $\mathcal{C}$ is unambiguous. Conversely, given any unambiguous code $\mathcal{C}$, one can always find (by definition) a decoder that is infallible. One example is the exhaustive decoder

$$\hat{x}(y) = \begin{cases} x & \text{if } y \in \mathcal{Y}(x) \text{ and } y \notin \mathcal{Y}(x') \text{ for all } x' \in \mathcal{C},\ x' = x \\ f & \text{otherwise.} \end{cases}$$

In other words, an exhaustive decoder returns x if x is the unique codeword that could possibly have been transmitted when y is received, and returns a failure otherwise.

Ideally, one would like to find a large (or largest) code that is unambiguous for a given adversarial channel, together with a decoder that is infallible (and computationally-efficient to implement).

[1] There is no loss of generality in considering a single channel use, since the channel may be taken to correspond to multiple uses of a simpler channel.

4.1.2 Discrepancy

It is useful to consider adversarial channels parameterized by an *adversarial effort* $t \in \mathbb{N}$. Assume that the output sets are of the form

$$\mathcal{Y}(x) = \{y \in \mathcal{Y} \colon \Delta(x, y) \leq t\} \tag{4.3}$$

for some $\Delta \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{N}$. The value $\Delta(x, y)$, which we call the *discrepancy* between x and y, represents the minimum effort needed for an adversary to transform an input x into an output y. The value of t then represents the maximum adversarial effort (maximum discrepancy) allowed in the channel.

In principle, there is no loss of generality in assuming (4.3) since, by properly defining $\Delta(x, y)$, one can always express any $\mathcal{Y}(x)$ in this form. For instance, one could set $\Delta(x, y) = 0$ if $y \in \mathcal{Y}(x)$, and $\Delta(x, y) = \infty$ otherwise. However, such a definition would be of no practical value since $\Delta(x, y)$ would be merely an indicator function. Thus, an effective limitation of our model is that it requires channels that are *naturally* characterized by some discrepancy function. In particular, one should be able to interpret the maximum discrepancy t as the level of "degradedness" of the channel.

On the other hand, the assumption $\Delta(x, y) \in \mathbb{N}$ imposes effectively no constraint. Since $|\mathcal{X} \times \mathcal{Y}|$ is finite, given any "naturally defined" $\Delta' \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, one can always shift, scale and round the image of Δ' in order to produce some $\Delta \colon \mathcal{X} \times \mathcal{Y} \to \mathbb{N}$ that induces the same output sets as Δ' for all t.

Example 4.1: Let us use the above notation to define a t-error channel, i.e., a vector channel that introduces at most t symbol errors (arbitrarily chosen by an adversary). Assume that the channel input and output alphabets are given by $\mathcal{X} = \mathcal{Y} = \mathbb{F}_q^n$. It is easy to see that the channel can be characterized by a discrepancy function that counts the number of components in which an input vector x and an output vector y differ. More precisely, we have $\Delta(x, y) = d_{\mathrm{H}}(x, y)$, where $d_{\mathrm{H}}(\cdot, \cdot)$ denotes the *Hamming distance* function. □

A main feature of our proposed discrepancy characterization is to allow us to study a whole family of channels (with various levels of degradedness) under the same framework. For instance, we can use a single decoder for all channels in the same family. Define the *minimum-discrepancy decoder* given by

$$\hat{x} = \operatorname*{argmin}_{x \in \mathcal{C}} \Delta(x, y) \tag{4.4}$$

where any ties in (4.4) are assumed to be broken arbitrarily. It is easy to see that a minimum-discrepancy decoder is infallible provided that the code is unambiguous. Thus, we can safely restrict attention to a minimum-discrepancy decoder, regardless of the maximum discrepancy t in the channel.

4.1.3 Correction Capability

Given a fixed family of channels—specified by $\mathcal{X}$, $\mathcal{Y}$ and $\Delta(\cdot, \cdot)$, and parameterized by a maximum discrepancy t—we wish to identify largest (worst) channel parameter for which we can guarantee successful decoding. We say that a code is *t-discrepancy-correcting* if it is unambiguous for a channel with maximum discrepancy t. The *discrepancy-correction capability* of a code $\mathcal{C}$ is the largest t for which $\mathcal{C}$ is t-discrepancy-correcting.

We start by giving a general characterization of the discrepancy-correction capability. Let the function $\tau \colon \mathcal{X} \times \mathcal{X} \to \mathbb{N}$ be given by

$$\tau(x, x') = \min_{y \in \mathcal{Y}} \max\{\Delta(x, y),\, \Delta(x', y)\} - 1. \tag{4.5}$$

We have the following result.

Proposition 4.1: The discrepancy-correction capability of a code $\mathcal{C}$ is given exactly by $\tau(\mathcal{C})$. In other words, $\mathcal{C}$ is t-discrepancy-correcting if and only if $t \leq \tau(\mathcal{C})$.

Proof: Suppose that the code is not t-discrepancy-correcting, i.e., that there exist some distinct $x, x' \in \mathcal{C}$ and some $y \in \mathcal{Y}$ such that $\Delta(x, y) \leq t$ and $\Delta(x', y) \leq t$. Then

$\tau(\mathcal{C}) \leq \tau(x,x') \leq \max\{\Delta(x,y), \Delta(x',y)\} - 1 \leq t-1 < t$. In other words, $\tau(\mathcal{C}) \geq t$ implies that the code is t-discrepancy-correcting.

Conversely, suppose that $\tau(\mathcal{C}) < t$, i.e., $\tau(\mathcal{C}) \leq t-1$. Then there exist some distinct $x, x' \in \mathcal{C}$ such that $\tau(x,x') \leq t-1$. This in turn implies that there exists some $y \in \mathcal{Y}$ such that $\max\{\Delta(x,y), \Delta(x',y)\} \leq t$. Since this implies that both $\Delta(x,y) \leq t$ and $\Delta(x',y) \leq t$, it follows that the code is not t-discrepancy-correcting. ∎

At this point, it is tempting to define a "distance-like" function given by $2(\tau(x,x')+1)$, since this would enable us to immediately obtain results analogous to those of classical coding theory (such as the error correction capability being half the minimum distance of the code). This approach has indeed been taken in previous works, such as [50]. However, there is a subtle reason why such a definition is neither necessary nor desirable: the function (4.5) is not generally easy to handle analytically. It is important to note that, while $\tau(\mathcal{C})$ has an important practical interpretation—it gives the discrepancy-correction capability of the code—the same cannot (necessarily) be said for $\tau(x,x')$.

It is the objective of this section to propose a fundamentally different function $\delta\colon \mathcal{X} \times \mathcal{X} \to \mathbb{N}$ that, while easier to handle analytically, is such that $\lfloor \frac{\delta(\mathcal{C})-1}{2} \rfloor = \tau(\mathcal{C})$ *for every code* $\mathcal{C}$. Thus, we shall be able to obtain the same results as [50] with much greater mathematical simplicity—which will later turn out to be instrumental for code design.

For $x, x' \in \mathcal{X}$, define the Δ-*distance* between x and x' as

$$\delta(x,x') \triangleq \min_{y \in \mathcal{Y}} \Delta(x,y) + \Delta(x',y). \tag{4.6}$$

Example 4.2: Let us compute the Δ-distance for the channel of Example 4.1. We have $\delta(x,x') = \min_y d_{\mathrm{H}}(x,y) + d_{\mathrm{H}}(x',y) \geq d_{\mathrm{H}}(x,x')$, since the Hamming distance satisfies the triangle inequality. This bound is achievable by taking, for instance, $y = x'$. Thus, $\delta(x,x') = d_{\mathrm{H}}(x,x')$, i.e., the Δ-distance for this channel is given precisely by the Hamming distance. □

The following result justifies our definition of the Δ-distance.

Proposition 4.2: For any code $\mathcal{C}$, we have $\tau(\mathcal{C}) \geq \lfloor (\delta(\mathcal{C}) - 1)/2 \rfloor$.

Proof: This follows from the fact that $\lfloor (a + b + 1)/2 \rfloor \leq \max\{a, b\}$ for all $a, b \in \mathbb{Z}$. ∎

Proposition 4.2 shows that $\delta(\mathcal{C})$ gives a lower bound on the correction capability—therefore providing a correction guarantee. The converse result, however, is not necessarily true in general. Thus, up to this point, the proposed function is only partially useful: it is conceivable that the Δ-distance might be too conservative and give a guaranteed correction capability that is lower than the actual one.

A special case where the converse is true is for a family of channels whose discrepancy function satisfies the following condition:

Definition 4.1: A discrepancy function $\Delta \colon \mathcal{X} \times \mathcal{X} \to \mathbb{N}$ is said to be *normal* if, for all $x, x' \in \mathcal{X}$ and all $0 \leq i \leq \delta(x, x')$, there exists some $y \in \mathcal{Y}$ such that $\Delta(x, y) = i$ and $\Delta(x', y) = \delta(x, x') - i$.

Theorem 4.3: Suppose that $\Delta(\cdot, \cdot)$ is normal. For every code $\mathcal{C} \subseteq \mathcal{X}$, we have $\tau(\mathcal{C}) = \lfloor (\delta(\mathcal{C}) - 1)/2 \rfloor$.

Proof: We just need to show that $\lfloor (\delta(\mathcal{C}) - 1)/2 \rfloor \geq \tau(\mathcal{C})$. Take any $x, x' \in \mathcal{X}$. Since $\Delta(\cdot, \cdot)$ is normal, there exists some $y \in \mathcal{Y}$ such that $\Delta(x, y) + \Delta(x', y) = \delta(x, x')$ and either $\Delta(x, y) = \Delta(x', y)$ or $\Delta(x, y) = \Delta(x', y) - 1$. Thus, $\delta(x, x') \geq 2 \max\{\Delta(x, y), \Delta(x', y)\} - 1$ and therefore $\lfloor (\delta(x, x') - 1)/2 \rfloor \geq \tau(x, x')$. ∎

Theorem 4.3 shows that, for certain families of channels, our proposed Δ-distance achieves the goal of this section: it is a (seemingly) tractable function that precisely describes the correction capability of a code. In particular, the basic result of classical coding theory—that the Hamming distance precisely describes the error correction capability of a code—follows from the fact that the Hamming distance (as a discrepancy function) is normal. As we shall see, much of our effort in the next sections reduces to showing that a specified discrepancy function is normal.

Example 4.3: To give a nontrivial example, let us consider a binary vector channel that introduces at most ρ erasures (arbitrarily chosen by an adversary). The input alphabet is given by $\mathcal{X} = \{0,1\}^n$, while the output alphabet is given by $\mathcal{X} = \{0,1,\epsilon\}^n$, where ϵ denotes an erasure. We may define $\Delta(x,y) = \sum_{i=1}^{n} \Delta(x_i, y_i)$, where

$$\Delta(x_i, y_i) = \begin{cases} 0 & \text{if } y_i = x_i \\ 1 & \text{if } y_i = \epsilon \\ \infty & \text{otherwise} \end{cases} .$$

The output sets are then given by $\mathcal{Y}(x) = \{y \in \mathcal{Y} \colon \Delta(x,y) \leq \rho\}$. In order to compute $\delta(x,x')$, observe the minimization in (4.6). It is easy to see that we should choose $y_i = x_i$ when $x_i = x'_i$, and $y_i = \epsilon$ when $x_i \neq x'_i$. It follows that $\delta(x,x') = 2d_{\mathrm{H}}(x,x')$. Note that $\Delta(x,y)$ is normal. It follows from Theorem 4.3 that a code $\mathcal{C}$ can correct all the ρ erasures introduced by the channel if and only if $2d_{\mathrm{H}}(\mathcal{C}) > 2\rho$. This result precisely matches the well-known result of classical coding theory. □

It is worth clarifying that, while we call $\delta(\cdot,\cdot)$ a "distance," this function may not necessarily be a metric. While symmetry and non-negativity follow from the definition, a Δ-distance may not always satisfy "$\delta(x,y) = 0 \iff x = y$" or the triangle inequality. Nevertheless, we keep the terminology for convenience.

Although this is not our main interest in this thesis, it is worth pointing out that the framework of this section is also useful for obtaining results on error *detection*. Namely, the Δ-distance gives, in general, a lower bound on the discrepancy detection capability of a code under a bounded discrepancy-correcting decoder; when the discrepancy function is normal, then the Δ-distance precisely characterizes this detection capability (similarly as in classical coding theory). For more details on this topic, see Appendix A.

4.2 Coherent Network Coding

4.2.1 A Worst-Case Model and the Rank Metric

The basic channel model for coherent network coding with adversarial errors is a matrix channel with input $X \in \mathbb{F}_q^{n\times m}$, output $Y \in \mathbb{F}_q^{N\times m}$, and channel law given by (3.4), where $A \in \mathbb{F}_q^{N\times n}$ is fixed and known to the receiver, and $Z \in \mathbb{F}_q^{t\times m}$ is arbitrarily chosen by an adversary. Here, we make the following additional assumptions:

- The adversary has unlimited computational power and is omniscient; in particular, the adversary knows both A and X;
- The matrix $D \in \mathbb{F}_q^{N\times t}$ is arbitrarily chosen by the adversary.

We also assume that $t < n$ (more precisely, we should assume $t < \mathsf{rank}\ A$); otherwise, the adversary may always choose $DZ = -AX$, leading to a trivial communications scenario.

The first assumption above allows us to use the approach of Section 4.1. The second assumption may seem somewhat "pessimistic," but it has the analytical advantage of completely eliminating from the problem the influence of the matrix F (recall that, in principle, D should be a submatrix of F).

The power of the approach of Section 4.1 lies in the fact that the channel model defined above can be *completely* described by the following discrepancy function

$$\Delta_A(X,Y) \triangleq \min_{\substack{r\in\mathbb{N},\, D\in\mathbb{F}_q^{N\times r},\, Z\in\mathbb{F}_q^{r\times m}:\\ Y=AX+DZ}} r. \tag{4.8}$$

The discrepancy $\Delta_A(X,Y)$ represents the minimum number of error packets that the adversary needs to inject in order to transform an input X into an output Y, given that the transfer matrix is A. The subscript in $\Delta_A(X,Y)$ is to emphasize the dependence on A. For this discrepancy function, the minimum-discrepancy decoder becomes

$$\hat{X} = \operatorname*{argmin}_{X\in\mathcal{C}} \Delta_A(X,Y). \tag{4.9}$$

Similarly, the Δ-distance induced by $\Delta_A(X, Y)$ is given by

$$\delta_A(X, X') \triangleq \min_{Y \in \mathbb{F}_q^{N \times m}} \Delta_A(X, Y) + \Delta_A(X', Y) \tag{4.10}$$

for $X, X' \in \mathbb{F}_q^{n \times m}$.

We now wish to find a simpler expression for $\Delta_A(X, Y)$ and $\delta_A(X, X')$, and show that $\Delta_A(X, Y)$ is normal.

Lemma 4.4:

$$\Delta_A(X, Y) = \mathsf{rank}\,(Y - AX). \tag{4.11}$$

Proof: Consider $\Delta_A(X, Y)$ as given by (4.8). For any feasible triple (s, D, Z), we have $s \geq \mathsf{rank}\, Z \geq \mathsf{rank}\, DZ = \mathsf{rank}\,(Y - AX)$. This bound is achievable by setting $s = \mathsf{rank}\,(Y - AX)$ and letting DZ be a full-rank decomposition of $Y - AX$. ∎

Lemma 4.5:

$$\delta_A(X, X') = d_{\mathrm{R}}(AX, AX') = \mathsf{rank}\, A(X' - X).$$

Proof: From (4.10) and Lemma 4.4, we have $\delta_A(X, X') = \min_Y d_{\mathrm{R}}(Y, AX) + d_{\mathrm{R}}(Y, AX')$. Since the rank metric satisfies the triangle inequality, we have $d_{\mathrm{R}}(AX, Y) + d_{\mathrm{R}}(AX', Y) \geq d_{\mathrm{R}}(AX, AX')$. This lower bound can be achieved by choosing, e.g., $Y = AX$. ∎

Note that $\delta_A(\cdot, \cdot)$ is a metric if and only if A has full column rank—in which case it is precisely the rank metric. (If $\mathsf{rank}\, A < n$, then there exist $X \neq X'$ such that $\delta_A(X, X') = 0$.)

Theorem 4.6: The discrepancy function $\Delta_A(\cdot, \cdot)$ is normal.

Proof: Let $X, X' \in \mathbb{F}_q^{n \times m}$ and let $0 \leq i \leq d = \delta_A(X, X')$. Then $\mathsf{rank}\, A(X' - X) = d$. By performing a full-rank decomposition of $A(X' - X)$, we can always find two matrices W and W' such that $W + W' = A(X' - X)$, $\mathsf{rank}\, W = i$ and $\mathsf{rank}\, W' = d - i$. Taking $Y = AX + W = AX' - W'$, we have that $\Delta_A(X, Y) = i$ and $\Delta_A(X', Y) = d - i$. ∎

Note that, under the discrepancy $\Delta_A(X, Y)$, a t-discrepancy-correcting code is a code that can correct *any* t packet errors injected by the adversary. Using Theorem 4.6 and Theorem 4.3, we have the following result.

Theorem 4.7: A code $\mathcal{C}$ is guaranteed to correct any t packet errors if and only if $\delta_A(\mathcal{C}) > 2t$.

Theorem 4.7 shows that $\delta_A(\mathcal{C})$ is indeed a fundamental parameter characterizing the error correction capability of a code in our model. Note that, if the condition of Theorem 4.7 is violated, then there exists at least one codeword for which the adversary can certainly induce a decoding failure.

Note that the error correction capability of a code $\mathcal{C}$ is dependent on the network code through the matrix A. Let $\rho = n - \mathsf{rank}\, A$ be the column-rank deficiency of A. Since $\delta_A(X, X') = \mathsf{rank}\, A(X' - X)$, it follows from (2.5) that

$$d_{\mathrm{R}}(X, X') - \rho \leq \delta_A(X, X') \leq d_{\mathrm{R}}(X, X')$$

and

$$d_{\mathrm{R}}(\mathcal{C}) - \rho \leq \delta_A(\mathcal{C}) \leq d_{\mathrm{R}}(\mathcal{C}). \tag{4.14}$$

Thus, the error correction capability of a code is strongly tied to its minimum rank distance; in particular, $\delta_A(\mathcal{C}) = d_{\mathrm{R}}(\mathcal{C})$ if $\rho = 0$. While the lower bound $\delta_A(\mathcal{C}) \geq d_{\mathrm{R}}(\mathcal{C}) - \rho$ may be not be tight in general, we should expect it to be tight when $\mathcal{C}$ is sufficiently large. This is indeed the case for MRD codes, as discussed in Section 4.2.3. Thus, a rank deficiency of A will typically reduce the error correction capability of a code.

Taking into account the worst case, we can use Theorem 4.12 to give a correction guarantee in terms of the minimum rank distance of the code.

Proposition 4.8: A code $\mathcal{C}$ is guaranteed to correct t packet errors, under rank deficiency ρ, if $d_{\mathrm{R}}(\mathcal{C}) > 2t + \rho$.

Note that the guarantee of Proposition 4.8 depends only on ρ and t; in particular, it is independent of the network code or the specific transfer matrix A.

4.2.2 Reinterpreting the Model of Yeung et al.

In this subsection, we investigate the model for coherent network coding studied by Yeung et al. in [10–12, 48], which is similar to the one considered in the previous subsection. The model is that of a matrix channel with input $X \in \mathbb{F}_q^{n\times m}$, output $Y \in \mathbb{F}_q^{N\times m}$, and channel law given by (3.2), where $A \in \mathbb{F}_q^{N\times n}$ and $F \in \mathbb{F}_q^{N\times|\mathcal{E}|}$ are fixed and known to the receiver, and $E \in \mathbb{F}_q^{|\mathcal{E}|\times m}$ is arbitrarily chosen by an adversary provided $\mathsf{wt}(E) \leq t$. In addition, the adversary has unlimited computational power and is omniscient, knowing, in particular, A, F and X.

We now show that some of the concepts defined in [48], such as "network Hamming distance," can be reinterpreted in the framework of Section 4.1. As a consequence, we can easily recover the results of [48] on error correction and detection guarantees.

First, note that the current model can be completely described by the following discrepancy function

$$\Delta_{A,F}(X,Y) \triangleq \min_{\substack{E\in\mathbb{F}_q^{|\mathcal{E}|\times m}:\\ Y=AX+FE}} \mathsf{wt}(E). \tag{4.15}$$

The Δ-distance induced by this discrepancy function is given by

$$\begin{aligned}
\delta_{A,F}(X_1,X_2) &\triangleq \min_{Y} \Delta_{A,F}(X_1,Y) + \Delta_{A,F}(X_2,Y) \\
&= \min_{\substack{Y,E_1,E_2:\\ Y=AX_1+FE_1\\ Y=AX_2+FE_2}} \mathsf{wt}(E_1)+\mathsf{wt}(E_2) \\
&= \min_{\substack{E_1,E_2:\\ A(X_2-X_1)=F(E_1-E_2)}} \mathsf{wt}(E_1)+\mathsf{wt}(E_2) \\
&= \min_{\substack{E:\\ A(X_2-X_1)=FE}} \mathsf{wt}(E)
\end{aligned}$$

where the last equality follows from the fact that $\mathsf{wt}(E_1-E_2) \leq \mathsf{wt}(E_1)+\mathsf{wt}(E_2)$,

achievable if $E_1 = 0$. The minimum Δ-distance is defined similarly as

$$\delta_{A,F}(\mathcal{C}) \triangleq \min_{X,X' \in \mathcal{C}:\, X \neq X'} \delta_{A,F}(X, X').$$

Let us now examine some of the concepts defined in [48]. For a specific sink node, the decoder proposed in [48, Eq. (2)] has the form

$$\hat{X} = \operatorname*{argmin}_{X \in \mathcal{C}} \Psi_{A,F}(X, Y).$$

The definition of the objective function $\Psi_{A,F}(X, Y)$ requires several other definitions presented in [48]. Specifically, $\Psi_{A,F}(X, Y) \triangleq D^{rec}(AX, Y)$, where $D^{rec}(Y_1, Y_2) \triangleq W^{rec}(Y_2 - Y_1)$, $W^{rec}(Y) \triangleq \min_{E \in \Upsilon(Y)} \mathsf{wt}(E)$, and $\Upsilon(Y) \triangleq \{E \colon Y = FE\}$. Substituting all these values into $\Psi_{A,F}(X, Y)$, we obtain

$$\begin{aligned}
\Psi_{A,F}(X, Y) &= D^{rec}(AX, Y) \\
&= W^{rec}(Y - AX) \\
&= \min_{E \in \Upsilon(Y - AX)} \mathsf{wt}(E) \\
&= \min_{E \colon Y - AX = FE} \mathsf{wt}(E) \\
&= \Delta_{A,F}(X, Y).
\end{aligned}$$

Thus, the decoder in [48] is precisely a minimum-discrepancy decoder.

In [48], the "network Hamming distance" between two messages X_1 and X_2 is defined as $D^{msg}(X_1, X_2) \triangleq W^{msg}(X_2 - X_1)$, where $W^{msg}(X) \triangleq W^{rec}(AX)$. Again, simply

substituting the corresponding definitions yields

$$\begin{aligned} D^{msg}(X_1, X_2) &= W^{msg}(X_2 - X_1) \\ &= W^{rec}(A(X_2 - X_1)) \\ &= \min_{E \in \Upsilon(A(X_2 - X_1))} \mathsf{wt}(E) \\ &= \min_{E:\, A(X_2 - X_1) = FE} \mathsf{wt}(E) \\ &= \delta_{A,F}(X_1, X_2). \end{aligned}$$

Thus, the "network Hamming distance" is precisely the Δ-distance induced by the discrepancy function $\Delta_{A,F}(X, Y)$. Finally, the "unicast minimum distance" of a network code with message set $\mathcal{C}$ [48] is precisely $\delta_{A,F}(\mathcal{C})$.

Let us return to the problem of characterizing the correction capability of a code.

Proposition 4.9: The discrepancy function $\Delta_{A,F}(\cdot, \cdot)$ is normal.

Proof: Let $X_1, X_2 \in \mathbb{F}_q^{n \times m}$ and let $0 \leq i \leq d = \delta_{A,F}(X_1, X_2)$. Let $E \in \mathbb{F}_q^{|\mathcal{E}| \times m}$ be a solution to the minimization in (4.15). Then $A(X_2 - X_1) = FE$ and $\mathsf{wt}(E) = d$. By partitioning E, we can always find two matrices E_1 and E_2' such that $E_1 + E_2' = E$, $\mathsf{rank}\ E_1 = i$ and $\mathsf{rank}\ E_2' = d - i$. Taking $Y = AX_1 + FE_1 = AX_2 - FE_2$, we have that $\Delta_{A,F}(X_1, Y) \leq i$ and $\Delta_{A,F}(X_2, Y) \leq d - i$. Since $d \leq \Delta_{A,F}(X_1, Y) + \Delta_{A,F}(X_2, Y)$, it follows that $\Delta_{A,F}(X_1, Y) = i$ and $\Delta_{A,F}(X_2, Y) = d - i$. ∎

It follows that a code $\mathcal{C}$ is guaranteed to correct any t packet errors if and only if $\delta_{A,F}(\mathcal{C}) > 2t$. Thus, we recover theorems 2 and 3 in [48] (for error detection, see Appendix A). The analogous results for the multicast case can be obtained in a straightforward manner.

We now wish to compare the parameters devised in this subsection with those of Section 4.2.1. From the description of the LNCC in Chapter 3, it is intuitive that the model of this subsection should be equivalent to that of the previous subsection if the

matrix F, rather than fixed and known to the receiver, is arbitrarily and secretly chosen by the adversary. A formal proof of this fact is given in the following proposition.

Proposition 4.10:

$$\Delta_A(X,Y) = \min_{F\in\mathbb{F}_q^{N\times|\mathcal{E}|}} \Delta_{A,F}(X,Y)$$

$$\delta_A(X,X') = \min_{F\in\mathbb{F}_q^{N\times|\mathcal{E}|}} \delta_{A,F}(X,X')$$

$$\delta_A(\mathcal{C}) = \min_{F\in\mathbb{F}_q^{N\times|\mathcal{E}|}} \delta_{A,F}(\mathcal{C}).$$

Proof: Consider the minimization

$$\min_{F\in\mathbb{F}_q^{N\times|\mathcal{E}|}} \Delta_{A,F}(X,Y) = \min_{\substack{F\in\mathbb{F}_q^{N\times|\mathcal{E}|},\, E\in\mathbb{F}_q^{|\mathcal{E}|\times m}:\\ Y=AX+FE}} \mathsf{wt}(E).$$

For any feasible (F,E), we have $\mathsf{wt}(E) \geq \mathsf{rank}\, E \geq \mathsf{rank}\, FE = \mathsf{rank}(Y-AX)$. This lower bound can be achieved by taking $F = \begin{bmatrix} F' & 0 \end{bmatrix}$ and $E = \begin{bmatrix} E' \\ 0 \end{bmatrix}$, where $F'E'$ is a full-rank decomposition of $Y-AX$. This proves the first statement. The second statement follows from the first by noticing that $\delta_A(X,X') = \Delta_A(X,AX')$ and $\delta_{A,F}(X,X') = \Delta_{A,F}(X,AX')$. The third statement is immediate. ∎

Proposition 4.10 shows that the model of Section 4.2.1 is indeed more pessimistic, as the adversary has additional power to choose the worst possible F. It follows that any code that is t-error-correcting for that model must also be t-error-correcting for the model of Yeung et al.

4.2.3 Optimality of MRD Codes

Let us now evaluate the performance of an MRD code under the models of the two previous subsections.

The Singleton bound of [11] (see also [51]) states that

$$|\mathcal{C}| \leq Q^{n-\rho-\delta_{A,F}(\mathcal{C})+1} \tag{4.19}$$

where Q is the size of the alphabet[2] from which packets are drawn. Note that $Q = q^m$ in our setting, since each packet consists of m symbols from $\mathbb{F}_q$. Using Proposition 4.10, we can also obtain

$$|\mathcal{C}| \leq Q^{n-\rho-\delta_A(\mathcal{C})+1} = q^{m(n-\rho-\delta_A(\mathcal{C})+1)}. \tag{4.20}$$

On the other hand, the size of an MRD code, for $m \geq n$, is given by

$$\begin{aligned} |\mathcal{C}| &= q^{m(n-d_{\mathrm{R}}(\mathcal{C})+1)} && (4.21)\\ &\geq q^{m(n-\rho-\delta_A(\mathcal{C})+1)} && (4.22)\\ &\geq q^{m(n-\rho-\delta_{A,F}(\mathcal{C})+1)} \end{aligned}$$

where (4.22) follows from (4.14). Since $Q = q^m$, both (4.19) and (4.20) are achieved in this case. Thus, we have the following result.

Theorem 4.11: When $m \geq n$, an MRD code $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ achieves maximum cardinality with respect to both δ_A and $\delta_{A,F}$.

Theorem 4.11 shows that, if an alphabet of size $Q = q^m \geq q^n$ is allowed (i.e., a packet size of at least $n \log_2 q$ bits), then MRD codes turn out to be optimal under both models of sections 4.2.1 and 4.2.2.

Remark: It is straightforward to extend the results of Section 4.2.1 for the case of multiple heterogeneous receivers, where each receiver u experiences a rank deficiency $\rho^{(u)}$. In this case, it can be shown that an MRD code with $m \geq n$ achieves the refined Singleton bound of [51]. □

Note that, due to (4.20), (4.21) and (4.22), it follows that $\delta_A(\mathcal{C}) = d_{\mathrm{R}}(\mathcal{C}) - \rho$ for an MRD code with $m \geq n$. Thus, in this case, we can restate Theorem 4.7 in terms of the minimum rank distance of the code.

[2]This alphabet is usually assumed a finite field, but, for the Singleton bound of [11], it is sufficient to assume an abelian group, e.g., a vector space over $\mathbb{F}_q$.

Theorem 4.12: An MRD code $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ with $m \geq n$ is guaranteed to correct t packet errors, under rank deficiency ρ, if and only if $d_{\mathrm{R}}(\mathcal{C}) > 2t + \rho$.

Observe that Theorem 4.12 holds regardless of the specific transfer matrix A, depending only on its column-rank deficiency ρ.

The results of this section imply that, when designing a linear network code, we may focus solely on the objective of making the network code feasible, i.e., maximizing $\mathsf{rank}\ A$. If an error correction guarantee is desired, then an outer code can be applied end-to-end without requiring any modifications on (or even knowledge of) the underlying network code. The design of the outer code is essentially trivial, as any MRD code can be used, with the only requirement that the number of $\mathbb{F}_q$-symbols per packet, m, is at least n.

4.3 Noncoherent Network Coding

4.3.1 A Worst-Case Model and the Injection Metric

Our model for noncoherent network coding with adversarial errors differs from its coherent counterpart of Section 4.2.1 only with respect to the transfer matrix A. Namely, the matrix A is unknown to the receiver and is freely chosen by the adversary while respecting the constraint $\mathsf{rank}\ A \geq n - \rho$. The parameter ρ, the maximum column rank deficiency of A, is a parameter of the system that is known to all. Note that, as discussed above for the matrix D, the assumption that A is chosen by the adversary is what provides the conservative (worst-case) nature of the model. The constraint on the rank of A is required for a meaningful coding problem; otherwise, the adversary could prevent communication by simply choosing $A = 0$.

As before, we assume a minimum-discrepancy decoder

$$\hat{X} = \operatorname*{argmin}_{X\in\mathcal{C}} \Delta_\rho(X, Y) \tag{4.23}$$

with discrepancy function given by

$$\Delta_\rho(X,Y) \triangleq \min_{\substack{A\in\mathbb{F}_q^{N\times n}, r\in\mathbb{N}, D\in\mathbb{F}_q^{N\times r}, Z\in\mathbb{F}_q^{r\times m}: \\ Y=AX+DZ \\ \mathsf{rank}\ A\geq n-\rho}} r \tag{4.24}$$

$$= \min_{\substack{A\in\mathbb{F}_q^{N\times n}: \\ \mathsf{rank}\ A\geq n-\rho}} \Delta_A(X,Y).$$

Again, $\Delta_\rho(X,Y)$ represents the minimum number of error packets needed to produce an output Y given an input X under the current adversarial model. The subscript is to emphasize that $\Delta_\rho(X,Y)$ is still a function of ρ.

The Δ-distance induced by $\Delta_\rho(X,Y)$ is defined below. For $X, X' \in \mathbb{F}_q^{n\times m}$, let

$$\delta_\rho(X,X') \triangleq \min_{Y\in\mathbb{F}_q^{N\times m}} \Delta_\rho(X,Y) + \Delta_\rho(X',Y). \tag{4.25}$$

We now prove that $\Delta_\rho(X,Y)$ is normal and therefore $\delta_\rho(\mathcal{C})$ characterizes the correction capability of a code.

First, observe that, using Lemma 4.4, we may rewrite $\Delta_\rho(X,Y)$ as

$$\Delta_\rho(X,Y) = \min_{\substack{A\in\mathbb{F}_q^{N\times n}: \\ \mathsf{rank}\ A\geq n-\rho}} \mathsf{rank}(Y-AX). \tag{4.26}$$

Also, note that

$$\begin{aligned}
\delta_\rho(X,X') &= \min_Y \Delta_\rho(X,Y) + \Delta_\rho(X',Y) \\
&= \min_{\substack{A,A'\in\mathbb{F}_q^{N\times n}: \\ \mathsf{rank}\ A\geq n-\rho \\ \mathsf{rank}\ A'\geq n-\rho}} \min_Y d_{\mathrm{R}}(AX,Y) + d_{\mathrm{R}}(A'X',Y) \\
&= \min_{\substack{A,A'\in\mathbb{F}_q^{N\times n}: \\ \mathsf{rank}\ A\geq n-\rho \\ \mathsf{rank}\ A'\geq n-\rho}} d_{\mathrm{R}}(AX,A'X')
\end{aligned} \tag{4.27}$$

where the last equality follows from the fact that $d_{\mathrm{R}}(AX,Y)+d_{\mathrm{R}}(A'X',Y) \geq d_{\mathrm{R}}(AX,A'X')$, achievable by choosing, e.g., $Y = AX$.

Theorem 4.13: The discrepancy function $\Delta_\rho(\cdot,\cdot)$ is normal.

Proof: Let $X, X' \in \mathbb{F}_q^{n\times m}$ and let $0 \le i \le d = \delta_\rho(X, X')$. Let $A, A' \in \mathbb{F}_q^{N\times n}$ be a solution to the minimization in (4.27). Then $\mathsf{rank}(A'X' - AX) = d$. By performing a full-rank decomposition of $A'X' - AX$, we can always find two matrices W and W' such that $W+W' = A'X'-AX$, $\mathsf{rank}\ W = i$ and $\mathsf{rank}\ W' = d-i$. Taking $Y = AX+W = AX'-W'$, we have that $\Delta_\rho(X, Y) \le i$ and $\Delta_\rho(X', Y) \le d - i$. Since $d \le \Delta_\rho(X, Y) + \Delta_\rho(X', Y)$, it follows that $\Delta_\rho(X, Y) = i$ and $\Delta_\rho(X', Y) = d - i$. ∎

As a consequence of Theorem 4.13, we have the following result.

Theorem 4.14: A code $\mathcal{C}$ is guaranteed to correct any t packet errors if and only if $\delta_\rho(\mathcal{C}) > 2t$.

Similarly as in Section 4.2.1, Theorem 4.14 shows that $\delta_\rho(\mathcal{C})$ is a fundamental parameter characterizing the error correction capability of a code in the current model. In contrast to Section 4.2.1, however, the expression for $\Delta_\rho(X, Y)$ (and, consequently, $\delta_\rho(X, X')$) does not seem mathematically appealing since it involves a minimization. We now proceed to finding simpler expressions for $\Delta_\rho(X, Y)$ and $\delta_\rho(X, X')$.

The minimization in (4.26) is a special case of a more general expression, which we give as follows. For $X \in \mathbb{F}_q^{n\times m}$, $Y \in \mathbb{F}_q^{N\times m}$ and $L \ge \max\{n - \rho,\ N - \sigma\}$, let

$$\Delta_{\rho,\sigma,L}(X, Y) \triangleq \min_{\substack{A\in\mathbb{F}_q^{L\times n}, B\in\mathbb{F}_q^{L\times N}: \\ \mathsf{rank}\ A\ge n-\rho \\ \mathsf{rank}\ B\ge N-\sigma}} \mathsf{rank}(BY - AX).$$

The quantity defined above is computed in the following lemma.

Lemma 4.15:

$$\Delta_{\rho,\sigma,L}(X, Y) = \Big[\max\{\mathsf{rank}\ X - \rho,\ \mathsf{rank}\ Y - \sigma\} - \mathsf{dim}\left(\langle X\rangle \cap \langle Y\rangle\right)\Big]^+.$$

Proof: See Appendix B.1. ∎

Note that $\Delta_{\rho,\sigma,L}(X,Y)$ is independent of L, for all valid L. Thus, we may drop the subscript and write simply $\Delta_{\rho,\sigma}(X,Y) \triangleq \Delta_{\rho,\sigma,L}(X,Y)$.

We can now provide a simpler expression for $\Delta_\rho(X,Y)$.

Theorem 4.16:

$$\Delta_\rho(X,Y) = \max\{\mathsf{rank}\ X - \rho,\ \mathsf{rank}\ Y\} - \mathsf{dim}\,(\langle X\rangle \cap \langle Y\rangle).$$

Proof: This follows immediately from Lemma 4.15 by noticing that $\Delta_\rho(X,Y) = \Delta_{\rho,0}(X,Y)$.■

From Theorem 4.16, we observe that $\Delta_\rho(X,Y)$ depends on the matrices X and Y only through their row spaces, i.e., only the transmitted and received row spaces have a role in the decoding. Put another way, we may say that the channel really accepts an input subspace $\langle X\rangle$ and delivers an output subspace $\langle Y\rangle$. Thus, all the communication is made via subspace selection. This observation provides a fundamental justification for the approach of [18].

At this point, it is useful to introduce the following definition.

Definition 4.2: The *injection distance* between subspaces $\mathcal{U}$ and $\mathcal{V}$ in $\mathcal{P}(\mathbb{F}_q^m)$ is defined as

$$\begin{aligned} d_{\mathrm{I}}(\mathcal{U},\mathcal{V}) &\triangleq \max\{\mathsf{dim}\,\mathcal{U},\ \mathsf{dim}\,\mathcal{V}\} - \mathsf{dim}\,(\mathcal{U}\cap\mathcal{V}) \\ &= \mathsf{dim}\,(\mathcal{U}+\mathcal{V}) - \min\{\mathsf{dim}\,\mathcal{U},\ \mathsf{dim}\,\mathcal{V}\}. \end{aligned} \tag{4.31}$$

The *minimum injection distance* of a set $\Omega \subseteq \mathcal{P}(\mathbb{F}_q^m)$, denoted $d_{\mathrm{I}}(\Omega)$, is the minimum injection distance between all pairs of distinct elements in Ω.

The injection distance can be interpreted as measuring the number of error packets that an adversary needs to inject in order to transform an input subspace $\langle X\rangle$ into an output subspace $\langle Y\rangle$. This can be clearly seen from the fact that $d_{\mathrm{I}}(\langle X\rangle, \langle Y\rangle) = \Delta_0(X,Y)$. Thus, the injection distance is essentially equal to the discrepancy $\Delta_\rho(X,Y)$

when the channel is influenced only by the adversary, i.e., when the non-adversarial aspect of the channel (the column-rank deficiency of A) is removed from the problem. Note that, in this case, the decoder (4.23) becomes precisely a minimum-injection-distance decoder.

Proposition 4.17: The injection distance is a metric.

We delay the proof of Proposition 4.17 until Section 4.3.2.

We can now use the definition of the injection distance to simplify the expression for the Δ-distance.

Proposition 4.18:

$$\delta_\rho(X, X') = [d_{\mathrm{I}}(\langle X \rangle, \langle X' \rangle) - \rho]^+.$$

Proof: This follows immediately after realizing that $\delta_\rho(X, X') = \Delta_{\rho,\rho}(X, X')$. ∎

From Proposition 4.18, it is clear that $\delta_\rho(\cdot, \cdot)$ is a metric if and only if $\rho = 0$ (in which case it is precisely the injection metric). If $\rho > 0$, then $\delta_\rho(\cdot, \cdot)$ does not satisfy the triangle inequality.

It is worth noticing that $\delta_\rho(X, X') = 0$ for any two matrices X and X' that share the same row space. Thus, any reasonable code $\mathcal{C}$ should avoid this situation.

For $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$, let

$$\langle \mathcal{C} \rangle = \{\langle X \rangle : X \in \mathcal{C}\}$$

be the subspace code (i.e., a collection of subspaces) consisting of the row spaces of all matrices in $\mathcal{C}$. The following corollary of Proposition 4.18 is immediate.

Corollary 4.19: Suppose $\mathcal{C}$ is such that $|\mathcal{C}| = |\langle \mathcal{C} \rangle|$, i.e., no two codewords of $\mathcal{C}$ have the same row space. Then

$$\delta_\rho(\mathcal{C}) = [d_{\mathrm{I}}(\langle \mathcal{C} \rangle) - \rho]^+.$$

Using Corollary 4.19, we can restate Theorem 4.14 more simply in terms of the injection distance.

Theorem 4.20: A code $\mathcal{C}$ is guaranteed to correct t packet errors, under rank deficiency ρ, if and only if $d_{\mathrm{I}}(\langle\mathcal{C}\rangle) > 2t + \rho$.

Note that, due to equality in Corollary 4.19, a converse is indeed possible in Theorem 4.20 (contrast with Proposition 4.8 for the coherent case).

Theorem 4.20 shows that $d_{\mathrm{I}}(\langle\mathcal{C}\rangle)$ is a fundamental parameter characterizing the *complete* correction capability (i.e., error correction capability and "rank-deficiency correction" capability) of a code in our noncoherent model. Put another way, we may say that a code $\mathcal{C}$ is good for the model of this subsection if and only if its subspace version $\langle\mathcal{C}\rangle$ is a good code in the injection metric.

4.3.2 Comparison with the Metric of Kötter and Kschischang

Let $\Omega \subseteq \mathcal{P}_n^{\max}(\mathbb{F}_q^m)$ be a subspace code whose elements have maximum dimension n. In [18], the network is modeled as an operator channel that takes in a subspace $\mathcal{V} \in \mathcal{P}(\mathbb{F}_q^m)$ and puts out a possibly different subspace $\mathcal{U} \in \mathcal{P}(\mathbb{F}_q^m)$. The kind of disturbance that the channel applies to $\mathcal{V}$ is captured by the notions of "insertions" and "deletions" of dimensions (represented mathematically using operators), and the degree of such a dissimilarity is captured by the subspace distance

$$\begin{aligned} d_{\mathrm{S}}(\mathcal{V},\mathcal{U}) &\triangleq \dim(\mathcal{V}+\mathcal{U}) - \dim(\mathcal{V}\cap\mathcal{U}) \\ &= \dim\mathcal{V} + \dim\mathcal{U} - 2\dim(\mathcal{V}\cap\mathcal{U}) \\ &= \dim(\mathcal{V}+\mathcal{U}) - \dim\mathcal{V} - \dim\mathcal{U}. \end{aligned} \tag{4.35}$$

The transmitter selects some $\mathcal{V} \in \Omega$ and transmits $\mathcal{V}$ over the channel. The receiver receives some subspace $\mathcal{U}$ and, using a *minimum subspace distance decoder*, decides that

the subspace $\hat{\mathcal{V}} \subseteq \Omega$ was sent, where

$$\hat{\mathcal{V}} = \underset{\mathcal{V} \in \Omega}{\operatorname{argmin}} \, d_{\mathrm{S}}(\mathcal{V}, \mathcal{U}). \tag{4.36}$$

This decoder is guaranteed to correct all disturbances applied by the channel if $d_{\mathrm{S}}(\mathcal{V}, \mathcal{U}) < d_{\mathrm{S}}(\Omega)/2$, where $d_{\mathrm{S}}(\Omega)$ is the minimum subspace distance between all pairs of distinct codewords of Ω.

First, let us point out that this setup is indeed the same as that of Section 4.3.1 if we set $\mathcal{V} = \langle X \rangle$, $\mathcal{U} = \langle Y \rangle$ and $\Omega = \langle \mathcal{C} \rangle$, where $\mathcal{C}$ is such that $|\mathcal{C}| = |\langle \mathcal{C} \rangle|$. Also, any disturbance applied by an operator channel can be realized by a matrix model, and vice-versa. Thus, the difference between the approach of this section and that of [18] lies in the choice of the decoder.

Indeed, by using Theorem 4.16 and the definition of subspace distance, we get the following relationship:

Proposition 4.21:

$$\Delta_\rho(X, Y) = \frac{1}{2} d_{\mathrm{S}}(\langle X \rangle, \langle Y \rangle) - \frac{1}{2}\rho + \frac{1}{2} |\operatorname{\mathsf{rank}} X - \operatorname{\mathsf{rank}} Y - \rho|.$$

Thus, we can see that when the matrices in $\mathcal{C}$ do not all have the same rank (i.e., Ω is a *non-constant-dimension code*), then the decoding rules (4.23) and (4.36) may produce different decisions.

Using $\rho = 0$ in the above proposition (or simply using (4.31) and (4.35)) gives us another formula for the injection distance:

$$d_{\mathrm{I}}(\mathcal{V}, \mathcal{U}) = \frac{1}{2} d_{\mathrm{S}}(\mathcal{V}, \mathcal{U}) + \frac{1}{2} |\operatorname{\mathsf{dim}} \mathcal{V} - \operatorname{\mathsf{dim}} \mathcal{U}|. \tag{4.38}$$

We can now prove a result that was postponed in the previous section.

Theorem 4.22: The injection distance is a metric.

Proof: Since $d_{\mathrm{S}}(\cdot,\cdot)$ is a metric on $\mathcal{P}(\mathbb{F}_q^m)$ and $|\cdot|$ is a norm on $\mathbb{R}$, it follows from (4.38) that $d_{\mathrm{I}}(\cdot,\cdot)$ is also a metric on $\mathcal{P}(\mathbb{F}_q^m)$. ■

We now examine in more detail an example situation where the minimum-subspace-distance decoder and the minimum-discrepancy decoder produce different decisions.

Example 4.4: For simplicity, assume $\rho = 0$. Consider a subspace code that contains two codewords $\mathcal{V}_1 = \langle X_1 \rangle$ and $\mathcal{V}_2 = \langle X_2 \rangle$ such that $\gamma \triangleq \mathsf{dim}\ \mathcal{V}_2 - \mathsf{dim}\ \mathcal{V}_1$ satisfies $d/3 < \gamma < d/2$, where $d \triangleq d_{\mathrm{S}}(\mathcal{V}_1, \mathcal{V}_2)$.

Suppose the received subspace $\mathcal{U} = \langle Y \rangle$ is such that $\mathcal{V}_1 \subseteq \mathcal{U} \subseteq \mathcal{V}_1 + \mathcal{V}_2$ and $\mathsf{dim}\,\mathcal{U} = \mathsf{dim}\ \mathcal{V}_1 + \gamma = \mathsf{dim}\ \mathcal{V}_2$, as illustrated in Fig. 4.1. Then $d_{\mathrm{S}}(\mathcal{V}_1, \mathcal{U}) = \gamma$ and $d_{\mathrm{S}}(\mathcal{V}_2, \mathcal{U}) = d - \gamma$, while Proposition (4.21) gives $\Delta_\rho(X_1, Y) = \gamma$ and $\Delta_\rho(X_2, Y) = (d-\gamma)/2 \triangleq \epsilon$. Since, by assumption, $d - \gamma > \gamma$ and $\epsilon < \gamma$, it follows that $d_{\mathrm{S}}(\mathcal{V}_1, \mathcal{U}) < d_{\mathrm{S}}(\mathcal{V}_2, \mathcal{U})$ but $\Delta_\rho(X_1, Y) > \Delta_\rho(X_2, Y)$, i.e., the decoders (4.36) and (4.23) will produce different decisions.

This situation can be intuitively explained as follows. The decoder (4.36) favors the subspace $\mathcal{V}_1$, which is closer in subspace distance to $\mathcal{U}$ than $\mathcal{V}_2$. However, since $\mathcal{V}_1$ is low-dimensional, $\mathcal{U}$ can only be produced from $\mathcal{V}_1$ by the *insertion* of γ dimensions. The decoder (4.23), on the other hand, favors $\mathcal{V}_2$, which, although farther in subspace distance, can produce $\mathcal{U}$ after the *replacement* of $\epsilon < \gamma$ dimensions. Since one packet error must occur for each inserted or replaced dimension, we conclude that the decoder (4.23) finds the solution that minimizes the number of packet errors observed. □

Remark: The subspace metric of [18] treats insertions and deletions of dimensions (called in [18] "errors" and "erasures", respectively) symmetrically. However, depending upon the position of the adversary in the network (namely, if there is a source-destination min-cut between the adversary and the destination) then a single error packet may cause the replacement of a dimension (i.e., a simultaneous "error" and "erasure" in the terminology of [18]). The injection distance, which is designed to "explain" a received subspace with as few error-packet injections as possible, properly accounts for this phenomenon,

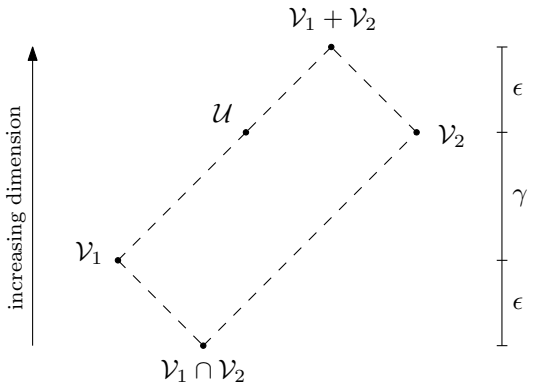

Figure 4.1: Lattice of subspaces in Example 4.4. Two spaces are joined with a dashed line if one is a subspace of the other.

and hence the corresponding decoder produces a different result than a minimum subspace distance decoder. If it were possible to restrict the adversary so that error-packet injections would cause only *insertion* of dimensions, then the subspace distance of [18] would indeed be appropriate. However, this is not the model considered here. □

Let us now discuss an important fact about the subspace distance for general subspace codes (assuming for simplicity that $\rho = 0$). The packet error correction capability of a minimum-subspace-distance decoder, t_S, is not necessarily equal to $\lfloor (d_{\mathrm{S}}(\mathcal{C}) - 1)/2 \rfloor$ or $\lfloor (d_{\mathrm{S}}(\mathcal{C}) - 2)/4 \rfloor$, but lies somewhere in between. For instance, in the case of a constant-dimension code Ω, we have

$$d_{\mathrm{I}}(\mathcal{V}, \mathcal{V}') = \frac{1}{2} d_{\mathrm{S}}(\mathcal{V}, \mathcal{V}'), \quad \forall \mathcal{V}, \mathcal{V}' \in \Omega,$$
$$d_{\mathrm{I}}(\Omega) = \frac{1}{2} d_{\mathrm{S}}(\Omega).$$

Thus, Theorem 4.20 implies that $t_S = \lfloor (d_{\mathrm{S}}(\mathcal{C}) - 2)/4 \rfloor$ exactly. In other words, in this special case, the approach in [18] coincides with that of this chapter, and Theorem 4.20 provides a converse that was missing in [18]. On the other hand, suppose Ω is a subspace code consisting of just two codewords, one of which is a subspace of the other. Then we have precisely $t_S = \lfloor (d_{\mathrm{S}}(\mathcal{C}) - 1)/2 \rfloor$, since $t_S + 1$ packet-injections are needed to get past halfway between the codewords.

Since no single quantity is known that perfectly describes the packet error correction capability of the minimum-subspace-distance decoder (4.36) for general subspace codes, we cannot provide a definitive comparison between decoders (4.36) and (4.23). However, we can still compute bounds for codes that fit into Example 4.4.

Example 4.5: Let us continue with Example 4.4. Now, we adjoin another codeword $\mathcal{V}_3 = \langle X_3 \rangle$ such that $d_{\mathrm{S}}(\mathcal{V}_1, \mathcal{V}_3) = d$ and where $\gamma' \triangleq \mathsf{dim}\ \mathcal{V}_3 - \mathsf{dim}\ \mathcal{V}_1$ satisfies $d/3 < \gamma' < d/2$. Also we assume that $d_{\mathrm{S}}(\mathcal{V}_2, \mathcal{V}_3)$ is sufficiently large so as not to interfere with the problem (e.g., $d_{\mathrm{S}}(\mathcal{V}_2, \mathcal{V}_3) > 3d/2$).

Let t_{S} and t_{M} denote the packet error correction capabilities of the decoders (4.36) and (4.23), respectively. From the argument of Example 4.4, we get $t_{\mathrm{M}} \geq \max\{\epsilon, \epsilon'\}$, while $t_{\mathrm{S}} < \min\{\epsilon, \epsilon'\}$, where $\epsilon' = (d - \gamma')/2$. By choosing $\gamma \approx d/3$ and $\gamma' \approx d/2$, we get $\epsilon \approx d/3$ and $\epsilon' \approx d/4$. Thus, $t_{\mathrm{M}} \geq (4/3)t_{\mathrm{S}}$, i.e., we obtain a $1/3$ increase in error correction capability by using the decoder (4.23). □

4.3.3 Near-Optimality of Liftings of MRD Codes

Unlike the coherent case, where optimal codes can be easily constructed, it seems a hard combinatorial problem to achieve optimality for noncoherent network coding. So far, no nontrivial optimal codes are known for either the injection or the subspace metric. Even in the constant-dimension case, only very simple optimal codes (with very low rate) are known [52]. Nevertheless, it is a natural question to ask whether MRD codes can be adapted to noncoherent network coding, and if so, what their relative performance is.

In this subsection, we propose a class of subspace codes that can be easily constructed from MRD codes and achieve nearly optimal performance.

Lifting Construction

Assume $m = n + m'$, where $m' \geq 0$. Let $I = I_{n \times n}$.

Definition 4.3: Let the function $\Pi\colon \mathbb{F}_q^{n\times m'} \to \mathcal{P}(\mathbb{F}_q^{n+m'})$ be given by $x \mapsto \Pi(x) = \langle [I \quad x] \rangle$. The subspace $\Pi(x)$ is called the *lifting* of the matrix x. Similarly, define the *lifting* of the matrix code $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m'}$, denoted $\Pi(\mathcal{C})$, as the subspace code obtained by lifting each codeword of $\mathcal{C}$.

Definition 4.3 provides an injective mapping between matrix codes and subspace codes. Note that a subspace code constructed by lifting is always a constant-dimension code, with codeword dimension n.

The next proposition shows that rank-metric codes are a natural choice of codes in which to apply the lifting construction.

Proposition 4.23: Let $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m'}$ and $x, x' \in \mathcal{C}$. Then

$$d_{\mathrm{I}}(\Pi(x), \Pi(x')) = d_{\mathrm{R}}(x, x')$$
$$d_{\mathrm{I}}(\Pi(\mathcal{C})) = d_{\mathrm{R}}(\mathcal{C}).$$

Proof: We have

$$\begin{aligned} d_{\mathrm{I}}(\Pi(x), \Pi(x')) &= \mathsf{dim}\,(\Pi(x) + \Pi(x')) - \min\{\mathsf{dim}\ \Pi(x),\ \mathsf{dim}\ \Pi(x')\} \\ &= \mathsf{rank} \begin{bmatrix} I & x \\ I & x' \end{bmatrix} - n \\ &= \mathsf{rank} \begin{bmatrix} I & x \\ 0 & x' - x \end{bmatrix} - n \\ &= \mathsf{rank}\,(x' - x). \end{aligned}$$

The second statement is immediate. ∎

Proposition 4.23 shows that a subspace code constructed by lifting inherits the distance properties of its underlying rank-metric code. It follows that a subspace code

$\Omega \subseteq \mathcal{P}_n(\mathbb{F}_q^{n+m'})$ with constant dimension n, minimum injection distance d and cardinality

$$|\Omega| = q^{\max\{n,m'\}(\min\{n,m'\}-d+1)} \tag{4.39}$$

can be constructed as the lifting $\Omega = \Pi(\mathcal{C})$ of an MRD code $\mathcal{C} \in \mathbb{F}_q^{n\times m'}$ with minimum rank distance d.

In this context, it is worth mentioning that the Reed-Solomon-like constant-dimension codes proposed in [18] correspond exactly to the lifting of Gabidulin codes.

Performance

We now investigate whether such liftings of MRD codes are really "good" in terms of the injection metric. We start with some results on the size of optimal subspace codes.

Let $A_q(m,d,n)$ denote the size of a largest subspace code in $\mathcal{P}_n^{\max}(\mathbb{F}_q^m)$ with minimum injection distance d. Similarly, let $B_q(m,d,n)$ denote the size of a largest constant-dimension code in $\mathcal{P}_n(\mathbb{F}_q^m)$ with minimum injection distance d.

It is shown in [53] that, for $1 \leq d \leq n$,

$$B_q(m,d,n) \leq \frac{\begin{bmatrix} m \\ n-d+1 \end{bmatrix}}{\begin{bmatrix} n \\ n-d+1 \end{bmatrix}}.$$

It follows from (2.12) that, for $1 \leq d \leq n$,

$$B_q(m,d,n) \leq 4q^{(m-n)(n-d+1)}.$$

Note that $B_q(m,d,n) = 1$ for $d > n$.

A very simple bound on $A_q(m,d,n)$ can be obtained by noticing that $\mathcal{P}_n^{\max}(\mathbb{F}_q^m) = \mathcal{P}_n(\mathbb{F}_q^m) \cup \mathcal{P}_{n-1}^{\max}(\mathbb{F}_q^m)$. Thus,

$$A_q(m,d,n) \leq B_q(m,d,n) + A_q(m,d,n-1).$$

Iterating the bound, and using the fact that $B_q(m,d,k)$ is nondecreasing for $0 \le k \le m/2$, we have, for $n \le m/2$,

$$\begin{aligned} A_q(m,d,n) &\le B_q(m,d,n) + B_q(m,d,n-1) + \cdots + B_q(m,d,0) \\ &\le B_q(m,d,n) + nB_q(m,d,n-1) \\ &\le 4q^{(m-n)(n-d+1)} + 4nq^{(m-n+1)(n-d)}. \end{aligned} \tag{4.43}$$

For a subspace code $\Omega \subseteq \mathcal{P}_n^{\max}(\mathbb{F}_q^m)$ with $d_{\mathrm{I}}(\Omega) = d$, let

$$\alpha(\Omega) \triangleq \frac{\log_q A_q(m,d,n) - \log_q |\Omega|}{\log_q A_q(m,d,n)}$$

denote the *rate loss* incurred when using Ω rather than an optimal subspace code.

The following theorem shows that liftings of MRD codes are asymptotically optimal for large packet sizes.

Theorem 4.24: Let $\Omega \subseteq \mathcal{P}_n^{\max}(\mathbb{F}_q^m)$ be the lifting of an MRD code with $m \ge 2n$ and $d_{\mathrm{I}}(\Omega) = d$. Then

$$\alpha(\Omega) \le \frac{1}{(m-n)(n-d+1)} \left(\frac{2}{\log_2 q} + \frac{n}{q^{m-2n+d} \ln q} \right) \le \frac{4}{P} + \frac{1}{2^{P/2} \ln 2}$$

where $P = m \log_2 q$ is the packet size in bits.

Proof: We have

$$\begin{aligned}
\alpha(\Omega) &\leq \frac{\log_q A_q(m,d,n) - \log_q |\Omega|}{\log_q |\Omega|} \\
&= \frac{1}{\log_q |\Omega|} \log_q \left(\frac{A_q(m,d,n)}{|\Omega|} \right) \\
&\leq \frac{1}{\log_q |\Omega|} \log_q \left(\frac{4q^{(m-n)(n-d+1)} + 4nq^{(m-n+1)(n-d)}}{|\Omega|} \right) && (4.46) \\
&= \frac{1}{(m-n)(n-d+1)} \left(\log_q 4 + \log_q \left(1 + \frac{nq^{(m-n+1)(n-d)}}{q^{(m-n)(n-d+1)}} \right) \right) && (4.47) \\
&= \frac{1}{(m-n)(n-d+1)} \left(\log_q 4 + \log_q \left(1 + nq^{2n-m-d} \right) \right) \\
&\leq \frac{1}{(m-n)(n-d+1)} \left(\log_q 4 + \frac{nq^{2n-m-d}}{\ln q} \right) && (4.48) \\
&\leq \frac{1}{(m-n)(n-d+1)} \left(\frac{2}{\log_2 q} + \frac{n}{q^{m-2n+d} \ln q} \right) && (4.49)
\end{aligned}$$

where (4.46) follows from (4.39), (4.47) follows from (4.43), and (4.48) follows from the fact that $\ln(1+x) \leq x$. Let $f_n(d)$ denote the RHS of (4.48). Since $f_n(d)$ is a convex function of d for $1 \leq d \leq n$, it is maximized at one of the boundaries, i.e., either $f_n(1)$ or $f_n(n)$. Let $L = \max_{1 \leq n \leq m/2} f_n(1)$ and $U = \max_{1 \leq n \leq m/2} f_n(n)$. We now show that $U \geq L$.

Clearly, since $f_n(n)$ is an increasing function of n, it is maximized at $n = m/2$. We have

$$U = f_{m/2}(m/2) = \frac{1}{m/2} \left(\frac{2}{\log_2 q} + \frac{m/2}{q^{m/2} \ln q} \right) = \frac{4}{P} + \frac{1}{2^{P/2} \ln q}$$

Since $m = 2, 3$ forces $L = f_1(1) = U$, it suffices to assume $m \geq 4$. A simple evaluation

shows that $L = \max\{f_1(1), f_2(1)\} < U$ for $m = 4$, so we assume that $m \geq 5$. We have

$$\begin{aligned} f_n(1) &= \frac{2}{(m-n)n\log_2 q} + \frac{1}{(m-n)q^{m-2n+1}\ln q} \\ &\leq \frac{2}{(m-n)n\log_2 q} + \frac{1}{(m/2)q\ln q} \\ &\leq \frac{2}{(m-1)\log_2 q} + \frac{1}{(m/2)q\ln q} \\ &= \frac{2m}{(m-1)P} + \frac{2}{Pq\ln 2} \\ &\leq \frac{1}{P}\left(\frac{2m}{(m-1)} + \frac{1}{\ln 2}\right) \leq \frac{4}{P} \quad \text{for } m \geq 5. \end{aligned}$$

Thus, $L \leq U$ for all m. It follows that

$$\alpha(\Omega) \leq U \leq \frac{4}{P} + \frac{1}{2^{P/2}\ln 2}.$$ ■

Theorem 4.24 shows that, for all practical purposes, liftings of MRD codes are essentially optimal. Indeed, for packet sizes of more than 400 bits, the rate loss is smaller than 1%.

4.4 Equivalence of Coherent and Noncoherent Decoding Problems

We have seen in Section 4.2.1 that, for coherent network coding, the minimum-discrepancy decoding problem is given by

$$\hat{X}_{\mathsf{coh}} = \operatorname*{argmin}_{X \in \mathcal{C}} \mathsf{rank}\,(Y_{\mathsf{coh}} - A_{\mathsf{coh}} X) \tag{4.52}$$

where $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ is the code, $A_{\mathsf{coh}} \in \mathbb{F}_q^{N\times n}$ is the (known) transfer matrix, and $Y_{\mathsf{coh}} \in \mathbb{F}_q^{N\times m}$ is the received matrix.

For noncoherent network coding, the minimum-discrepancy decoding problem has, in general, quite a different nature. However, consider the case of a lifted code. Specifically, let $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ be a rank-metric code, and suppose that each transmitted matrix X has the

form $X = \begin{bmatrix} I & x \end{bmatrix}$, where $x \in \mathcal{C}$. (*Note: For notational convenience, we are assuming now that each packet consists of $n+m$ symbols, rather than m symbols as before.*) Because the subspace code $\Pi(\mathcal{C})$ is constant-dimension, minimum-discrepancy decoding is equivalent to minimum-injection-distance decoding. Moreover, note that only the matrix x is needed in order to identify X. Thus, we can express the decoding problem as

$$\hat{x} = \operatorname*{argmin}_{x \in \mathcal{C}} d_{\mathrm{I}}(\Pi(x), \langle Y \rangle) \tag{4.53}$$

where $Y \in \mathbb{F}_q^{N \times (n+m)}$ is the received matrix. Without loss of generality, we can assume that $\mathsf{rank}\, Y = N$, since any linearly dependent rows of Y can be safely discarded without changing $\langle Y \rangle$. Now, define matrices $\hat{A} \in \mathbb{F}_q^{N \times n}$ and $y \in \mathbb{F}_q^{n \times m}$ such that $Y = \begin{bmatrix} \hat{A} & y \end{bmatrix}$. Following the proof of Proposition 4.23, we can write

$$\begin{aligned} d_{\mathrm{I}}(\Pi(x), \langle Y \rangle) &= \mathsf{rank} \begin{bmatrix} I & x \\ \hat{A} & y \end{bmatrix} - \min\{n, N\} \\ &= \mathsf{rank} \begin{bmatrix} I & x \\ 0 & y - \hat{A}x \end{bmatrix} - \min\{n, N\} \\ &= \mathsf{rank}\,(y - \hat{A}x) - \min\{0, N - n\}. \end{aligned} \tag{4.54}$$

Since the rightmost term in the above equation does not depend on x, we conclude that (4.53) is equivalent to

$$\hat{x} = \operatorname*{argmin}_{x \in \mathcal{C}} \mathsf{rank}\,(y - \hat{A}x). \tag{4.55}$$

Comparing (4.52) and (4.55), we can see that both problems are mathematically equivalent. In other words, we can solve the noncoherent problem with a coherent decoder by setting $(A_{\mathsf{coh}}, Y_{\mathsf{coh}}) = (\hat{A}, y)$ and $\hat{x} = \hat{X}_{\mathsf{coh}}$, and we can solve the coherent problem with a noncoherent decoder by setting $(\hat{A}, y) = (A_{\mathsf{coh}}, Y_{\mathsf{coh}})$ and $\hat{X}_{\mathsf{coh}} = \hat{x}$. This latter case can be interpreted as a noncoherent model where the error matrix Z is constrained to have all-zeros as its first n columns, so that $\hat{A} = A$ $(= A_{\mathsf{coh}})$ (see the definition of the

LNCC). With this understanding, we will from now on consider only the noncoherent decoding problem.

In the next chapter, we will look for computationally efficient approaches to solving (4.55).

Chapter 5

Generalized Decoding of Rank-Metric Codes

The objective of this chapter is to develop an approach to solving (4.55) that can exploit the structure of rank-metric codes.

It is important to note that a different algorithm for solving (4.55) has been previously proposed in [18] for the case where the rank-metric code is a Gabidulin code. This algorithm is based on Sudan-style list decoding methods and can be performed in $O(n^3m)$ operations in $\mathbb{F}_q$. On one hand, the literature on rank-metric codes suggests that faster decoding algorithms should be possible; on the other hand, it is not obvious how to reconcile the theory of rank-metric codes and our decoding problem (4.55). Thus, the goal of this chapter is to propose a rank-metric approach to solving (4.55). As we will see in Chapter 6, this approach indeed allows much faster decoding algorithms, with complexity $O(d^2nm + n^2m)$.

In Section 5.1, we propose a generalized rank-metric decoding problem that captures both (4.55) and the conventional rank-metric decoding problem (2.22). In Section 5.2 we propose a coding-theoretic perspective on this problem. We show that the generalized problem differs from the conventional problem by the fact that partial information is

available about the error word. This partial information may be in the form of *erasures* (knowledge of an error location but not its value) and *deviations* (knowledge of an error value but not its location). As we show in Section 5.3, taking erasures and deviations into account (when they occur) is strictly necessary for appropriately solving (4.55).

5.1 Problem Formulation

5.1.1 Motivation

The problem we are interested in solving, (4.55), looks quite similar to a conventional rank-metric decoder problem. Indeed, (2.22) is the special case of (4.55) where $\hat{A}$ is an identity matrix. Since both problems are similar, and efficient techniques already exist for solving (2.22), one might wonder if (4.55) can be solved by the same techniques for (2.22). The goal of this section is to reformulate (4.55) in order to make this possible.

A naive approach would be to consider the auxiliary code $\mathcal{C}_{\hat{A}} = \{\hat{A}x,\, x \in \mathcal{C}\}$. Then, (4.55) can be reexpressed as the conventional rank-metric decoding problem

$$\hat{x'} = \min_{x' \in \mathcal{C}_{\hat{A}}} \mathsf{rank}(y - x') \tag{5.1}$$

where any $\hat{x} \in \{x \mid \hat{A}x = \hat{x'}\}$ is a solution to (4.55).

Although conceptually simple, this approach can only be computationally efficient if an efficient decoder for $\mathcal{C}_{\hat{A}}$ (for every admissible $\hat{A}$) is readily available at the decoder. This requirement is likely to render the approach infeasible to implement in practice. This is especially true for noncoherent network coding, since the matrix $\hat{A}$—which could be any—is only known at the time of decoding.

For this reason, we seek to reformulate the decoding problem in such a way that the structure of the code $\mathcal{C}$ can be exploited. Roughly speaking, we should attempt to manipulate y rather than x in the expression for (4.55).

Let us first consider a special case. Suppose $N = n$ and $\hat{A}$ is invertible. Then we can

write

$$\mathsf{rank}\,(y - \hat{A}x) = \mathsf{rank}\;\hat{A}^{-1}(y - \hat{A}x) = \mathsf{rank}\,(r - x)$$

where $r = \hat{A}^{-1}y$. In this case, (4.55) becomes precisely a conventional decoding problem for the rank-metric code $\mathcal{C}$, with received word $r \in \mathbb{F}_q^{n\times m}$.

In general, however, the matrix $\hat{A}$ may not be invertible. The next subsection presents our approach to this general case.

5.1.2 The Reduction Transformation

Let $\mu = n - \mathsf{rank}\;\hat{A}$ and let $\delta = \mathsf{rank}\;Y - \mathsf{rank}\;\hat{A}$. Recall that $Y = \begin{bmatrix}\hat{A} & y\end{bmatrix}$ is assumed to be a matrix with full row rank. It follows that μ and $\delta = N - \mathsf{rank}\;\hat{A}$ give the row- and column-rank deficiencies of $\hat{A}$. The essence of our approach lies in converting Y (through row operations) into a form that resembles $X = \begin{bmatrix}I & x\end{bmatrix}$; for instance, into $\bar{Y} = [I \quad r]$ as in the special case above. Then we can hope to (partially) cancel out the left portion of Y when computing (4.54).

Before proceeding, let us define a convenient notation. Let $\mathsf{dflip}\colon \mathbb{F}_q^{n\times\mu} \to \mathbb{F}_q^{\mu\times n}$ be given by $(\mathsf{dflip}(L))_{i,j} = L_{\mu-j+1,n-i+1}$. That is, $\mathsf{dflip}(\cdot)$ flips a matrix along its main diagonal, by exchanging its bottom-right and top-left corners (analogously to the transpose).

Let us now describe the steps of our proposed transformation.

Step 1: Convert Y to reduced row echelon form

Let $\mathsf{RRE}\,(Y)$ denote the reduced row echelon form of Y. For $i = 1, \ldots, N$, let p_i be the column position of the leading entry of row i in $\mathsf{RRE}\,(Y)$. Let $\mathcal{U}^c = \{p_1, \ldots, p_{n-\mu}\}$ and $\mathcal{U} = \{1, \ldots, n\} \setminus \mathcal{U}^c$. Note that $|\mathcal{U}| = \mu$.

Step 2: Insert μ all-zero rows

Let $p_0 = 0$. Let $\bar{Y}$ be a matrix constructed by inserting all-zero rows in certain positions of $\mathsf{RRE}\,(Y)$. Namely, for $i = 1, \ldots, n - \mu$, insert $p_i - p_{i-1} - 1$ all-zero rows just before the ith row, and insert further $n - p_{n-\mu}$ all-zero rows just after the $(n - \mu)$th

row. The resulting matrix $\bar{Y}$ will have all-zero rows precisely in the positions indexed by $\mathcal{U}$, and the columns indexed by $\mathcal{U}^c$ will coincide with those of an identity matrix. More precisely, $\bar{Y}$ will be an $(n+\delta) \times (n+m)$ matrix of the form

$$\bar{Y} = \begin{bmatrix} J & r \\ 0 & \hat{V} \end{bmatrix}$$

where $J \in \mathbb{F}_q^{n \times n}$, $r \in \mathbb{F}_q^{n \times m}$ and $\hat{V} \in \mathbb{F}_q^{\delta \times m}$ satisfy

$$I_{\mathcal{U}}^T \begin{bmatrix} J & r \end{bmatrix} = 0 \tag{5.4}$$

$$J I_{\mathcal{U}^c} = I_{\mathcal{U}^c} \tag{5.5}$$

$$\mathsf{RRE}(\hat{V}) = \hat{V} \tag{5.6}$$

$$\mathsf{rank}\, \hat{V} = \delta. \tag{5.7}$$

Step 3: Extract matrices r, $\hat{L}$, $\hat{V}$

Note that $J I_{\mathcal{U}^c} = I_{\mathcal{U}^c}$ implies $(J - I) I_{\mathcal{U}^c} = 0$. This means that the columns of $J - I$ indexed by $\mathcal{U}^c$ are all-zero (or, equivalently, only the columns indexed by $\mathcal{U}$ are nonzero). Thus, we can write $J - I = \hat{L} I_{\mathcal{U}}^T$, for some $\hat{L} \in \mathbb{F}_q^{n \times \mu}$. Note that, by construction, $\hat{L}$ is such that $\mathsf{dflip}(-\hat{L})$ is in RRE form. Finally, we can express $\bar{Y}$ as

$$\bar{Y} = \begin{bmatrix} I + \hat{L} I_{\mathcal{U}}^T & r \\ 0 & \hat{V} \end{bmatrix}. \tag{5.8}$$

Note that $\bar{Y}$ has the same row space as Y.

The tuple $(r, \hat{L}, \hat{V}) \in \mathbb{F}_q^{n \times m} \times \mathbb{F}_q^{n \times \mu} \times \mathbb{F}_q^{\delta \times m}$ obtained by the above procedure is called a *reduction* of the matrix Y. Since we are really only interested in the row space of Y, we may also call $(r, \hat{L}, \hat{V})$ a reduction of the subspace $\langle Y \rangle$.

Remark: The set $\mathcal{U}$ is not included in the definition of reduction since it can be easily obtained from $\hat{L}$: each element of $\mathcal{U}$ corresponds to the row position of the last nonzero entry of a column of $\hat{L}$. (This holds because $\mathsf{dflip}(-\hat{L})$ is in RRE form.) Thus, every

subspace is mapped to a unique reduction and every reduction uniquely identifies a subspace. □

Example 5.1: Let $q = 7$, $n = m = 5$ and suppose

$$Y = \left[\begin{array}{cc} \hat{A} & y \end{array}\right] = \left[\begin{array}{ccccc|ccccc} 3 & 0 & 4 & 5 & 4 & 4 & 6 & 2 & 3 & 5 \\ 1 & 5 & 5 & 3 & 5 & 4 & 0 & 1 & 2 & 6 \\ 4 & 3 & 1 & 6 & 1 & 0 & 4 & 3 & 1 & 0 \\ 5 & 6 & 5 & 5 & 1 & 4 & 2 & 1 & 3 & 1 \\ 6 & 2 & 2 & 4 & 4 & 6 & 6 & 0 & 1 & 2 \end{array}\right]. \tag{5.9}$$

Note that $\mathsf{rank}\, Y = N = 5$, while $\mathsf{rank}\, \hat{A} = 4$, i.e., $\mu = \delta = 1$. We first convert Y to RRE form to obtain

$$\mathsf{RRE}\,(Y) = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 6 & 0 & 0 & 0 & 4 & 2 & 3 & 1 \\ 0 & 1 & 4 & 0 & 0 & 0 & 4 & 6 & 2 & 5 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 5 & 2 & 6 & 5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 6 & 4 & 4 \end{array}\right].$$

Note that $\mathcal{U} = \{3\}$. After inserting $\mu = 1$ all-zero row in the appropriate place, we obtain

$$\bar{Y} = \left[\begin{array}{ccccc|ccccc} 1 & 0 & 6 & 0 & 0 & 0 & 4 & 2 & 3 & 1 \\ 0 & 1 & 4 & 0 & 0 & 0 & 4 & 6 & 2 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 4 & 2 & 3 \\ 0 & 0 & 0 & 0 & 1 & 0 & 5 & 2 & 6 & 5 \\ \hline 0 & 0 & 0 & 0 & 0 & 1 & 1 & 6 & 4 & 4 \end{array}\right].$$

We can now extract matrices (recall that $-1 = 6$ in $\mathbb{F}_7$)

$$\hat{L} = \begin{bmatrix} 6 \\ 4 \\ 6 \\ 0 \\ 0 \end{bmatrix} \quad r = \begin{bmatrix} 0 & 4 & 2 & 3 & 1 \\ 0 & 4 & 6 & 2 & 5 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 2 & 3 \\ 0 & 5 & 2 & 6 & 5 \end{bmatrix} \quad \hat{V} = \begin{bmatrix} 1 & 1 & 6 & 4 & 4 \end{bmatrix}$$

such that (5.8) is satisfied. Thus, $(r, \hat{L}, \hat{V})$ is a reduction of Y. □

It is worth mentioning that if $\Pi(x)$ is the lifting of x, then $(x, [\,], [\,])$ is the reduction of $\Pi(x)$. That is, reduction can be interpreted as the inverse of lifting.

We can now prove the main result of this section.

Theorem 5.1: Let $(r, \hat{L}, \hat{V}) \in \mathbb{F}_q^{n\times m} \times \mathbb{F}_q^{n\times\mu} \times \mathbb{F}_q^{\delta\times m}$ be a reduction of Y. Then

$$d_{\mathrm{I}}(\Pi(x), \langle Y\rangle) = \mathsf{rank} \begin{bmatrix} r - x & \hat{L} \\ \hat{V} & 0 \end{bmatrix} - \min\{\mu, \delta\}.$$

Proof: Let $\mathcal{U}$ be a set consistent with the reduction transformation, and let $J = I + \hat{L} I_{\mathcal{U}}^T$.

We have

$$\begin{aligned}
\mathsf{rank}\begin{bmatrix} X \\ Y \end{bmatrix} &= \mathsf{rank}\begin{bmatrix} I & x \\ J & r \\ 0 & \hat{V} \end{bmatrix} = \mathsf{rank}\begin{bmatrix} J-I & r-x \\ J & r \\ 0 & \hat{V} \end{bmatrix} \\
&= \mathsf{rank}\begin{bmatrix} J-I & r-x \\ I_{\mathcal{U}^c}^T J & I_{\mathcal{U}^c}^T r \\ 0 & \hat{V} \end{bmatrix} && (5.13) \\
&= \mathsf{rank}\begin{bmatrix} \hat{L} I_{\mathcal{U}}^T & r-x \\ I_{\mathcal{U}^c}^T & I_{\mathcal{U}^c}^T x \\ 0 & \hat{V} \end{bmatrix} && (5.14) \\
&= \mathsf{rank}\begin{bmatrix} \hat{L} I_{\mathcal{U}}^T & r-x \\ 0 & \hat{V} \end{bmatrix} + \mathsf{rank}\begin{bmatrix} I_{\mathcal{U}^c}^T & I_{\mathcal{U}^c}^T x \end{bmatrix} && (5.15) \\
&= \mathsf{rank}\begin{bmatrix} \hat{L} & r-x \\ 0 & \hat{V} \end{bmatrix} + n - \mu && (5.16)
\end{aligned}$$

where (5.13) follows from (5.4), (5.14) follows by multiplying the top submatrix on the left by $I_{\mathcal{U}^c}^T$ and subtracting from the middle submatrix, (5.15) follows from the fact that $\hat{L} I_{\mathcal{U}}^T$ and $I_{\mathcal{U}^c}^T$ have row spaces that intersect trivially, and (5.16) follows by deleting the all-zero columns in $\hat{L} I_{\mathcal{U}}^T$.

Thus,

$$\begin{aligned}
d_{\mathrm{I}}(\Pi(x), \langle Y \rangle) &= \mathsf{rank}\begin{bmatrix} X \\ Y \end{bmatrix} - \min\{\mathsf{rank}\, X, \mathsf{rank}\, Y\} \\
&= \mathsf{rank}\begin{bmatrix} r-x & \hat{L} \\ \hat{V} & 0 \end{bmatrix} + n - \mu - \min\{n, n-\mu+\delta\}
\end{aligned}$$

from which the result follows. ∎

Theorem 5.1 shows that, with the help of the reduction operation, the problem (4.53) (which is equivalent to (4.55)) can be converted to a generalized decoding problem for

the rank-metric code $\mathcal{C}$. Notably, the structure of the code is preserved in this new formulation. The problem is summarized below.

Generalized Rank-Metric Decoding: *Let $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ be a rank-metric code. Given a tuple $(r, \hat{L}, \hat{V}) \in \mathbb{F}_q^{n\times m} \times \mathbb{F}_q^{n\times \mu} \times \mathbb{F}_q^{\delta\times m}$ with $\mathsf{rank}\ \hat{L} = \mu$ and $\mathsf{rank}\ \hat{V} = \delta$, compute*

$$\hat{x} = \underset{x\in\mathcal{C}}{\operatorname{argmin}}\ \mathsf{rank} \begin{bmatrix} r - x & \hat{L} \\ \hat{V} & 0 \end{bmatrix}. \tag{5.17}$$

Note that, when $\mu = \delta = 0$, we recover precisely the conventional rank-metric decoding problem (2.22).

5.2 A Coding Theory Perspective

In this section, we develop a perspective on the generalized rank-metric decoding problem that proves useful to the understanding of the correction capability of rank-metric codes, as well as to the formulation of an efficient decoding algorithm.

5.2.1 Error Locations and Error Values

Let $\mathcal{C} \in \mathbb{F}_q^{n\times m}$ be a rank-metric code. For a transmitted codeword x and a received word r, define $e \triangleq r - x$ as the error word.

Note that if an error word e has rank τ, then we can write $e = LV$ for some full-rank matrices $L \in \mathbb{F}_q^{n\times\tau}$ and $V \in \mathbb{F}_q^{\tau\times m}$, as in (2.1). Let $L_1, \ldots, L_\tau \in \mathbb{F}_q^n$ denote the columns of L and let $V_1, \ldots, V_\tau \in \mathbb{F}_q^{1\times m}$ denote the rows of V. Then we can expand e as a summation of outer products

$$e = LV = \sum_{j=1}^{\tau} L_j V_j. \tag{5.18}$$

We will now borrow some terminology from classical coding theory. Recall that an

error vector $e \in \mathbb{F}_q^n$ of Hamming weight τ can be expanded uniquely as a sum of products

$$e = \sum_{j=1}^{\tau} I_{i_j} e_j$$

where $1 \leq i_1 < \cdots < i_\tau \leq n$ and $e_1, \ldots, e_\tau \in \mathbb{F}_q$. The index i_j (or the unit vector I_{i_j}) specifies the *location* of the jth error, while e_j specifies the *value* of the jth error.

Analogously, in the sum-of-outer-products expansion (5.18) we will refer to $L_1, \ldots, L_\tau$ as the *error locations* and to $V_1, \ldots, V_\tau$ as the *error values*. The location L_j (a column vector) indicates that, for $i = 1, \ldots, n$, the jth error value V_j (a row vector) occurred in row i multiplied by the coefficient L_{ij}. Of course, $L_{ij} = 0$ means that the jth error value is not present in row i.

Note that, in contrast to the classical case, the distinction between error locations and error values in the rank metric is merely a convention. If we prefer to think of errors as occurring on columns rather than rows, then the roles of L_j and V_j would be interchanged. The same observation will also apply to any concept derived from the interpretation of these quantities as error locations and error values.

It is important to mention that, in contrast with classical coding theory, the expansion (5.18) is not unique, since

$$e = LV = LT^{-1}TV$$

for any nonsingular $T \in \mathbb{F}_q^{\tau \times \tau}$. Thus, strictly speaking, $L_1, \ldots, L_\tau$ and $V_1, \ldots, V_\tau$ are just one possible set of error locations/values describing the error word e.

5.2.2 Erasures and Deviations

We now reformulate the generalized rank-metric decoding problem in a way that facilitates its understanding and solution.

First, observe that the problem (5.17) is equivalent to the problem of finding an error

word $\hat{e}$, given by

$$\hat{e} = \underset{e \in r - \mathcal{C}}{\operatorname{argmin}} \ \mathsf{rank} \begin{bmatrix} e & \hat{L} \\ \hat{V} & 0 \end{bmatrix}, \tag{5.20}$$

from which the output of the decoder can be computed as $\hat{x} = r - \hat{e}$.

Proposition 5.2: Let $e \in \mathbb{F}_q^{n \times m}$, $\hat{L} \in \mathbb{F}_q^{n \times \mu}$ and $\hat{V} \in \mathbb{F}_q^{\delta \times n}$. The following statements are equivalent:

1) $\tau^* = \mathsf{rank} \begin{bmatrix} e & \hat{L} \\ \hat{V} & 0 \end{bmatrix}$.

2) $\tau^* - \mu - \delta$ is the minimum value of

$$\mathsf{rank}\,(e - \hat{L}V^{(1)} - L^{(2)}\hat{V})$$

for all $V^{(1)} \in \mathbb{F}_q^{\mu \times m}$ and all $L^{(2)} \in \mathbb{F}_q^{n \times \delta}$.

3) τ^* is the minimum value of τ for which there exist $L_1, \ldots, L_\tau \in \mathbb{F}_q^n$ and $V_1, \ldots, V_\tau \in \mathbb{F}_q^{1 \times m}$ satisfying:

$$e = \sum_{j=1}^{\tau} L_j V_j$$

$$L_j = \hat{L}_j, \quad j = 1, \ldots, \mu$$

$$V_{\mu+j} = \hat{V}_j, \quad j = 1, \ldots, \delta.$$

Proof: See Appendix B.2. ∎

With the help of Proposition 5.2, the influence of $\hat{L}$ and $\hat{V}$ in the decoding problem can be interpreted as follows. Suppose $e \in r - \mathcal{C}$ is the unique solution to (5.20). Then e can be expanded as $e = \sum_{j=1}^{\tau} L_j V_j$, where $L_1, \ldots, L_\mu$ and $V_{\mu+1}, \ldots, V_{\mu+\delta}$ are *known* to the decoder. In other words, the decoding problem is facilitated, since the decoder has side information about the expansion of e.

Recall the terminology of Section 5.2.1. Observe that, for $j \in \{1, \ldots, \mu\}$, the decoder knows the *location* of the jth error term but not its value, while for $j \in \{\mu+1, \ldots, \mu+\delta\}$,

the decoder knows the *value* of the jth error term but not its location. Since in classical coding theory knowledge of an error location but not its value corresponds to an erasure, we will adopt a similar terminology here. However we will need to introduce a new term to handle the case where the value of an error is known, but not its location. In the expansion (5.18) of the error word, each term $L_j V_j$ will be called

- an *erasure*, if L_j is known;
- a *deviation*, if V_j is known; and
- a *full error* (or simply an *error*), if neither L_j nor V_j are known.

Collectively, erasures, deviations and errors will be referred to as "errata." We say that an errata pattern is *correctable* when (5.17) has a unique solution equal to the original transmitted codeword.

5.2.3 Errata Correction Capability of Rank-Metric Codes

The following theorem characterizes the errata correction capability of rank-metric codes.

Theorem 5.3: A rank-metric code $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ of minimum distance d is able to correct every pattern of ϵ errors, μ erasures and δ deviations if and only if $2\epsilon + \mu + \delta \leq d - 1$.

Proof: Let $x \in \mathcal{C}$ be a transmitted codeword and let $(r, \hat{L}, \hat{V}) \in \mathbb{F}_q^{n\times m} \times \mathbb{F}_q^{n\times\mu} \times \mathbb{F}_q^{\delta\times m}$ be a received tuple such that $\mathsf{rank}\begin{bmatrix} r-x & \hat{L} \\ \hat{V} & 0 \end{bmatrix} = \mu + \delta + \epsilon$. Suppose $x' \in \mathcal{C}$ is another codeword such that $\mathsf{rank}\begin{bmatrix} r-x & \hat{L} \\ \hat{V} & 0 \end{bmatrix} = \mu + \delta + \epsilon'$, where $\epsilon' \leq \epsilon$. From Proposition 5.2, we can write

$$e = r - x = \hat{L}V^{(1)} + L^{(2)}\hat{V} + L^{(3)}V^{(3)}$$

$$e' = r - x' = \hat{L}V^{(4)} + L^{(5)}\hat{V} + L^{(6)}V^{(6)}$$

for some $V^{(1)}, L^{(2)}, \ldots, V^{(6)}$ with appropriate dimensions such that $\mathsf{rank}\, L^{(3)}V^{(3)} = \epsilon$ and $\mathsf{rank}\, L^{(6)}V^{(6)} = \epsilon'$.

Thus,

$$e - e' = \hat{L}(V^{(1)} - V^{(4)}) + (L^{(2)} - L^{(5)})\hat{V} + L^{(3)}V^{(3)} + L^{(6)}V^{(6)}$$

and

$$\mathsf{rank}\,(x' - x) = \mathsf{rank}\,(e - e') \leq \mu + \delta + \epsilon + \epsilon' \leq d - 1$$

contradicting the minimum distance of the code.

Conversely, let $x, x' \in \mathcal{C}$ be two codewords such that $\mathsf{rank}\,(x' - x) = d$. For all μ, δ and ϵ such that $\mu + \delta + 2\epsilon \geq d$, we can write

$$x' - x = L^{(1)}V^{(1)} + L^{(2)}V^{(2)} + L^{(3)}V^{(3)} + L^{(4)}V^{(4)}$$

where the four terms above have inner dimensions equal to μ, δ, ϵ and $\epsilon' = d - \mu - \delta - \epsilon$, respectively. Let

$$e = L^{(1)}V^{(1)} + L^{(2)}V^{(2)} + L^{(3)}V^{(3)}$$
$$e' = -L^{(4)}V^{(4)}$$

and observe that $x' - x = e - e'$. Let $r = x + e = x' + e'$, $\hat{L} = L^{(1)}$ and $\hat{V} = V^{(2)}$. Suppose that x is transmitted and the tuple $(r, \hat{L}, \hat{V})$ is received. Then

$$\mathsf{rank} \begin{bmatrix} r - x & \hat{L} \\ \hat{V} & 0 \end{bmatrix} = \mathsf{rank} \begin{bmatrix} e & \hat{L} \\ \hat{V} & 0 \end{bmatrix} = \mu + \delta + \epsilon$$
$$\mathsf{rank} \begin{bmatrix} r - x' & \hat{L} \\ \hat{V} & 0 \end{bmatrix} = \mathsf{rank} \begin{bmatrix} e' & \hat{L} \\ \hat{V} & 0 \end{bmatrix} = \mu + \delta + \epsilon'.$$

Since $\epsilon' = d - \mu - \delta - \epsilon \leq \epsilon$, it follows that x cannot be the unique solution to (5.17) and therefore the errata pattern cannot be corrected. ∎

Theorem 5.3 shows that, similarly to erasures in the Hamming metric, erasures and deviations cost half of an error in the rank metric.

Theorem 5.3 also shows that taking into account information about erasures and deviations (when they occur) can strictly increase the error correction capability of a rank-metric code. Indeed, suppose that an error word of rank $t = \mu + \delta + \epsilon$ is applied to a codeword, where μ, δ and ϵ are the number of erasures, deviations and full errors, respectively, in the errata pattern. It follows that a conventional rank decoder (which ignores the information about erasures and deviations) can only guarantee successful decoding if $2t \leq d-1$, where d is the minimum rank distance of the code. On the other hand, a generalized rank decoder requires only $2\epsilon + \mu + \delta \leq d-1$, or $2t \leq d-1+\mu+\delta$, in order to guarantee successful decoding. In this case, the error correction capability is increased by $(\mu+\delta)/2$ if a generalized rank decoder is used instead of a conventional one.

5.2.4 Comparison with Previous Work

We conclude this chapter by comparing our generalized rank-metric decoding problem with other decoding problems previously proposed for rank-metric codes.

There has been a significant amount of research on the problem of correcting rank errors in the presence of "row and column erasures" [54–58], where a row erasure means that all entries of that row are replaced by an erasure symbol, and similarly for a column erasure. The decoding problem in this setting is naturally defined as finding a codeword such that, when the erased entries in the received word are replaced by those of the codeword, the difference between this new matrix and the codeword has the smallest possible rank. We now show that this problem is a special case of (5.17).

First, we force the received word r to be in $\mathbb{F}_q^{n\times m}$ by replacing each erasure symbol with an arbitrary symbol in $\mathbb{F}_q$, say 0. Suppose that the rows $i_1, \ldots, i_\mu$ and the columns $k_1, \ldots, k_\delta$ have been erased. Let $\hat{L} \in \mathbb{F}_q^{n\times\mu}$ be given by $\hat{L}_{i_j,j} = 1$ and $\hat{L}_{i,j} = 0$, $\forall i \neq i_j$, for $j = 1, \ldots, \mu$ and let $\hat{V} \in \mathbb{F}_q^{\delta\times m}$ be given by $\hat{V}_{j,k_j} = 1$ and $\hat{V}_{j,k} = 0$, $\forall k \neq k_j$, for

$j = 1, \ldots, \delta$. Since

$$\begin{bmatrix} r - x & \hat{L} \\ \hat{V} & 0 \end{bmatrix} = \begin{bmatrix} r & \hat{L} \\ \hat{V} & 0 \end{bmatrix} - \begin{bmatrix} x & 0 \\ 0 & 0 \end{bmatrix} \tag{5.25}$$

it is easy to see that we can perform column operations on (5.25) to replace the erased rows of r with the same entries as x, and similarly we can perform row operations on (5.25) to replace the erased columns of r with the same entries as x. The decoding problem (5.17) is unchanged by these operations and reduces exactly to the decoding problem with "row and column erasures" described in the previous paragraph. An example is given below.

Example 5.2: Let $n = m = 3$. Suppose the third row and the second column have been erased in the received word. Then

$$r = \begin{bmatrix} r_{11} & 0 & r_{13} \\ r_{21} & 0 & r_{23} \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{L} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \hat{V} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}.$$

Since

$$\begin{bmatrix} r_{11} & 0 & r_{13} & 0 \\ r_{21} & 0 & r_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} r_{11} & x_{12} & r_{13} & 0 \\ r_{21} & x_{22} & r_{23} & 0 \\ x_{31} & x_{32} & x_{33} & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

are row equivalent, we obtain that

$$\begin{aligned}\mathsf{rank}\begin{bmatrix} r-x & \hat{L} \\ \hat{V} & 0 \end{bmatrix} &= \mathsf{rank}\begin{bmatrix} r_{11}-x_{11} & 0 & r_{13}-x_{13} & 0 \\ r_{21}-x_{21} & 0 & r_{23}-x_{23} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\ &= 2 + \mathsf{rank}\begin{bmatrix} r_{11}-x_{11} & 0 & r_{13}-x_{13} \\ r_{21}-x_{21} & 0 & r_{23}-x_{23} \\ 0 & 0 & 0 \end{bmatrix}\end{aligned}$$

which is essentially the same objective function as in the decoding problem with "row and column erasures" described above. □

While row/column erasures are a special case of erasures/deviations, it also true that the latter can always be transformed into the former. This can be accomplished by multiplying all rank-metric codewords to the left and to the right by nonsingular matrices in such a way that the corresponding $\hat{L}_j$ and $\hat{V}_j$ become unit vectors. The drawback of this approach, as pointed out in Section 5.1.1, is that the structure of the code is changed at each decoding instance, which may increase complexity and/or raise implementation issues. Thus, it is probably more advantageous in practice to fix the structure of the code and construct a decoder that can handle the generalized notions of erasures and deviations. This is the approach that is taken in Chapter 6.

5.3 Relationship with the Linear Network Coding Channel

In this section, we relate the concepts of erasures, deviations and full errors to the model of Section 4.3. More precisely, we determine all possible values for μ, δ and ϵ under that model.

Recall once again that we are assuming that packets consist of $n+m$ symbols. Let $\mathcal{H} = \{1, \ldots, n\}$ denote the column positions of the first n entries of a packet (which we interpret as the packet *header*). Let $Y_{\mathcal{H}}$ denote the sub-matrix of Y consisting of the columns indexed by $\mathcal{H}$. In addition, note that, from the definition of full errors together with (5.16), we have

$$\epsilon \triangleq \mathsf{rank} \begin{bmatrix} r - x & \hat{L} \\ \hat{V} & 0 \end{bmatrix} - \mu - \delta = \mathsf{rank} \begin{bmatrix} X \\ Y \end{bmatrix} - n - \delta.$$

Definition 5.1: Let $A \in \mathbb{F}_q^{N \times n}$ and $X \in \mathbb{F}_q^{n \times (n+m)}$ be such that $X_{\mathcal{H}} = I$. A tuple (μ, δ, ϵ) is *admissible* for (A, X, t) if there exists an output of the LNCC that induces precisely μ erasures, δ deviations and ϵ full errors; in other words, there exist matrices $D \in \mathbb{F}_q^{N \times t}$ and $Z \in \mathbb{F}_q^{t \times (n+m)}$ such that $Y = AX + DZ$ satisfies

$$\mathsf{rank}\, Y_{\mathcal{H}} = n - \mu \tag{5.29}$$

$$\mathsf{rank}\, Y = n - \mu + \delta \tag{5.30}$$

$$\mathsf{rank} \begin{bmatrix} X \\ Y \end{bmatrix} = n + \delta + \epsilon. \tag{5.31}$$

When A and X are fixed, we may simply say that (μ, δ, ϵ) is admissible for t.

Theorem 5.4: For any $A \in \mathbb{F}_q^{N \times n}$ and $X \in \mathbb{F}_q^{n \times (n+m)}$ such that $X_{\mathcal{H}} = I$, a tuple (μ, δ, ϵ) is admissible for t if and only if

$$\begin{aligned} \max\{\rho - t,\, n - N,\, 0\} \leq \mu &\leq \rho + t \\ 0 \leq \delta &\leq \min\{t,\, t - \rho + \mu,\, N - n + \mu\} \\ 0 \leq \epsilon &\leq t - \max\{\mu - \rho,\, \delta\} \end{aligned}$$

where $\rho = n - \mathsf{rank}\, A$.

Proof: See Appendix B.2. ∎

Remark: The above result is also valid for the coherent model of Section 4.2, with the only difference that $\mu = \rho$ is fixed (see Section 4.4). □

Remark: An alternative way of writing the equations in Theorem 5.4 is

$$\mu, \delta, \epsilon \geq 0$$

$$\delta + \epsilon \leq t$$

$$n - \mu + \delta \leq N$$

$$-(t - \delta) \leq \mu - \rho \leq t - \epsilon.$$

□

Theorem 5.4 determines all values of μ erasures, δ deviations and ϵ full errors that may possibly appear in a LNCC with rank deficiency ρ and t injected error packets. As one example, consider the special case $t = 0$. Then $\mu = \rho$ and $\delta = \epsilon = 0$, i.e., the channel is subject to erasures only, and the number of erasures corresponds precisely to the column-rank deficiency of A.

As another example, consider the case $\rho = 0$, $t = 1$ and N sufficiently large. According to the theorem, the admissible values of (μ, δ, ϵ) are $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, and $(1, 1, 0)$. It is instructive to consider a subspace interpretation of this fact. Roughly speaking, the first three tuples correspond to: no change in the transmitted subspace; deletion of a dimension; and insertion of a dimension. The last two tuples both correspond to a replacement of a dimension. For the fourth tuple, the new dimension contains a nonzero header, and is interpreted as a full error; for fifth tuple, however, the new dimension contains an all-zero header, and is therefore interpreted as a deviation.

Consider the lifting $\Pi(\mathcal{C})$ of a rank-metric code $\mathcal{C} \subseteq \mathbb{F}_q^{n \times m}$ with $d = d_{\mathrm{R}}(\mathcal{C}) > 2t + \rho$. A suitable decoder for $\Pi(\mathcal{C})$ must be able to correct any patterns of μ erasures, δ deviations and ϵ full errors for any admissible (μ, δ, ϵ). Since

$$\mu + \delta + 2\epsilon \leq \mu + \delta + 2t - 2\max\{\mu - \rho,\, \delta\} = 2t + \rho - |\mu - \rho - \delta| < d$$

it follows that a generalized rank-metric decoder is a suitable decoder for $\Pi(\mathcal{C})$.

We conclude this chapter by presenting an example that illustrates all the ideas discussed.

Example 5.3: Let $q = 7$, let $n = m = 5$, and let $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ be a rank-metric code with $d_{\mathrm{R}}(\mathcal{C}) = 5$. Consider an LNCC with $\rho = 0$ and $t = 2$. Since $2t + \rho = 4 < d_{\mathrm{R}}(\mathcal{C})$, we know that the lifting $\Pi(\mathcal{C})$ is able to guarantee reliable communication.

Suppose that $X = \begin{bmatrix} I & x \end{bmatrix}$ is transmitted through the LNCC, where

$$x = \begin{bmatrix} 6 & 3 & 2 & 1 & 6 \\ 2 & 5 & 0 & 4 & 6 \\ 5 & 2 & 5 & 0 & 4 \\ 3 & 5 & 2 & 5 & 0 \\ 1 & 3 & 5 & 2 & 5 \end{bmatrix}$$

is a codeword of $\mathcal{C}$. Let the channel parameters be

$$A = \begin{bmatrix} 3 & 0 & 4 & 5 & 4 \\ 1 & 5 & 0 & 3 & 5 \\ 4 & 3 & 6 & 6 & 1 \\ 5 & 6 & 6 & 5 & 1 \\ 6 & 2 & 5 & 4 & 4 \end{bmatrix} \qquad D = \begin{bmatrix} 0 & 4 \\ 2 & 5 \\ 5 & 2 \\ 1 & 0 \\ 3 & 5 \end{bmatrix}$$

and

$$Z = \left[\begin{array}{ccccc|ccccc} 0 & 0 & 6 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 & 6 & 2 & 4 & 4 & 0 \end{array}\right].$$

Then the channel output $Y = AX + DZ$ is given precisely by (5.9). The reduction $(r, \hat{L}, \hat{V})$ of this matrix was computed in Example 5.1. It follows that there are $\mu = 1$

erasures and $\delta = 1$ deviations. The number of full errors is given by

$$\epsilon = \mathsf{rank} \begin{bmatrix} r - x & \hat{L} \\ \hat{V} & 0 \end{bmatrix} - \mu - \delta = \mathsf{rank} \left[\begin{array}{ccccc|c} 1 & 1 & 0 & 2 & 2 & 6 \\ 5 & 6 & 6 & 5 & 6 & 4 \\ 2 & 5 & 2 & 0 & 3 & 6 \\ 4 & 3 & 2 & 4 & 3 & 0 \\ 6 & 2 & 4 & 4 & 0 & 0 \\ \hline 1 & 1 & 6 & 4 & 4 & 0 \end{array}\right] - 2 = 1.$$

Since $\mu + \delta + 2\epsilon = 4 < d_{\mathrm{R}}(\mathcal{C})$, we confirm that a generalized rank-metric decoding will indeed decode correctly.

Note that if erasures and deviations are ignored, then $\mathsf{rank}(r - x) = 3$, which violates the correction capability of the code under a conventional rank-metric decoder. □

Chapter 6

Efficient Encoding and Decoding of Gabidulin Codes

The goal of this chapter is to propose a computationally efficient algorithm for solving the generalized rank-metric decoding problem (with errors, erasures and deviations) up to the errata correction capability of the code. Our algorithm is applicable to Gabidulin codes.

More precisely, we show in Section 6.2 how to modify the standard decoding algorithm for Gabidulin codes (reviewed in Section 6.1) in order to incorporate erasures and deviations. No significant increase of complexity is observed with this modification. Moreover, in Section 6.3 we propose the use of optimal (or low-complexity) normal bases to significantly speed up the algorithm. In addition, in Section 6.4, we propose a novel algorithm based on transform-domain approach. Such a transform-domain approach was originally proposed by Blahut for Reed-Solomon codes [59, 60]; here, we adapt it Gabidulin codes, which requires new results and interpretations regarding linearized polynomials. Additional contributions of this chapter are two new encoding algorithms for Gabidulin codes, which are described in Section 6.5. Together, our proposed encoding and decoding algorithms appear to be the fastest know to date.

In order to deal with Gabidulin codes as defined in Section 2.3, this chapter makes use of the bijection $[\cdot]_{\mathcal{A}} \colon \mathbb{F}_q^{1\times m} \leftrightarrow \mathbb{F}_{q^m}$, where $\mathcal{A} = \{\alpha_0, \ldots, \alpha_{m-1}\}$ is a basis for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. For convenience, we will treat as elements of $\mathbb{F}_{q^m}$ any vectors in $\mathbb{F}_q^{1\times m}$ defined in previous chapters. For instance, the matrices r and $\hat{V}$ are now regarded as column vectors $r \in \mathbb{F}_{q^m}^n$ and $\hat{V} \in \mathbb{F}_{q^m}^\delta$. Note that the original matrices can be recovered simply by taking $\underline{r} = [r]_{\mathcal{A}} \in \mathbb{F}_q^{n\times m}$ and $\underline{\hat{V}} = [\hat{V}]_{\mathcal{A}} \in \mathbb{F}_q^{\delta\times m}$ (recall the notations of Sections 2.2 and 2.3).

Throughout this chapter, let $\mathcal{C} \subseteq \mathbb{F}_{q^m}^n$ be a Gabidulin code with $d_{\mathrm{R}}(\mathcal{C}) = d$ defined by the parity-check matrix (2.26). For convenience, define also the column vector $h = \begin{bmatrix} h_0 & \cdots & h_{n-1} \end{bmatrix}^T \in \mathbb{F}_{q^m}^n$ such that h^T is the first row of H.

6.1 Standard Decoding Algorithm

Let $c \in \mathcal{C}$ be a transmitted word, let $e \in \mathbb{F}_{q^m}^n$ be an error word such that

$$\tau \triangleq \mathsf{rank}\, e \leq \frac{d-1}{2} \tag{6.1}$$

and let $r = c + e$. Given the received word r, the conventional rank-metric decoding problem is to find the unique error word $e \in r - \mathcal{C}$ satisfying (6.1). Since e can be expanded as

$$e = \sum_{j=1}^{\tau} L_j V_j \tag{6.2}$$

the problem can be broken down into finding the error locations $L_1, \ldots, L_\tau \in \mathbb{F}_q^n$ and the error values $V_1, \ldots, V_\tau \in \mathbb{F}_{q^m}$.

We now describe two versions of the standard decoding algorithm.

6.1.1 ESP Version

The first step in the decoding consists of computing the *syndromes*

$$S_\ell = \sum_{i=0}^{n-1} h_i^{[\ell]} r_i, \quad \ell = 0, \ldots, d-2. \tag{6.3}$$

Since $Hr = He$, we can express the syndromes as

$$\begin{aligned} S_\ell &= \sum_{i=0}^{n-1} h_i^{[\ell]} e_i = \sum_{i=0}^{n-1} h_i^{[\ell]} \sum_{j=1}^{\tau} L_{ij} V_j \\ &= \sum_{j=1}^{\tau} X_j^{[\ell]} V_j, \quad \ell = 0, \ldots, d-2 \end{aligned} \tag{6.4}$$

where

$$X_j = \sum_{i=0}^{n-1} L_{ij} h_i = L_j^T h, \quad j = 1, \ldots, \tau \tag{6.5}$$

are called the *error locators* associated with $L_1, \ldots, L_\tau$.

The solution of the syndrome equation (6.4) can be facilitated by the use of linearized polynomials. Define the the *error span polynomial* (ESP) $\Gamma(x)$ as the minimal q-polynomial of $V_1, \ldots, V_\tau$. Then, it can be shown [37, 56] that

$$\sum_{i=0}^{\tau} \Gamma_i S_{\ell-i}^{[i]} = 0, \quad \ell = \tau, \ldots, d-2. \tag{6.6}$$

The equation above is referred to as the *key equation* for Gabidulin codes. This key equation can be efficiently solved using either the Euclidean algorithm for linearized polynomials [37] or the modified Berlekamp-Massey algorithm proposed in [56].

After the ESP is found, the error values can be obtained by computing a basis $V_1, \ldots, V_\tau$ for the root space of $\Gamma(x)$. This can be performed efficiently either by the probabilistic algorithm in [61] or by the methods in [62, Chapter 11]. In the latter approach, we first compute the matrix

$$\underline{\gamma} = [\Gamma(x)]_{\mathcal{A}}^{\mathcal{A}} \tag{6.7}$$

representing $\Gamma(x)$ as a linear map. Then, we may use Gaussian elimination to find a basis for the left null space of $\underline{\gamma}$, i.e., linearly independent $V_1, \ldots, V_\tau$ such that

$$\begin{bmatrix} V_1 \\ \vdots \\ V_\tau \end{bmatrix} \underline{\gamma} = 0. \tag{6.8}$$

To find the error locators, one can use Gabidulin's algorithm [37, pp. 9–10], which is an efficient algorithm to solve a system of the form (6.4).

After $X_1, \ldots, X_\tau$ and $V_1, \ldots, V_\tau$ are found, the error locations can be computed by inverting (6.5), i.e., by

$$L_j^T = \underline{X_j}(\underline{h})^\dagger, \tag{6.9}$$

where $(\underline{h})^\dagger$ is a right-inverse of $\underline{h}$. Note that $(\underline{h})^\dagger$ can be precomputed. Finally, the error word is computed from (6.2).

6.1.2 ELP Version

Due to the similarity between error locators and error values, an alternative approach exists where the error locators are computed before the error values.

First, compute the "reversed syndromes"

$$\tilde{S}_\ell \triangleq S_{d-2-\ell}^{[\ell-d+2]} = \sum_{j=1}^{\tau} V_j^{[\ell-d+2]} X_j, \quad \ell = 0, \ldots, d-2. \tag{6.10}$$

Note that (6.10) has the same form as (6.4), with the roles of V_j and X_j exchanged.

Define the *error locator polynomial* (ELP) $\Lambda(x)$ as the minimal q-polynomial of $X_1, \ldots, X_\tau$. Then, it can be shown [57] that

$$\sum_{i=0}^{\tau} \Lambda_i \tilde{S}_{\ell-i}^{[i]} = 0, \quad \ell = \tau, \ldots, d-2. \tag{6.11}$$

This *key equation* has the same form as (6.8) and can be solved by exactly the same methods.

After the ELP is found, we can use exactly the same procedure outlined in Section 6.1.1 to find a basis $X_1, \ldots, X_\tau$ for the root space of $\Lambda(x)$; namely, we first compute

$$\underline{\lambda} = [\Lambda(x)]_{\mathcal{A}}^{\mathcal{A}} \tag{6.12}$$

and then solve

$$\begin{bmatrix} X_1 \\ \vdots \\ X_\tau \end{bmatrix} \underline{\lambda} = 0. \tag{6.13}$$

Then, as before, Gabidulin's algorithm may be used to solve (6.10) and find the error values.

The remainder of the algorithm is identical to that of Section 6.1.1.

6.1.3 Summary and Complexity

A summary of the algorithm is given below. The algorithm consists of six steps:

1) *Find the syndromes:* Compute (6.3);
2) *Find the ESP/ELP:* Solve (6.6)/(6.11) using the Berlekamp-Massey algorithm [56];
3) *Find a basis for the root space of the ESP/ELP:*
    a) Compute (6.7)/(6.12);
    b) Solve (6.8)/(6.13) using Gaussian elimination;
4) *Find the error locators/error values:* Solve (6.4)/(6.10) using Gabidulin's algorithm [37];
5) *Find the error locations:* Compute (6.9);
6) *Find the error word:* Compute (6.2).

A breakdown of complexity is given in Table 6.1. The complexity of Step 3a corresponds to m evaluations of a linearized polynomial (with lowest-order coefficient equal to 1) of q-degree τ (see Section 2.4), while the complexity of Step 3b corresponds to converting an $m \times m$ matrix of rank $m - \tau$ to reduced column-echelon form (see Section 2.1). The complexity of the remaining steps is taken from [63]. It can be seen that the overall complexity is dominated by Steps 1 and 3a, each requiring $O(dm^3)$ operations in $\mathbb{F}_q$.

Table 6.1: Complexity of the standard decoding algorithm

Step	Operations in specified field			
	Multiplications	Additions	Inversions	Field
1	$(d-1)n$	$(d-1)(n-1)$	–	$\mathbb{F}_{q^m}$
2	$(d-1)(d-2)$	$\frac{1}{2}(d-1)(d-2)$	$\frac{1}{2}(d-1)$	$\mathbb{F}_{q^m}$
3a	τm	τm	–	$\mathbb{F}_{q^m}$
3b	$\frac{1}{2}(m-\tau)(m+\tau-1)m$	$\frac{1}{2}(m-\tau)(m+\tau-1)(m-1)$	–	$\mathbb{F}_q$
4	$\frac{3}{2}\tau^2+\frac{1}{2}\tau-1$	$\frac{3}{2}\tau(\tau-1)$	τ	$\mathbb{F}_{q^m}$
5	τnm	$\tau n(m-1)$	–	$\mathbb{F}_q$
6	τnm	$(\tau-1)nm$	–	$\mathbb{F}_q$

6.2 Incorporating Erasures and Deviations

Let $c \in \mathcal{C}$ be a transmitted word, and let $\hat{L} \in \mathbb{F}_q^{n\times\mu}$, $\hat{V} \in \mathbb{F}_{q^m}^{\delta}$ and $e \in \mathbb{F}_{q^m}^n$ be such that $\mathsf{rank}\,\hat{L} = \mu$, $\mathsf{rank}\,\hat{V} = \delta$ and

$$\tau \triangleq \mathsf{rank}\begin{bmatrix} e & \hat{L} \\ \hat{V} & 0 \end{bmatrix} \leq \frac{d-1+\mu+\delta}{2}. \tag{6.14}$$

Let $r = c + e$. Given the received tuple $(r, \hat{L}, \hat{V})$, the generalized rank-metric decoding problem is to find the unique error word $e \in r - \mathcal{C}$ satisfying (6.14). As discussed in Section 5.2, the problem can be reexpressed as finding the error locations $L_1, \ldots, L_\tau \in \mathbb{F}_q^n$ and the error values $V_1, \ldots, V_\tau \in \mathbb{F}_{q^m}$ such that

$$e = \sum_{j=1}^{\tau} L_j V_j$$

and such that $L_j = \hat{L}_j$, $j = 1, \ldots, \mu$, and $V_{\mu+j} = \hat{V}_j$, $j = 1, \ldots, \delta$.

In order to solve this problem, note that, except for Step 2, the standard decoding algorithm of Section 6.1 still applies. Namely, the syndromes $S_0, \ldots, S_{d-2}$ bear the same

relationship (6.4) to the error locators $X_1, \ldots, X_\mu$ and the error values $V_1, \ldots, V_\delta$ (or (6.10) for the reversed syndromes). Moreover, if either the ESP or the ELP is found, then the algorithm can be resumed at Step 3 and carried out unchanged. (This is because (6.14) implies $\mu + \delta \leq (d-1)/2$ and therefore $\tau \leq d-1$, which is the only assumption needed at Step 4.) Thus, only Step 2 needs to be modified in order to incorporate erasures and deviations.

In the following, we present such modification for both the ESP version and ELP version of the algorithm. In both cases, the decoder computes

$$\hat{X}_j = \hat{L}_j^T h, \quad j = 1, \ldots, \mu \tag{6.16}$$

$$\Lambda_U(x) = M_{\{\hat{X}_1, \ldots, \hat{X}_\mu\}}(x) \tag{6.17}$$

$$\Gamma_D(x) = M_{\{\hat{V}_1, \ldots, \hat{V}_\delta\}}(x). \tag{6.18}$$

Let $S(x) = \sum_{\ell=0}^{d-2} S_\ell x^{[\ell]}$ be the syndrome polynomial and let $\tilde{S}(x)$ be the partial q-reverse of $S(x)$. Also, let $\tilde{\Lambda}_U(x)$ and $\tilde{\Gamma}_D(x)$ be the partial q-reverses of $\Lambda_U(x)$ and $\Gamma_D(x)$, respectively. Let $\epsilon = \tau - \mu - \delta$.

6.2.1 ESP Version

Let $\Gamma_F(x)$ and $\Gamma_U(x)$ be the minimal linearized polynomials satisfying

$$\Gamma_F(\Gamma_D(V_j)) = 0, \quad j = \mu + \delta + 1, \ldots, \tau$$

$$\Gamma_U(\Gamma_F(\Gamma_D(V_j))) = 0, \quad j = 1, \ldots, \mu.$$

Since $V_{\mu+1}, \ldots, V_\tau$ are necessarily linearly independent, we have that the q-degrees of $\Gamma_D(x)$ and $\Gamma_F(x) \otimes \Gamma_D(x)$ are δ and $\delta + \epsilon$, respectively, and therefore the q-degree of $\Gamma_F(x)$ must be ϵ. On the other hand, all the V_j may not be linearly independent, so the q-degree of $\Gamma_U(x)$ is at most μ. It follows that the error span polynomial is given by

$$\Gamma(x) = \Gamma_U(x) \otimes \Gamma_F(x) \otimes \Gamma_D(x) \tag{6.19}$$

and it has q-degree at most τ. Since $\Gamma_D(x)$ is known to the decoder, finding $\Gamma(x)$ reduces to the determination of $\Gamma_F(x)$ and $\Gamma_U(x)$.

We define an auxiliary syndrome polynomial as

$$S_{DU}(x) = \Gamma_D(x) \otimes S(x) \otimes \tilde{\Lambda}_U(x). \tag{6.20}$$

Observe that $S_{DU}(x)$ incorporates all the information that is known at the decoder, including erasures and deviations.

Our modified key equation is given in the following theorem.

Theorem 6.1:

$$\Gamma_F(x) \otimes S_{DU}(x) \equiv \Omega(x) \mod x^{[d-1]}$$

where $\Omega(x)$ is a linearized polynomial of q-degree $\leq \tau - 1$. Equivalently, we may write

$$\sum_{i=0}^{\epsilon} \Gamma_{F,i} S^{[i]}_{DU,\mu+\delta+\ell-i} = 0, \quad \ell = \epsilon, \ldots, d-2-\mu-\delta. \tag{6.22}$$

Proof: Let $\Omega(x) = \Gamma_F(x) \otimes S_{DU}(x) \bmod x^{[d-1]}$. If $\tau \geq d-1$, we have nothing to prove, so let us assume $\tau \leq d-2$. We will show that $\Omega_\ell = 0$ for $\ell = \tau, \ldots, d-2$.

Let $\Gamma_{FD}(x) = \Gamma_F(x) \otimes \Gamma_D(x)$ and

$$S_{FD}(x) = \Gamma_F(x) \otimes \Gamma_D(x) \otimes S(x) = \Gamma_{FD}(x) \otimes S(x). \tag{6.23}$$

According to (2.30), for $\epsilon + \delta \leq \ell \leq d-2$ we have

$$\begin{aligned} S_{FD,\ell} &= \sum_{i=0}^{\epsilon+\delta} \Gamma_{FD,i} S^{[i]}_{\ell-i} = \sum_{i=0}^{\epsilon+\delta} \Gamma_{FD,i} \left(\sum_{j=1}^{\tau} X_j^{[\ell-i]} V_j \right)^{[i]} \\ &= \sum_{j=1}^{\tau} X_j^{[\ell]} \Gamma_{FD}(V_j) = \sum_{j=1}^{\mu} X_j^{[\ell]} \zeta_j, \end{aligned} \tag{6.24}$$

where

$$\zeta_j = \Gamma_{FD}(V_j), \quad j = 1, \ldots, \mu.$$

Note that $\Gamma_F(x) \otimes S_{DU}(x) = S_{FD}(x) \otimes \tilde{\Lambda}_U(x)$. Using (2.31) and (6.24), for $\mu+\epsilon+\delta \leq \ell \leq d-2$ we have

$$\begin{aligned}\Omega_\ell &= \sum_{i=0}^{\mu} \tilde{\Lambda}_{U,i}^{[\ell-i]} S_{FD,\ell-i} = \sum_{i=0}^{\mu} \Lambda_{U,\mu-i}^{[\ell-\mu]} \sum_{j=1}^{\mu} X_j^{[\ell-i]} \zeta_j \\ &= \sum_{j=1}^{\mu} \sum_{i=0}^{\mu} \Lambda_{U,i}^{[\ell-\mu]} X_j^{[\ell-\mu+i]} \zeta_j = \sum_{j=1}^{\mu} \Lambda_U(X_j)^{[\ell-\mu]} \zeta_j = 0.\end{aligned}$$

This completes the proof of the theorem. ∎

Note that this key equation reduces to the original key equation (6.6) when there are no erasures or deviations. Moreover, it can be solved by the same methods as the original key equation (6.6), e.g., using the Euclidean algorithm for linearized polynomials [37] or using the modified Berlekamp-Massey algorithm from [56], provided $2\epsilon \leq d-1-\mu-\delta$ (which is true by assumption). Note that we should take "S_ℓ" in (6.6) as $S_{DU,\ell+\mu+\delta}$ and take "d" as $d-\mu-\delta$.

After computing $\Gamma_F(x)$, we still need to determine $\Gamma_U(x)$. In the proof of Theorem 6.1, observe that (6.24) has the same form as (6.4); thus, $\zeta_1, \ldots, \zeta_\mu$ can be computed using Gabidulin's algorithm [37], since $S_{FD}(x)$ and $X_1, \ldots, X_\mu$ are known. Finally, $\Gamma_U(x)$ can be obtained as the minimal q-polynomial of $\zeta_1, \ldots, \zeta_\mu$, i.e.,

$$\Gamma_U(x) = M_{\{\zeta_1,\ldots,\zeta_\mu\}}(x). \tag{6.26}$$

6.2.2 ELP Version

Let $\Lambda_F(x)$ and $\Lambda_D(x)$ be the minimal linearized polynomials satisfying

$$\begin{aligned}\Lambda_F(\Lambda_U(X_j)) &= 0, \quad j = \mu+\delta+1, \ldots, \tau \\ \Lambda_D(\Lambda_F(\Lambda_U(X_j))) &= 0, \quad j = \mu+1, \ldots, \mu+\delta.\end{aligned}$$

As before, we have that the q-degree of $\Lambda_F(x)$ is ϵ and the q-degree of $\Lambda_D(x)$ is at most δ. The error locator polynomial is given by

$$\Lambda(x) = \Lambda_D(x) \otimes \Lambda_F(x) \otimes \Lambda_U(x) \tag{6.27}$$

and it has q-degree at most τ. Since $\Lambda_U(x)$ is known to the decoder, we only need to determine $\Lambda_F(x)$ and $\Lambda_D(x)$.

Let the auxiliary syndrome polynomial be defined as

$$S_{UD}(x) = \Lambda_U(x) \otimes \tilde{S}(x) \otimes \tilde{\Gamma}_D(x^{[d-2]})^{[-d+2]}. \tag{6.28}$$

We have the following key equation:

Theorem 6.2:

$$\Lambda_F(x) \otimes S_{UD}(x) \equiv \Psi(x) \mod x^{[d-1]}$$

where $\Psi(x)$ is a linearized polynomial of q-degree $\leq \tau - 1$. Equivalently, we may write

$$\sum_{i=0}^{\epsilon} \Lambda_{F,i} S^{[i]}_{UD,\mu+\delta+\ell-i} = 0, \quad \ell = \epsilon, \ldots, d-2-\mu-\delta. \tag{6.30}$$

Proof: Let $\Psi(x) = \Lambda_F(x) \otimes S_{UD}(x) \bmod x^{[d-1]}$. We will show that $\Psi_\ell = 0$ for $\ell = \tau, \ldots, d-2$.

Let $\Lambda_{FU}(x) = \Lambda_F(x) \otimes \Lambda_U(x)$ and

$$S_{FU}(x) = \Lambda_F(x) \otimes \Lambda_U(x) \otimes \tilde{S}(x) = \Lambda_{FU}(x) \otimes \tilde{S}(x). \tag{6.31}$$

According to (2.30), for $\epsilon + \mu \leq \ell \leq d-2$ we have

$$\begin{aligned} S_{FU,\ell} &= \sum_{i=0}^{\epsilon+\mu} \Lambda_{FU,i} \tilde{S}^{[i]}_{\ell-i} = \sum_{i=0}^{\epsilon+\mu} \Lambda_{FU,i} \left(\sum_{j=1}^{\tau} V_j^{[\ell-i-d+2]} X_j \right)^{[i]} \\ &= \sum_{j=1}^{\tau} V_j^{[\ell-d+2]} \Lambda_{FU}(X_j) = \sum_{j=1}^{\delta} V_{\mu+j}^{[\ell-d+2]} \xi_j, \end{aligned} \tag{6.32}$$

where $\xi_j = \Lambda_{FU}(X_{\mu+j})$, $j = 1, \ldots, \delta$.

Note that $\Lambda_F(x) \otimes S_{UD}(x) = S_{FU}(x) \otimes \tilde{\Gamma}_D(x^{[d-2]})^{[-d+2]}$. Using (2.31) and (6.32), for

$\delta + \epsilon + \mu \leq \ell \leq d - 2$ we have

$$\begin{aligned}\Psi_\ell &= \sum_{i=0}^{\delta} (\tilde{\Gamma}_{D,i}^{[-d+2]})^{[\ell-i]} S_{FU,\ell-i} \\ &= \sum_{i=0}^{\delta} \Gamma_{D,\delta-i}^{[\ell-\delta-d+2]} \sum_{j=1}^{\delta} V_{\mu+j}^{[\ell-i-d+2]} \xi_j \\ &= \sum_{j=1}^{\delta} \sum_{i=0}^{\delta} \Gamma_{D,i}^{[\ell-\delta-d+2]} V_{\mu+j}^{[\ell-\delta+i-d+2]} \xi_j \\ &= \sum_{j=1}^{\delta} \Gamma_D (V_{\mu+j})^{[\ell-\delta-d+2]} \xi_j = 0.\end{aligned}$$

This completes the proof of the theorem. ∎

After computing $\Lambda_F(x)$, we can determine $\Lambda_D(x)$ in the following way. First, we apply Gabidulin's algorithm on (6.32) in order to obtain $\xi_1, \ldots, \xi_\delta$. Then, we compute $\Lambda_D(x)$ as the minimal q-polynomial of $\xi_1, \ldots, \xi_\delta$, i.e.,

$$\Lambda_D(x) = M_{\{\xi_1,\ldots,\xi_\delta\}}(x). \tag{6.33}$$

6.2.3 Summary and Complexity

A summary of the modified Step 2 is given below. It consists of nine sub-steps:

2) *Find the ESP/ELP:*

   a) Compute (6.16);

   b) Compute (6.17);

   c) Compute (6.18);

   d) Compute (6.20)/(6.28);

   e) Solve (6.22)/(6.30) using the Berlekamp-Massey algorithm [56];

   f) Compute (6.23)/(6.31);

   g) Solve (6.24)/(6.32) using Gabidulin's algorithm [37];

Table 6.2: Complexity of modified Step 2 to incorporate erasures and deviations. Shorthand: $d' = d - \mu - \delta$.

Step	Operations in specified field			
	Multiplications	Additions	Inversions	Field
2a	μnm	$\mu(n-1)m$	–	$\mathbb{F}_q$
2b	μ^2	$\mu(\mu-1)$	μ	$\mathbb{F}_{q^m}$
2c	δ^2	$\delta(\delta-1)$	δ	$\mathbb{F}_{q^m}$
2d	$(d-1)(\mu+\delta)+\mu\delta$	$(d-2)(\mu+\delta)+\mu\delta$	–	$\mathbb{F}_{q^m}$
2e	$(d'-1)(d'-2)$	$\frac{1}{2}(d'-1)(d'-2)$	$\frac{1}{2}(d'-1)$	$\mathbb{F}_{q^m}$
2f (ESP)	$\epsilon(d-1+\delta)$	$\epsilon(d-2+\delta)$	–	$\mathbb{F}_{q^m}$
2g (ESP)	$\frac{3}{2}\mu^2+\frac{1}{2}\mu-1$	$\frac{3}{2}\mu(\mu-1)$	μ	$\mathbb{F}_{q^m}$
2h (ESP)	μ^2	$\mu(\mu-1)$	μ	$\mathbb{F}_{q^m}$
2f (ELP)	$\epsilon(d-1+\mu)$	$\epsilon(d-2+\mu)$	–	$\mathbb{F}_{q^m}$
2g (ELP)	$\frac{3}{2}\delta^2+\frac{1}{2}\delta-1$	$\frac{3}{2}\delta(\delta-1)$	δ	$\mathbb{F}_{q^m}$
2h (ELP)	δ^2	$\delta(\delta-1)$	δ	$\mathbb{F}_{q^m}$
2i	$\epsilon(\mu+\delta)+\mu\delta$	$\epsilon(\mu+\delta)+\mu\delta$	–	$\mathbb{F}_{q^m}$

h) Compute (6.26)/(6.33);

i) Compute (6.19)/(6.27).

The complexity of each sub-step is given in Table 6.2. A few comments are in order. The complexity of Steps 2a–2e and 2i do not vary between the ESP and the ELP versions. In Step 2f, a few operations are avoided since we can reuse the product $\Gamma_D(x) \otimes S(x)$ (or $\Lambda_U(x) \otimes \tilde{S}(x)$) computed in Step 2d. In the ESP version, Steps 2a, 2b, 2f, 2g and 2h can be skipped if there are no erasures. Similarly, in the ELP version, Steps 2c, 2f, 2g and 2h can be skipped if there are no deviations.

Assuming that the decoder can freely choose between the ESP version and the ELP version depending on the values μ and δ, the worst-case complexity is obtained in the case where $\mu = \delta = (d-1)/2$ (assuming for simplicity that d is odd). In this case, the complexity of Step 2 is given by

$$\begin{aligned} \frac{21}{8}(d-1)^2 + \frac{1}{4}(d-1) - 1 \quad & \text{multiplications in } \mathbb{F}_{q^m} \\ \frac{21}{8}(d-1)^2 - \frac{13}{4}(d-1) \quad & \text{additions in } \mathbb{F}_{q^m} \\ 2(d-1) \quad & \text{inversions in } \mathbb{F}_{q^m} \end{aligned}$$

where we have ignored the $\mathbb{F}_q$-operations in Step 2a. By comparing with Table 6.1, we can see that the complexity of Step 2 with erasures and deviations is at most 7 times the complexity of this step when only errors occur. Thus, incorporating erasures and deviations does not increase the order of complexity of the decoding algorithm.

6.3 Fast Decoding Using Low-Complexity Normal Bases

In this section, we describe how the decoding algorithm of the previous sections can be made much more efficient.

We assume that $\mathcal{A} = \{\alpha^{[i]}\}$ is a low-complexity normal basis with multiplication table T, and that $\mathbb{F}_q$ has characteristic 2. The essence of our approach lies in the following expression:

$$\underline{a\alpha^{[i]}} = \left(\underline{a}^{\leftarrow i} T\right)^{\rightarrow i}, \quad \forall i = 0, \ldots, m-1.$$

In other words, multiplying an element of $\mathbb{F}_{q^m}$ by a q-power of α costs only $C(T) - m$ *additions* in $\mathbb{F}_q$ (recall that T lies in $\mathbb{F}_2$), rather than $O(m^2)$ operations in $\mathbb{F}_q$ as in a general multiplication.

Let $H' = \left[\alpha^{[i+j]}\right]$, $0 \le i \le d-2$, $0 \le j \le n'-1$. Then $H = H'A$ for some $A \in \mathbb{F}_q^{n' \times n}$ and some $n \le n' \le m$. Thus, the map given by $c' = Ac$ is an injection from $\mathcal{C}$ to $\mathcal{C}'$, where $\mathcal{C}' \in \mathbb{F}_{q^m}^{n'}$ is the Gabidulin code defined by H'. Note that $d_{\mathrm{R}}(\mathcal{C}') = d$. Thus, a tuple $(r, \hat{L}, \hat{E})$ can be decoded by first applying a decoder for $\mathcal{C}'$ on the tuple $(Ar, A\hat{L}, \hat{E})$, yielding a codeword $c' = Ac$, and then computing $c = A^{\dagger}c'$, where $A^{\dagger}$ is a left inverse of A. The decoding complexity is equal to $n'n(2m + \mu)$ additions and multiplications in $\mathbb{F}_q$ plus the complexity of decoding $\mathcal{C}'$. Thus, we will assume in the following that $h_i = \alpha^{[i]}$, $i = 0, \ldots, n-1$. (Note that there is apparently no good reason for choosing a different H.)

Now, consider the syndrome computation in Step 1. We have

$$S_\ell = \sum_{i=0}^{n-1} r_i \alpha^{[i+\ell]}, \quad \ell = 0, \ldots, d-2. \tag{6.35}$$

It follows that the syndromes can be computed with only $(d-1)n(C(T)-m) + (d-1)(n-1)m = (d-1)(nC(T)-m)$ *additions* in $\mathbb{F}_q$ (no multiplications).

Consider the computation of

$$\gamma_j = \Gamma(\alpha^{[j]}) = \sum_{i=0}^{\tau} \Gamma_i \alpha^{[i+j]}, \quad j = 0, \ldots, m-1 \tag{6.36}$$

or

$$\lambda_j = \Lambda(\alpha^{[j]}) = \sum_{i=0}^{\tau} \Lambda_i \alpha^{[i+j]}, \quad j = 0, \ldots, m-1 \tag{6.37}$$

in Step 3a. Either computation can be done simply with $\tau m(C(T)-m) + \tau m^2 = \tau m C(T)$ additions in $\mathbb{F}_q$.

Thus, the steps that were once the most demanding ones are now among the easiest to perform.

There are some additional savings. Note that $\underline{h}$ is now an identity matrix with $m-n$ additional all-zero columns at the right; thus the cost of computing L_j from X_j in Step 5 reduces to zero. (In particular, if $n = m$, then $\underline{X_j} = L_j^T$, i.e., L_j^T is precisely the vector representation of X_j with respect to the normal basis.)

Table 6.3: Complexity of the updated decoding steps for a code matched to a normal basis of complexity $C(T)$ over a field of characteristic 2.

Step	Operations in specified field			
	Multiplications	Additions	Inversions	Field
1	–	$(d-1)(nC(T)-m)$	–	$\mathbb{F}_q$
3a	–	$\tau m C(T)$	–	$\mathbb{F}_q$
5	–	–	–	$\mathbb{F}_q$

The complexity of the updated steps is shown in Table 6.3. It follows that the decoding complexity is now dominated by Steps 2 and 4 (although the kernel computation in Step 3b may become significant if d is very small). For $n = m$ and $d = 2\tau + 1$, the overall complexity of the algorithm is approximately $\frac{11}{2}\tau^2 m^2 + \frac{1}{2}m^3$ multiplications and $\frac{11}{2}\tau^2 m C(T) + \frac{1}{2}m^3$ additions in $\mathbb{F}_q$. An example is illustrated in Fig. 6.1 for varying rates.

6.4 Transform-Domain Methods

6.4.1 Linear Maps over $\mathbb{F}_{q^m}$ and the q-Transform

In this section, unless otherwise mentioned, all polynomials are q-polynomials over $\mathbb{F}_{q^m}$ with q-degree smaller than m. If $v \in \mathbb{F}_{q^m}^n$ is a vector of length $n \leq m$ over $\mathbb{F}_{q^m}$, we will take v to have length m, i.e., $v \in \mathbb{F}_{q^m}^m$, and set $v_n = \cdots = v_{m-1} = 0$.

We adopt the following convenient notation: if $f(x) = \sum_{i=0} f_i x^{[i]}$ is q-polynomial, then $f = \begin{bmatrix} f_0 & \cdots & f_{m-1} \end{bmatrix}^T$ is a vector over $\mathbb{F}_{q^m}$, and vice-versa. Thus, $f(x)$ and f are simply equivalent representations for the sequence $f_0, \ldots, f_{m-1}$. In addition, we adopt a cyclic indexing for any such a sequence: namely, we define $f_i = f_{i \bmod m}$ for all i. With this notation, we can write the symbolic multiplication $h(x) = f(x) \otimes g(x) \bmod x^{[m]} - x$

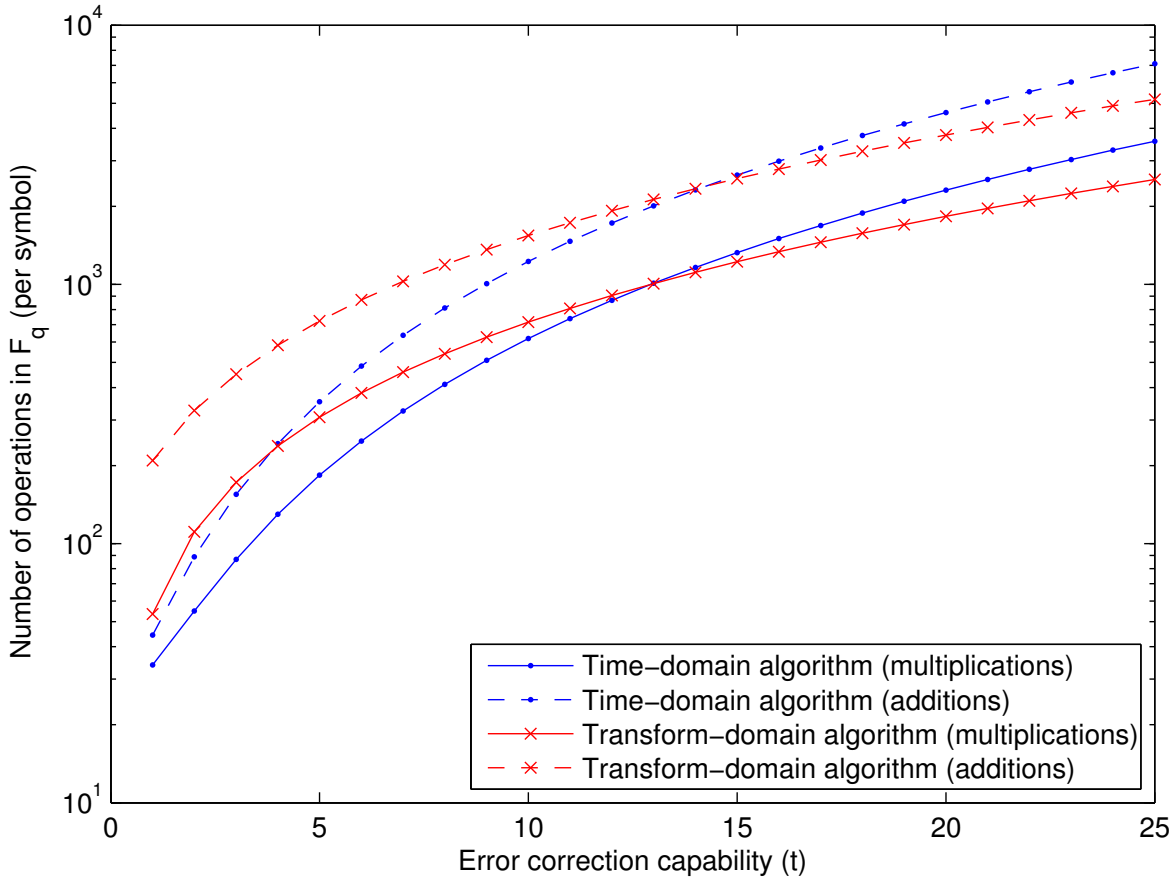

Figure 6.1: Complexity of the time-domain and transform-domain decoding algorithms, in operations per $\mathbb{F}_q$-symbol, as a function of the error correction capability t. An optimal self-dual normal basis is assumed. Parameters: $q = 256$, $n = m = 51$, and $d = 2t + 1$.

as a cyclic "q-convolution," namely,

$$h_\ell = \sum_{i=0}^{m-1} f_i g_{\ell-i}^{[i]} = \sum_{j=0}^{m-1} f_{\ell-j} g_j^{[\ell-j]}, \quad \ell = 0, \ldots, m-1. \tag{6.38}$$

Recall that the full q-reverse of a q-polynomial $f(x)$ is the q-polynomial $\bar{f}(x) = \sum_{i=0}^{m-1} \bar{f}_i x^{[i]}$, where $\bar{f}_i = f_{-i}^{[i]}$, $\forall i$.

For the remainder of this subsection, $\mathcal{A} = \{\alpha_i\}$, $\mathcal{B} = \{\beta_i\}$ and $\Theta = \{\theta_i\}$ are bases for $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$, with dual bases $\mathcal{A}' = \{\alpha_i'\}$, $\mathcal{B}' = \{\beta_i'\}$ and $\Theta' = \{\theta_i'\}$, respectively.

Lemma 6.3: $M = \left[f(x)\right]_{\mathcal{A}}^{\mathcal{B}} \iff M^T = \left[\bar{f}(x)\right]_{\mathcal{B}'}^{\mathcal{A}'}$.

Proof: By (2.16), we have

$$f(\alpha_i) = \sum_{j=0}^{m-1} M_{ij} \beta_j.$$

This implies that, for all i, j,

$$\begin{aligned}
M_{ij} &= \mathsf{Tr}\left(f(\alpha_i)\beta_j'\right) \\
&= \sum_{\ell=0}^{m-1} \mathsf{Tr}\left(f_\ell \alpha_i^{[\ell]} \beta_j'\right) \\
&= \sum_{\ell=0}^{m-1} \mathsf{Tr}\left(f_{-\ell} \alpha_i^{[-\ell]} \beta_j'\right) \\
&= \sum_{\ell=0}^{m-1} \mathsf{Tr}\left(f_{-\ell}^{[\ell]} \alpha_i \beta_j'^{[\ell]}\right) \qquad (6.40) \\
&= \mathsf{Tr}\left(\sum_{\ell=0}^{m-1} \bar{f}_\ell \beta_j'^{[\ell]} \alpha_i\right) \\
&= \mathsf{Tr}\left(\bar{f}(\beta_j')\alpha_i\right)
\end{aligned}$$

where (6.40) follows from (2.20).

Now, let $N = \left[\bar{f}(x)\right]_{\mathcal{B}'}^{\mathcal{A}'}$. From (2.16) we have

$$\bar{f}(\beta_j') = \sum_{k=0}^{m-1} N_{jk} \alpha_k'$$

which implies that, for all i, j,

$$N_{ji} = \mathsf{Tr}\left(\bar{f}(\beta'_j)\alpha_i\right) = M_{ij}.$$

Thus, $N = M^T$. The converse part follows immediately from the fact that $\bar{\bar{f}}(x) = f(x)$.■

Lemma 6.4: Suppose $\mathcal{A}$ is a normal basis. Let $F \in \mathbb{F}_{q^m}^m$ be such that $\left[F\right]_{\mathcal{B}} = \left[f(x)\right]_{\mathcal{A}}^{\mathcal{B}}$. Then $f_i = F(\alpha'_i)$, $i = 0, \ldots, m-1$. In particular, $\left[f\right]_{\Theta} = \left[F(x)\right]_{\mathcal{A}'}^{\Theta}$.

Proof: By (2.16), we have

$$F_j = f(\alpha^{[j]}), \quad j = 0, \ldots, m-1.$$

It follows that, for all i,

$$\begin{aligned} F(\alpha'^{[i]}) &= \sum_{j=0}^{m-1} F_j \alpha'^{[i+j]} \\ &= \sum_{j=0}^{m-1} f(\alpha^{[j]})\alpha'^{[i+j]} \\ &= \sum_{j=0}^{m-1}\sum_{\ell=0}^{m-1} f_\ell \alpha^{[\ell+j]}\alpha'^{[i+j]} \\ &= \sum_{\ell=0}^{m-1} f_\ell \,\mathsf{Tr}\left(\alpha^{[\ell]}\alpha'^{[i]}\right) \\ &= f_i. \end{aligned}$$

■

Definition 6.1: The q-transform of a vector $f \in \mathbb{F}_{q^m}^m$ (or a q-polynomial $f(x)$) with respect to a normal element α is the vector $F \in \mathbb{F}_{q^m}^m$ (or the q-polynomial $F(x)$) given by $F_j = f(\alpha^{[j]}) = \sum_{i=0}^{m-1} f_i \alpha^{[i+j]}$, $j = 0, \ldots, m-1$.

Theorem 6.5: The inverse q-transform of a vector $F \in \mathbb{F}_{q^m}^m$ (or a q-polynomial $F(x)$) with respect to α is given by $f_i = F(\alpha'^{[i]}) = \sum_{j=0}^{m-1} F_j \alpha^{[i+j]}$, $i = 0, \ldots, m-1$. In other words, the inverse q-transform with respect to α is equal to the forward q-transform with respect to α'.

Proof: Follows immediately from Lemma 6.4 by taking $\mathcal{B} = \Theta$ to be the basis for $\mathbb{F}_{q^m}$.■

6.4.2 Implications to the Decoding of Gabidulin Codes

Recall the notations of Section 6.1. Assume that $\mathcal{A} = \{\alpha^{[i]}\}$ is a normal basis and that the Gabidulin code has parity-check matrix $H = \left[\alpha^{[i+j]}\right]$.

As in the transform-domain decoding of Reed-Solomon codes [60], the equation $r = c+e$, or $r(x) = c(x)+e(x)$, is translated to the transform domain as $R(x) = C(x)+E(x)$, where $R(x)$, $C(x)$ and $E(x)$ are the q-transforms with respect to α of $r(x)$, $c(x)$ and $e(x)$, respectively. Now, the fact that $C_\ell = c(\alpha^{[\ell]}) = 0$, $\ell = 0, \ldots, d-2$, implies that $S_\ell = R_\ell = E_\ell$, $\ell = 0, \ldots, d-2$. Note also that $\tilde{S}_\ell = \bar{E}_{\ell-d+2}$, $\ell = 0, \ldots, d-2$.

Lemma 6.6: Let $A(x)$, $B(x)$ and $E(x)$ be linearized polynomials with q-degrees at most a, b and $m-1$, respectively. Let $S(x)$ be another linearized polynomial, and suppose that $E(x)$ and $S(x)$ agree in the coefficients $\ell = 0, \ldots, d-2$. Then $A(x) \otimes E(x) \otimes B(x) \bmod x^{[m]} - x$ and $A(x) \otimes S(x) \otimes B(x)$ agree in the coefficients $\ell = a+b, \ldots, d-2$.

Proof: Let $P(x) = A(x) \otimes E(x) \bmod x^{[m]} - x$ and let $Q(x) = A(x) \otimes S(x)$. From (6.38) and (2.30), we have that, for $\ell = a, \ldots, d-2$,

$$P_\ell = \sum_{i=0}^{a} A_i E_{\ell-i}^{[i]} = \sum_{i=0}^{a} A_i S_{\ell-i}^{[i]} = Q_\ell.$$

Now, let $\Psi(x) = P(x) \otimes B(x) \bmod x^{[m]} - x$ and let $\Omega(x) = Q(x) \otimes B(x)$. From (6.38) and (2.30), we have that, for $\ell = a+b, \ldots, d-2$,

$$\Psi_\ell = \sum_{j=0}^{b} E_{\ell-j} B_j^{[\ell-j]} = \sum_{j=0}^{b} S_{\ell-j} B_j^{[\ell-j]} = \Omega_\ell. \qquad \blacksquare$$

Theorem 6.7 (The Key Equations):

$$\Gamma(x) \otimes E(x) \equiv 0 \pmod{x^{[m]} - x} \tag{6.45}$$

$$\Lambda(x) \otimes \bar{E}(x) \equiv 0 \pmod{x^{[m]} - x}. \tag{6.46}$$

In particular, (6.6) and (6.11) hold.

Proof: For the first key equation, let $\underline{\gamma} = \Big[\Gamma(x)\Big]_{\mathcal{A}}^{\mathcal{A}}$. Note that $\Gamma(V_j) = 0$ implies $\underline{V_j}\,\underline{\gamma} = 0$, $j = 1, \ldots, \tau$. From (2.18) we have

$$\Big[\Gamma(x) \otimes E(x)\Big]_{\mathcal{A}'}^{\mathcal{A}} = \Big[E(x)\Big]_{\mathcal{A}'}^{\mathcal{A}} \Big[\Gamma(x)\Big]_{\mathcal{A}}^{\mathcal{A}} = \underline{e}\,\underline{\gamma} = \sum_{j=1}^{\tau} \underline{X_j}^T\, \underline{V_j}\,\underline{\gamma} = 0.$$

Thus, we obtain (6.45). The form (6.6) of this key equation follows immediately after applying Lemma 6.6.

For the second key equation, let $\underline{\lambda} = \Big[\Lambda(x)\Big]_{\mathcal{A}}^{\mathcal{A}}$. Note that $\Lambda(X_j) = 0$ implies $\underline{X_j}\,\underline{\lambda} = 0$, $j = 1, \ldots, \tau$. From (2.18) we have

$$\Big[\Lambda(x) \otimes \bar{E}(x)\Big]_{\mathcal{A}'}^{\mathcal{A}} = \Big[\bar{E}(x)\Big]_{\mathcal{A}'}^{\mathcal{A}} \Big[\Gamma(x)\Big]_{\mathcal{A}}^{\mathcal{A}} = \underline{e}^T\,\underline{\lambda} = \sum_{j=1}^{\tau} \underline{V_j}^T\, \underline{X_j}\,\underline{\lambda} = 0.$$

To obtain the form (6.11), we first have to multiply both sides of (6.46) on the right by $x^{[d-2]}$ (so that $\bar{E}(x) \otimes x^{[d-2]}$ agrees with $\tilde{S}(x)$), and then apply Lemma 6.6. ∎

Theorem 6.8 (The Modified Key Equations):

$$\Gamma_F(x) \otimes \Gamma_D(x) \otimes E(x) \otimes \bar{\Lambda}_U(x) \equiv 0 \pmod{x^{[m]} - x} \tag{6.49}$$

$$\Lambda_F(x) \otimes \Lambda_U(x) \otimes \bar{E}(x) \otimes \bar{\Gamma}_D(x) \equiv 0 \pmod{x^{[m]} - x}. \tag{6.50}$$

In particular, (6.22) and (6.30) hold.

Proof: For the first key equation, let $\Gamma_{FD}(x) = \Gamma_F(x) \otimes \Gamma_D(x)$ and $\underline{\gamma_{FD}} = \Big[\Gamma_{FD}(x)\Big]_{\mathcal{A}}^{\mathcal{A}}$. Note that $\Gamma_{FD}(V_j) = 0$ implies $\underline{V_j}\,\underline{\gamma_{FD}} = 0$, $j = \mu+1, \ldots, \tau$. Similarly, let $\underline{\lambda_U} = \Big[\Lambda_U(x)\Big]_{\mathcal{A}}^{\mathcal{A}}$ and note that $\underline{X_j}\,\underline{\lambda_U} = 0$, $j = 1, \ldots, \mu$. From (2.18) we have

$$\begin{aligned}\Big[\Gamma_{FD}(x) \otimes E(x) \otimes \bar{\Lambda}_U(x)\Big]_{\mathcal{A}'}^{\mathcal{A}} &= \Big[\bar{\Lambda}_U(x)\Big]_{\mathcal{A}'}^{\mathcal{A}'} \Big[E(x)\Big]_{\mathcal{A}'}^{\mathcal{A}} \Big[\Gamma_{FD}(x)\Big]_{\mathcal{A}}^{\mathcal{A}} \\ &= \underline{\lambda_U}^T\, \underline{e}\,\underline{\gamma_{FD}} \\ &= \sum_{j=1}^{\tau} \underline{\lambda_U}^T\, \underline{X_j}^T\, \underline{V_j}\,\underline{\gamma_{FD}} = 0.\end{aligned}$$

For the second key equation, let $\Lambda_{FU}(x) = \Lambda_F(x) \otimes \Lambda_U(x)$ and $\underline{\lambda_{FU}} = \left[\Lambda_{FU}(x)\right]_{\mathcal{A}}^{\mathcal{A}}$. Note that $\Lambda_{FU}(X_j) = 0$ implies $\underline{X_j}\,\underline{\lambda_{FU}} = 0$, $j = 1, \ldots, \mu, \mu+\delta+1, \ldots, \tau$. Similarly, let $\underline{\gamma_D} = \left[\Gamma_D(x)\right]_{\mathcal{A}}^{\mathcal{A}}$ and note that $\underline{V_j}\,\underline{\gamma_D} = 0$, $j = \mu+1, \ldots, \mu+\delta$. From (2.18) we have

$$\begin{aligned}\left[\Lambda_{FU}(x) \otimes \bar{E}(x) \otimes \bar{\Gamma}_D(x)\right]_{\mathcal{A}'}^{\mathcal{A}} &= \left[\bar{\Gamma}_D(x)\right]_{\mathcal{A}'}^{\mathcal{A}'} \left[\bar{E}(x)\right]_{\mathcal{A}'}^{\mathcal{A}} \left[\Lambda_{FU}(x)\right]_{\mathcal{A}}^{\mathcal{A}} \\ &= \underline{\gamma_D}^T \underline{e}^T\, \underline{\lambda_{FU}} \\ &= \sum_{j=1}^{\tau} \underline{\gamma_D}^T\, \underline{V_j}^T\, \underline{X_j}\, \underline{\lambda_{FU}} = 0.\end{aligned}$$

The alternative forms (6.22) and (6.30) follow after applying Lemma 6.6 and performing a few manipulations. In particular, (6.30) follows by applying Lemma 6.6 on $(\Lambda_F(x) \otimes \Lambda_U(x)) \otimes \left(\bar{E}(x) \otimes x^{[d-2]}\right) \otimes \left(x^{[-d+2]} \otimes \bar{\Gamma}_D(x) \otimes x^{[d-2]}\right) \bmod x^{[m]} - x$. ∎

Besides allowing us to give conceptually simpler proofs of the key equations, the transform approach also provides us with the theoretical ground for proposing a new decoding algorithm for Gabidulin codes. The main idea is that, after the ESP or the ELP is found, the remaining coefficients of $E(x)$ can be computed using the recursion

$$E_\ell = -\sum_{i=1}^{\tau} \Gamma_i E_{\ell-i}^{[i]} = 0, \quad \ell = d-1, \ldots, m-1 \tag{6.51}$$

$$\bar{E}_\ell = -\sum_{i=0}^{\tau} \Lambda_i \bar{E}_{\ell-i}^{[i]} = 0, \quad \ell = 1, \ldots, m-d+1. \tag{6.52}$$

Then, the error polynomial $e(x)$ can be obtained through an inverse q-transform, i.e.,

$$e_i = E(\alpha'^{[i]}) = \sum_{j=0}^{m-1} e_j \alpha'^{[i+j]}, \quad i = 0, \ldots, n-1 \tag{6.53}$$

where $\mathcal{A}' = \{\alpha'^{[i]}\}$ is the dual basis of $\mathcal{A}$.

Computing this inverse transform takes, in general, nm multiplications and additions in $\mathbb{F}_{q^m}$ (or km if the code is systematic and the parity portion is ignored). However, if $\mathcal{A}$ is a self-dual normal basis, then an inverse transform becomes a forward transform, and the same computational savings described in Section 6.3 can be obtained here. Note

Table 6.4: Complexity of the specific steps for transform-domain decoding of a code matched to a self-dual normal basis of complexity $C(T)$ over a field of characteristic 2.

Step	Operations in specified field			
	Multiplications	Additions	Inversions	Field
3'	$(m-d+1)\tau$	$(m-d+1)(\tau-1)$	–	$\mathbb{F}_q$
4' (nonsystematic)	–	$nm(C(T)-1)$	–	$\mathbb{F}_q$
4' (systematic)	–	$km(C(T)-1)$	–	$\mathbb{F}_q$

that most normal bases constructed via Gauss periods over fields of characteristic 2 are indeed self-dual (see Section 2.5 and, e.g., [45]).

The complete algorithm consists of four steps:

1) *Find the syndromes:* See Section 6.3;

2) *Find the ESP/ELP:* See Section 5.2.2;

3') *Find the error transform:* Compute (6.51)/(6.52) recursively;

4') *Find the error word:* Compute (6.53).

The complexity of the new Steps 3' and 4' is displayed in Table 6.4. As it can be seen from Step 3', the new algorithm essentially replaces the $O(d^2)$ operations of Gabidulin's algorithm with the $O(d(m-d))$ operations required for recursively computing $E(x)$. Thus, the algorithm is most beneficial for low-rate codes. For $n = m$ and $d = 2\tau + 1$, the overall complexity of the algorithm is approximately $(m+2t)tm^2$ multiplications and $(m+2t)tmC(T)$ additions in $\mathbb{F}_q$. An example is illustrated in Fig. 6.1 for varying rates.

6.5 Fast Encoding

As for any linear block code, encoding of Gabidulin codes requires, in general, $O(kn)$ operations in $\mathbb{F}_{q^m}$, or $O(k(n-k))$ operations in $\mathbb{F}_{q^m}$ if systematic encoding is used.

We show below that, if the code has a high rate and $\mathbb{F}_{q^m}$ admits a low-complexity normal basis, then the encoding complexity can be significantly reduced. Alternatively, if nonsystematic encoding is allowed and $\mathbb{F}_{q^m}$ admits a self-dual low-complexity normal basis, then very fast encoding is possible.

6.5.1 Systematic Encoding of High-Rate Codes

Let $c_{n-k}, \ldots, c_{n-1} \in \mathbb{F}_{q^m}$ denote the message coefficients. We set $r_i = 0$, $i = 0, \ldots, n-k-1$ and $r_i = c_i$, $i = n-k, \ldots, n-1$, and perform *erasure decoding* on $r = \begin{bmatrix} r_0 & \cdots & r_{n-1} \end{bmatrix}^T$ to obtain $c_0, \ldots, c_{n-k-1}$.

We use the algorithm of Section 6.1, with the computational savings of Section 6.3. Note that only steps 1, 4 and 5 need to be performed, since the error locations (and thus also the error locators) are known: for $j = 1, \ldots, d-1$, L_j is a column vector with a 1 in the jth position and zero in all others. Thus, the complexity is dominated by Gabidulin's algorithm, requiring $O(d^2)$ operations in $\mathbb{F}_{q^m}$ (see Section 6.1). For high-rate codes, this improves on the previous value of $O(dk)$ mentioned above. Note that, without the approach in Section 6.3, encoding by erasure decoding would cost $O(dn)$ operations in $\mathbb{F}_{q^m}$.

6.5.2 Nonsystematic Encoding

Here we assume that $n = m$. Let $F_{m-k}, \ldots, F_{m-1}$ denote the message coefficients, and let $F(x) = \sum_{j=m-k}^{m-1} F_j x^{[j]}$. We encode by taking the (inverse) q-transform with respect to α, where $\mathcal{A} = \{\alpha^{[i]}\}$ is a self-dual normal basis. Then $c_i = F(\alpha^{[i]})$, $i = 0, \ldots, m-1$. It is clear that this task takes only $mkC(T)$ additions in $\mathbb{F}_q$, and is therefore extremely fast. The decoding task, however, has to be slightly updated.

Since, by construction, every codeword satisfies $c(\alpha^{[i]}) = 0$ for $i = 0, \ldots, d-2$, most part of the decoding can remain the same. If decoding is performed in the time domain, then one additional step is needed to obtain the message: namely, computing the forward

q-transform $F_j = c(\alpha^{[j]})$, for $j = m-k, \ldots, m-1$. These extra $mkC(T)$ additions in $\mathbb{F}_q$ barely affect the decoding complexity. On the other hand, if decoding is performed in the transform domain, than the last step (obtaining $e(x)$ from $E(x)$) can be simply skipped, as $F(x) = R(x) - E(x)$. This further saves at least $mkC(T)$ additions in $\mathbb{F}_q$.

6.6 Practical Considerations

We have seen that the complexity of decoding a Gabidulin code $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ with $d_{\mathrm{R}}(\mathcal{C}) = d$ is given by $O(dm)$ operations in $\mathbb{F}_{q^m}$. In many applications, in particular for network coding, we have $m \gg n$. In such cases, the decoding complexity can be significantly reduced by using, rather than a Gabidulin code, an MRD code formed by the Cartesian product of many shorter Gabidulin codes with the same distance. More precisely, let $\ell = \lfloor \frac{m}{n} \rfloor$ and $n' = m - n(\ell - 1)$. Take $\mathcal{C} = C_1 \times C_2 \times \cdots \times C_\ell$, where $C_i \subseteq \mathbb{F}_q^{n\times n}$, $i = 1, \ldots, \ell - 1$, and $C_\ell \subseteq \mathbb{F}_q^{n\times n'}$ are Gabidulin codes with minimum rank distance d. Then, according to Theorem 2.2, $\mathcal{C}$ is an MRD code with $d_{\mathrm{R}}(\mathcal{C}) = d$.

Now, decoding of $\mathcal{C}$ can be performed by decoding each $\mathcal{C}_i$ individually. Thus, assuming for simplicity that $m = n\ell$, the overall decoding complexity is given by $\ell O(dn) = O(dm)$ operations in $\mathbb{F}_{q^n}$. In other words, operations in a potentially large field $\mathbb{F}_{q^m}$ can be replaced by operations in a much smaller field $\mathbb{F}_{q^n}$.

Note that, in this case, additional computational savings may be obtained, since all received words will share the same set of error locations. For instance, if all error locations are known and the decoding algorithm of Section 6.1 is used, then only steps 1 and 4–6 need to be performed.

Chapter 7

Error Control under a Probabilistic Error Model

In this chapter, we turn our attention to the LNCC under a probabilistic error model. In particular, we investigate the noncoherent channel described by the channel law

$$Y = AX + DZ \tag{7.1}$$

where $A \in \mathcal{T}_{n\times n}$, $D \in \mathcal{T}_{n,t}$ and $Z \in \mathcal{T}_{t\times m}$ are chosen uniformly at random and independently from other variables. Our main objective is to derive approximate expressions for the capacity of this channel and to devise efficient capacity-achieving schemes.

We start by highlighting the main differences between our approaches for adversarial and probabilistic channels. Then, we proceed by analyzing two simple channels: a multiplicative matrix channel, in Section 7.2, and an additive matrix channel, in Section 7.3. The multiplicative channel can be seen as a special case of (7.1) when no errors occur. The additive channel, on the other hand, can be seen as a coherent LNCC such that the transfer matrix is invertible. For both channels, we compute the exact capacity and propose simple capacity-achieving schemes. In Section 7.4, we combine both approaches to produce an additive-multiplicative matrix channel that is equivalent to (7.1). For this channel, we can only provide upper and lower bounds on the capacity, but we show that

these bounds match when either the field size or the packet length grows to infinity. We also propose a simple capacity-achieving scheme based on the ideas of Sections 7.2 and 7.3. Finally, in Section 7.5, we discuss extensions to the results of this chapter, such as the case where the error matrix may have variable rank.

The problem of error control in network coding under a probabilistic error model was initially suggested in [23], which provided the motivation for this chapter. The authors in [23] proposed a coding scheme based on random sparse graphs and iterative decoding. However, besides not being capacity-achieving, the scheme of [23] has significantly high decoding complexity ($O(n^3m)$) and a relatively slow decay in the probability of error. The scheme we propose not only is capacity-achieving but also has a lower decoding complexity ($O(n^2m)$) and a better decay in the probability of failure—besides being conceptually much simpler.

7.1 Matrix Channels

For clarity and consistency of notation, we recall a few definitions from information theory [64].

A discrete channel $(\mathcal{X}, \mathcal{Y}, p_{Y|X})$ consists of an input alphabet $\mathcal{X}$, an output alphabet $\mathcal{Y}$, and a conditional probability distribution $p_{Y|X}$ relating the channel input $X \in \mathcal{X}$ and the channel output $Y \in \mathcal{Y}$. An (M, ℓ) code for a channel $(\mathcal{X}, \mathcal{Y}, p_{Y|X})$ consists of an encoding function $\{1, \ldots, M\} \to \mathcal{X}^\ell$ and a decoding function $\mathcal{Y}^\ell \to \{1, \ldots, M, f\}$, where f denotes a decoding failure. It is understood that an (M, ℓ) code is applied to the ℓth extension of the discrete memoryless channel $(\mathcal{X}, \mathcal{Y}, p_{Y|X})$. A rate R (in bits) is said to be achievable if there exists a sequence of $(\lceil 2^{\ell R} \rceil, \ell)$ codes such that decoding is unsuccessful (either an error or a failure occurs) with probability arbitrarily small as $\ell \to \infty$. The capacity of the channel is the supremum of all achievable rates. It is well-known that the

capacity is given by

$$C = \max_{p_X} I(X;Y)$$

where p_X denotes the input distribution.

Here, we are interested in matrix channels, i.e., channels for which both the input and output variables are matrices. In particular, we are interested in a family of additive matrix channels given by the channel law

$$Y = AX + DZ \tag{7.3}$$

where $X, Y \in \mathbb{F}_q^{n\times m}$, $A \in \mathbb{F}_q^{n\times n}$, $D \in \mathbb{F}_q^{n\times t}$, $Z \in \mathbb{F}_q^{t\times m}$, and X, (A, D) and Z are statistically independent. Since the capacity of a matrix channel naturally scales with nm, we also define a *normalized capacity*

$$\overline{C} = \frac{1}{nm} C.$$

In the following, we assume that statistics of A, D and Z are given for all q, n, m, t. In this case, we may denote a matrix channel simply by the tuple (q, n, m, t), and we may also indicate this dependency in both C and $\overline{C}$. We now define two limiting forms of a matrix channel (strictly speaking, of a sequence of matrix channels). The first form, which we call the *infinite-field-size channel*, is obtained by taking $q \to \infty$. The capacity of this channel is given by

$$\lim_{q\to\infty} \frac{1}{\log_2 q} C(q, n, m, t)$$

represented in q-ary units per channel use. The second form, which we call the *infinite-rank channel*, is obtained by setting $t = \tau n$ and $n = \lambda m$, and taking $m \to \infty$. The normalized capacity of this channel is given by

$$\lim_{m\to\infty} \frac{1}{\log_2 q} \overline{C}(q, \lambda m, m, \tau\lambda m)$$

represented in q-ary units per transmitted q-ary symbol. We will hereafter assume that logarithms are taken to the base q and omit the factor $\frac{1}{\log_2 q}$ from the above expressions.

Note that, to achieve the capacity of an infinite-field-size channel (similarly for an infinite-rank channel), one should find a two-dimensional family of codes: namely, a sequence of codes with increasing block length ℓ for each q, as $q \to \infty$ (or for each m, as $m \to \infty$).

We will simplify our task here by considering only codes with block length $\ell = 1$, which we call *one-shot codes*. We will show, however, that these codes can achieve the capacity of both the infinite-field-size and the infinite-rank channels, at least for the classes of channels considered here. In other words, one-shot codes are asymptotically optimal as either $q \to \infty$ or $m \to \infty$.

For completeness, we define also two more versions of the channel: the *infinite-packet-length channel*, obtained by fixing q, t and n, and letting $m \to \infty$, and the *infinite-batch-size channel*, obtained by fixing q, t and m, and letting $n \to \infty$. These channels will only be discussed in Section 7.5.5.

It is important to note that a $(q, n, \ell m, t)$ channel is not the same as the ℓ-extension of a (q, n, m, t) channel. For instance, the 2-extension of a (q, n, m, t) channel has channel law

$$(Y_1,\, Y_2) = (A_1 X_1 + D_1 Z_1,\; A_2 X_2 + D_2 Z_2)$$

where $(X_1, X_2) \in \left(\mathbb{F}_q^{n \times m}\right)^2$, and (A_1, D_1, Z_1) and (A_2, D_2, Z_2) correspond to independent realizations of a (q, n, m, t) channel. This is not the same as the channel law for a $(q, n, 2m, t)$ channel,

$$\begin{bmatrix} Y_1 & Y_2 \end{bmatrix} = A_1 \begin{bmatrix} X_1 & X_2 \end{bmatrix} + D_1 \begin{bmatrix} Z_1 & Z_2 \end{bmatrix}$$

since (A_2, D_2) may not be equal to (A_1, D_1). This should be contrasted to the models used in [16] and [23]. Although both models are referred to simply as "random linear network

coding," the model implied by the results in [23] is in fact an infinite-rank channel, while the model implied by the results in [16] is an infinite-packet-length-infinite-field-size channel.

We now proceed to investigating special cases of (7.3), by considering specific statistics for A, D and Z.

7.2 The Multiplicative Matrix Channel

We define the *multiplicative matrix channel* (MMC) by the channel law

$$Y = AX$$

where $A \in \mathcal{T}_{n\times n}$ is chosen uniformly at random among all $n \times n$ nonsingular matrices, and independently from X. Note that the MMC is a $(q, n, m, 0)$ channel.

7.2.1 Capacity and Capacity-Achieving Codes

In order to find the capacity of this channel, we will first solve a more general problem.

Proposition 7.1: Let $\mathcal{G}$ be a finite group that acts on a finite set $\mathcal{S}$. Consider a channel with input variable $X \in \mathcal{S}$ and output variable $Y \in \mathcal{S}$ given by $Y = AX$, where $A \in \mathcal{G}$ is drawn uniformly at random and independently from X. The capacity of this channel, in bits per channel use, is given by

$$C = \log_2 |\mathcal{S}/\mathcal{G}|$$

where $|\mathcal{S}/\mathcal{G}|$ is the number of equivalence classes of $\mathcal{S}$ under the action of $\mathcal{G}$. Any complete set of representatives of the equivalence classes is a capacity-achieving code.

Proof: For each $x \in \mathcal{S}$, let $\mathcal{G}(x) = \{gx \mid g \in \mathcal{G}\}$ denote the orbit of x under the action of $\mathcal{G}$. Recall that $\mathcal{G}(y) = \mathcal{G}(x)$ for all $y \in \mathcal{G}(x)$ and all $x \in \mathcal{S}$, that is, the orbits form equivalence classes.

For $y \in \mathcal{G}(x)$, let $\mathcal{G}_{x,y} = \{g \in \mathcal{G} \mid gx = y\}$. By a few manipulations, it is easy to show that $|\mathcal{G}_{x,y}| = |\mathcal{G}_{x,y'}|$ for all $y, y' \in \mathcal{G}(x)$. Since A has a uniform distribution, it follows that $P[Y = y \mid X = x] = 1/|\mathcal{G}(x)|$, for all $y \in \mathcal{G}(x)$.

For any $x \in \mathcal{S}$, consider the same channel but with the input alphabet restricted to $\mathcal{G}(x)$. Note that the output alphabet will also be restricted to $\mathcal{G}(x)$. This is a $|\mathcal{G}(x)|$-ary channel with uniform transition probabilities; thus, the capacity of this channel is 0. Now, the overall channel can be considered as a sum (union of alphabets) of all the restricted channels. The capacity of a sum of M channels with capacities C_i, $i = 1, \ldots, M$, is known to be $\log_2 \sum_{i=1}^{M} 2^{C_i}$ bits. Thus, the capacity of the overall channel is $\log_2 M$ bits, where $M = |\mathcal{S}/\mathcal{G}|$ is the number of orbits. A capacity-achieving code (with block length 1) may be obtained by simply selecting one representative from each equivalence class. ■

Proposition 7.1 shows that in a channel induced by a group action, where the group elements are selected uniformly at random, the receiver cannot distinguish between transmitted elements that belong to the same equivalence class. Thus, the transmitter can only communicate the choice of a particular equivalence class.

Returning to our original problem, we have $\mathcal{S} = \mathbb{F}_q^{n \times m}$ and $\mathcal{G} = \mathcal{T}_{n \times n}$ (the general linear group $GL_n(\mathbb{F}_q)$). The equivalence classes of $\mathcal{S}$ under the action of $\mathcal{G}$ are the sets of matrices that share the same row space. Thus, we can identify each equivalence class with a subspace of $\mathbb{F}_q^m$ of dimension at most n. Recall that the Gaussian coefficient

$$\begin{bmatrix} m \\ k \end{bmatrix}_q = \prod_{i=0}^{k-1} (q^m - q^i)/(q^k - q^i)$$

denotes the number of k-dimensional subspaces of $\mathbb{F}_q^m$. We have the following corollary of Proposition 7.1.

Corollary 7.2: The capacity of the MMC, in q-ary units per channel use, is given by

$$C_{\text{MMC}} = \log_q \sum_{k=0}^{n} \begin{bmatrix} m \\ k \end{bmatrix}_q .$$

A capacity-achieving code $\mathcal{C} \subseteq \mathbb{F}_q^{n\times m}$ can be obtained by ensuring that each k-dimensional subspace of $\mathbb{F}_q^m$, $k \leq n$, is the row space of some unique $X \in \mathcal{C}$.

Note that Corollary 7.2 reinforces the idea introduced in [18] that, in order to communicate under random network coding, the transmitter should encode information in the choice of a subspace (see Section 4.3).

We now compute the capacity for the two limiting forms of the channel, as discussed in Section 7.1. We have the following result.

Proposition 7.3: Let $\lambda = n/m$ and assume $0 < \lambda \leq 1/2$. Then

$$\lim_{q\to\infty} C_{\text{MMC}} = (m-n)n \tag{7.13}$$

$$\lim_{\substack{m\to\infty \\ n=\lambda m}} \overline{C_{\text{MMC}}} = 1-\lambda. \tag{7.14}$$

Proof: First, observe that

$$\begin{bmatrix} m \\ n^* \end{bmatrix}_q < \sum_{k=0}^{n} \begin{bmatrix} m \\ k \end{bmatrix}_q < (n+1) \begin{bmatrix} m \\ n^* \end{bmatrix}_q \tag{7.15}$$

where $n^* = \min\{n, \lfloor m/2 \rfloor\}$. Using (2.12), it follows that

$$(m-n^*)n^* < C_{\text{MMC}} < (m-n^*)n^* + \log_q 4(n+1). \tag{7.16}$$

The last term on the right vanishes on both limiting cases. ∎

The case $\lambda \geq 1/2$ can also be readily obtained but is less interesting since, in practice, the packet length m will be much larger than the number of packets n.

Note that an expression similar to (7.16) has been found in [65] under a different assumption on the transfer matrix (namely, that A is uniform on $\mathbb{F}_q^{n\times n}$). It is interesting

to note that, also in that case, the same conclusion can be reached about the sufficiency of transmitting subspaces [65].

An intuitive way to interpret (7.13) is the following: out of the nm symbols obtained by the receiver, n^2 of these symbols are used to describe A, while the remaining ones are used to communicate X.

Note that both limiting capacity expressions (7.13) and (7.14) can be achieved using a simple coding scheme where an $n \times (m-n)$ data matrix U is concatenated on the left with an $n \times n$ identity matrix I, yielding a transmitted matrix $X = \begin{bmatrix} I & U \end{bmatrix}$. The first n symbols of each transmitted packet may be interpreted as pilot symbols used to perform "channel sounding". Observe that this is simply the standard way of using random network coding (see Section 3.1; compare also with the *lifting* approach of Section 4.3.3).

7.3 The Additive Matrix Channel

We define the *additive matrix channel* (AMC) according to

$$Y = X + W$$

where $W \in \mathcal{T}_{n\times m,t}$ is chosen uniformly at random among all $n \times m$ matrices of rank t, independently from X. Note that the AMC is a (q, n, m, t) channel with $D \in \mathcal{T}_{n\times t}$ and $Z \in \mathcal{T}_{t\times m}$ uniformly distributed, and $A = I$.

7.3.1 Capacity

The capacity of the AMC is computed in the next proposition.

Proposition 7.4: The capacity of the AMC is given by

$$C_{\text{AMC}} = nm - \log_q |\mathcal{T}_{n\times m,t}|.$$

For $\lambda = n/m$ and $\tau = t/n$, we have the limiting expressions

$$\lim_{q\to\infty} C_{\text{AMC}} = (m-t)(n-t) \tag{7.19}$$

$$\lim_{\substack{m\to\infty \\ n=\lambda m \\ t=\tau n}} \overline{C_{\text{AMC}}} = (1-\lambda\tau)(1-\tau). \tag{7.20}$$

Proof: To compute the capacity, we expand the mutual information

$$I(X;Y) = H(Y) - H(Y|X) = H(Y) - H(W)$$

where the last equality holds because X and W are independent. Note that $H(Y) \leq nm$, and the maximum is achieved when Y is uniform. Since $H(W)$ does not depend on the input distribution, we can maximize $H(Y)$ by choosing, e.g., a uniform p_X.

The entropy of W is given by $H(W) = \log_q |\mathcal{T}_{n\times m,t}|$. The number of $n \times m$ matrices of rank t is given by (2.15)

$$|\mathcal{T}_{n\times m,t}| = q^{(n+m-t)t} \prod_{i=0}^{t-1} \frac{(1-q^{i-n})(1-q^{i-m})}{(1-q^{i-t})}.$$

Thus,

$$\begin{aligned} C_{\text{AMC}} &= nm - \log_q |\mathcal{T}_{n\times m,t}| \\ &= (m-t)(n-t) + \log_q \prod_{i=0}^{t-1} \frac{(1-q^{i-t})}{(1-q^{i-n})(1-q^{i-m})} \end{aligned}$$

The limiting expressions (7.19) and (7.20) follow immediately from the equation above. ■

As can be seen from (2.15), an $n \times m$ matrix of rank t can be specified with approximately $(n+m-t)t$ symbols. Thus, the capacity (7.19) can be interpreted as the number of symbols conveyed by Y minus the number of symbols needed to describe W.

7.3.2 A Coding Scheme

We now present an efficient coding scheme that achieves (7.19) and (7.20). The scheme is based on an "error trapping" strategy.

Let $U \in \mathbb{F}_q^{(n-v)\times(m-v)}$ be a data matrix, where $v \geq t$. A codeword X is formed by adding all-zero rows and columns to U so that

$$X = \begin{bmatrix} 0_{v\times v} & 0_{v\times(m-v)} \\ 0_{(n-v)\times v} & U \end{bmatrix}.$$

These all-zero rows and columns may be interpreted as the "error traps." Clearly, the rate of this scheme is $R = (n-v)(m-v)$.

Since the noise matrix W has rank t, we can write it as

$$W = BZ = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{bmatrix} Z_1 & Z_2 \end{bmatrix}$$

where $B_1 \in \mathbb{F}_q^{v\times t}$, $B_2 \in \mathbb{F}_q^{(n-v)\times t}$, $Z_1 \in \mathbb{F}_q^{t\times v}$ and $Z_2 \in \mathbb{F}_q^{t\times(m-v)}$. The received matrix Y is then given by

$$Y = X + W = \begin{bmatrix} B_1Z_1 & B_1Z_2 \\ B_2Z_1 & U + B_2Z_2 \end{bmatrix}.$$

We define an error trapping failure to be the event that $\mathsf{rank}\, B_1Z_1 < t$. Intuitively, this corresponds to the situation where either the row space or the column space of the error matrix has not been "trapped".

For now, assume that the error trapping is successful, i.e., $\mathsf{rank}\, B_1 = \mathsf{rank}\, Z_1 = t$. Consider the submatrix corresponding to the first v columns of Y. Since $\mathsf{rank}\, B_1Z_1 = t$, the rows of B_2Z_1 are completely spanned by the rows of B_1Z_1. Thus, there exists some matrix $\bar{T}$ such that $B_2Z_1 = \bar{T}B_1Z_1$. But $(B_2 - \bar{T}B_1)Z_1 = 0$ implies that $B_2 - \bar{T}B_1 = 0$, since Z_1 has full row rank. 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,

$$TY = TX + TW = \begin{bmatrix} B_1 Z_1 & B_1 Z_2 \\ 0 & U \end{bmatrix}$$

from which the data matrix U can be readily obtained.

The complexity of the scheme is computed as follows. In order to obtain $\bar{T}$, it suffices to perform Gaussian elimination on the left $n \times v$ submatrix of Y, for a cost of $O(nv^2)$ operations. The data matrix can be extracted by multiplying $\bar{T}$ with the top right $v \times (n-v)$ submatrix of Y, which can be accomplished in $O((n-v)v(m-v))$ operations. Thus, the overall complexity of the scheme is $O(nmv)$ operations in $\mathbb{F}_q$.

Note that $B_1 Z_1$ is available at the receiver as the top-left submatrix of Y. Moreover, the rank of $B_1 Z_1$ is already computed during the Gaussian elimination step of the decoding. Thus, the event that the error trapping fails can be readily detected at the receiver, which can then declare a decoding failure. It follows that the error probability of the scheme is zero.

Let us now compute the probability of decoding failure. Consider, for instance, $P_1 = P[\mathsf{rank}\ Z_1 = t]$, where $Z = \begin{bmatrix} Z_1 & Z_2 \end{bmatrix}$ is a full-rank matrix chosen uniformly at random. An equivalent way of generating Z is to first generate the entries of a matrix $M \in \mathbb{F}_q^{t \times m}$ uniformly at random, and then discard M if it is not full-rank. Thus, we want to compute $P_1 = P[\mathsf{rank}\ M_1 = t \mid \mathsf{rank}\ M = t]$, where M_1 corresponds to the first v columns of M. This probability is

$$\begin{aligned} P_1 &= \frac{P[\mathsf{rank}\ M_1 = t]}{P[\mathsf{rank}\ M = t]} = \frac{q^{mt} \prod_{i=0}^{t-1} (q^v - q^i)}{q^{vt} \prod_{i=0}^{t-1} (q^m - q^i)} \\ &> \prod_{i=0}^{t-1} (1 - q^{i-v}) \geq (1 - q^{t-1-v})^t \geq 1 - \frac{t}{q^{1+v-t}}. \end{aligned}$$

The same analysis holds for $P_2 = P[\mathsf{rank}\ B_1 = t]$. By the union bound, it follows that the probability of failure satisfies

$$P_f < \frac{2t}{q^{1+v-t}}. \tag{7.28}$$

Proposition 7.5: The coding scheme described above can achieve both capacity expressions (7.19) and (7.20).

Proof: From (7.28), we see that achieving either of the limiting capacities amounts to setting a suitable v. To achieve (7.19), we set $v = t$ and let q grow. The resulting code will have the correct rate, namely, $R = (n-t)(m-t)$ in q-ary units, while the probability of failure will decrease exponentially with the field size in bits.

Alternatively, to achieve (7.20), we can choose some small $\epsilon > 0$ and set $v = (\tau+\epsilon)n$, where both $\tau = t/n$ and $\lambda = n/m$ are assumed fixed. By letting m grow, we obtain a probability of failure that decreases exponentially with m. The (normalized) gap to capacity of the resulting code will be

$$\begin{aligned}\bar{g} &\triangleq \lim_{m\to\infty} \overline{C_{\text{AMC}}} - R/(nm) \\ &= (1-\lambda\tau)(1-\tau) - (1-\lambda(\tau+\epsilon))(1-(\tau+\epsilon)) \\ &= \lambda\epsilon(1-(\tau+\epsilon)) + \epsilon(1-\lambda\tau) \\ &< \lambda\epsilon + \epsilon = (1+\lambda)\epsilon\end{aligned}$$

which can be made as small as we wish. ∎

7.4 The Additive-Multiplicative Matrix Channel

Let us now return to the LNCC described in (7.1). Since A is invertible, we can rewrite (7.1) as

$$Y = AX + DZ = A(X + A^{-1}DZ). \tag{7.29}$$

Now, since $\mathcal{T}_{n\times n}$ acts transitively on $\mathcal{T}_{n\times t}$, the channel law (7.29) is equivalent to

$$Y = A(X + W) \tag{7.30}$$

where $A \in \mathcal{T}_{n\times n}$ and $W \in \mathcal{T}_{n\times m,t}$ are chosen uniformly at random and independently from any other variables. We call (7.30) *the additive-multiplicative matrix channel* (AMMC).

7.4.1 Capacity

One of the main results of this section is the following theorem, which provides an upper bound on the capacity of the AMMC.

Theorem 7.6: For $n \leq m/2$, the capacity of the AMMC is upper bounded by

$$C_{\text{AMMC}} \leq (m-n)(n-t) + \log_q 4(1+n)(1+t).$$

Proof: Let $S = X + W$. By expanding $I(X, S; Y)$, and using the fact that X, S and Y form a Markov chain, in that order, we have

$$\begin{aligned} I(X;Y) &= I(S;Y) - I(S;Y|X) + \underbrace{I(X;Y|S)}_{=0} \\ &= I(S;Y) - I(W;Y|X) \\ &= I(S;Y) - H(W|X) + H(W|X,Y) \\ &= I(S;Y) - H(W) + H(W|X,Y) \qquad (7.32) \\ &\leq C_{\text{MMC}} - \log_q |\mathcal{T}_{n\times m,t}| + H(W|X,Y) \qquad (7.33) \end{aligned}$$

where (7.32) follows since X and W are independent.

We now compute an upper bound on $H(W|X,Y)$. Let $R = \mathsf{rank}\ Y$ and write $Y = G\bar{Y}$, where $G \in \mathcal{T}_{n\times R}$ and $\bar{Y} \in \mathcal{T}_{R\times m}$. Note that

$$X + W = A^{-1}Y = A^{-1}G\bar{Y} = A^*\bar{Y}$$

where $A^* = A^{-1}G$. Since $\bar{Y}$ is full-rank, it must contain an invertible $R\times R$ submatrix. By reordering columns if necessary, assume that the left $R \times R$ submatrix of $\bar{Y}$ is invertible. Write $\bar{Y} = \begin{bmatrix} \bar{Y}_1 & \bar{Y}_2 \end{bmatrix}$, $X = \begin{bmatrix} X_1 & X_2 \end{bmatrix}$ and $W = \begin{bmatrix} W_1 & W_2 \end{bmatrix}$, where $\bar{Y}_1$, X_1 and W_1 have R columns, and $\bar{Y}_2$, X_2 and W_2 have $m - R$ columns. We have

$$A^* = (X_1 + W_1)\bar{Y}_1^{-1} \quad \text{and} \quad W_2 = A^*\bar{Y}_2 - X_2.$$

It follows that W_2 can be computed if W_1 is known. Thus,

$$\begin{aligned} H(W|X,Y) &= H(W_1|X,Y) \leq H(W_1|R) \leq H(W_1|R=n) \\ &\leq \log_q \sum_{i=0}^{t} |\mathcal{T}_{n\times n,i}| \leq \log_q(t+1)|\mathcal{T}_{n\times n,t}| \end{aligned} \tag{7.36}$$

where (7.36) follows since W_1 may possibly be any $n \times n$ matrix with rank $\leq t$.

Applying this result in (7.33), and using (7.15) and (2.14), we have

$$\begin{aligned} I(X,Y) &\leq \log_q(n+1)\begin{bmatrix} m \\ n \end{bmatrix} + \log_q(t+1)\frac{|\mathcal{T}_{n\times t}|\begin{bmatrix} n \\ t \end{bmatrix}}{|\mathcal{T}_{n\times t}|\begin{bmatrix} m \\ t \end{bmatrix}} \\ &\leq \log_q(n+1)(t+1)\begin{bmatrix} m-t \\ n-t \end{bmatrix} \\ &\leq (m-n)(n-t) + \log_q 4(1+n)(1+t). \end{aligned} \tag{7.37}$$

where (7.37) follows from 2.13. ∎

We now develop a connection with the subspace approach of [18] (see also Section 4.3) that will be useful to obtain a lower bound on the capacity. From Section 7.2, we know that, in a multiplicative matrix channel, the receiver can only distinguish between transmitted subspaces. Thus, we can equivalently express

$$C_{\text{AMMC}} = \max_{p_X} I(\mathcal{X};\mathcal{Y})$$

where $\mathcal{X}$ and $\mathcal{Y}$ denote the row spaces of X and Y, respectively.

Using this interpretation, we can obtain the following lower bound on capacity.

Theorem 7.7: Assume $n \leq m$. For any $\epsilon \geq 0$, we have

$$C_{\text{AMMC}} \geq (m-n)(n-t-\epsilon t) - \log_q 4 - \frac{2tnm}{q^{1+\epsilon t}}.$$

In order to prove Theorem 7.7, we need a few lemmas.

Lemma 7.8: Let $X \in \mathbb{F}_q^{n\times m}$ be a matrix of rank k, and let $W \in \mathbb{F}_q^{n\times m}$ be a random matrix chosen uniformly among all matrices of rank t. If $k+t \leq \min\{n,m\}$, then

$$P[\mathsf{rank}\,(X+W) < k+t] < \frac{2t}{q^{\min\{n,m\}-k-t+1}}.$$

Proof: Write $X = X'X''$, where $X' \in \mathbb{F}_q^{n\times k}$ and $X'' \in \mathbb{F}_q^{k\times m}$ are full-rank matrices. We can generate W as $W = W'W''$, where $W' \in \mathcal{T}_{n\times t}$ and $W'' \in \mathcal{T}_{t\times m}$ are chosen uniformly at random and independently from each other. Then we have

$$X + W = X'X'' + W'W'' = \begin{bmatrix} X' & W' \end{bmatrix} \begin{bmatrix} X'' \\ W'' \end{bmatrix}.$$

Note that $\mathsf{rank}\,(X+W) = k+t$ if and only if the column spaces of X' and W' intersect trivially *and* the row spaces of X'' and W'' intersect trivially. Let P' and P'' denote the probabilities of these two events, respectively. By a simple counting argument, we have

$$\begin{aligned} P' &= \frac{(q^n - q^k)\cdots(q^n - q^{k+t-1})}{(q^n-1)\cdots(q^n-q^{t-1})} = \prod_{i=0}^{t-1} \frac{(1-q^{k-n+i})}{(1-q^{-n+i})} \\ &> \prod_{i=0}^{t-1}(1-q^{k-n+i}) \geq (1-q^{k-n+t-1})^t \geq 1 - tq^{k-n+t-1}. \end{aligned}$$

Similarly, we have $P'' > 1 - tq^{k-m+t-1}$. Thus,

$$\begin{aligned} P[\mathsf{rank}\,(X+W) < k+t] &< \frac{t}{q^{n-k-t+1}} + \frac{t}{q^{m-k-t+1}} \\ &\leq \frac{2t}{q^{\min\{n,m\}-k-t+1}}. \end{aligned}$$

■

For $\mathsf{dim}\,\mathcal{X} \leq n \leq m$, let $\mathcal{S}_{\mathcal{X},n}$ denote the set of all n-dimensional subspaces of $\mathbb{F}_q^m$ that contain a subspace $\mathcal{X} \subseteq \mathbb{F}_q^m$.

Lemma 7.9:

$$|\mathcal{S}_{\mathcal{X},n}| = \begin{bmatrix} m-k \\ n-k \end{bmatrix}_q$$

where $k = \mathsf{dim}\,\mathcal{X}$.

Proof: By the fourth isomorphism theorem [66], there is a bijection between $\mathcal{S}_{\mathcal{X},n}$ and the set of all $(n-k)$-dimensional subspaces of the quotient space $\mathbb{F}_q^m/\mathcal{X}$. Since $\mathsf{dim}\ \mathbb{F}_q^m/\mathcal{X} = m-k$, the result follows. ∎

We can now give a proof of Theorem 7.7.

Proof (of Theorem 7.7): Assume that X is selected from $\mathcal{T}_{n\times m,k}$, where $k = n-(1+\epsilon)t$ and $\epsilon \geq 0$. Define a random variable Q as

$$Q = \begin{cases} 1 & \text{if } \mathsf{dim}\ \mathcal{Y} = \mathsf{rank}\,(X+W) = k+t \\ 0 & \text{otherwise.} \end{cases}$$

Note that $\mathcal{X} \subseteq \mathcal{Y}$ when $Q = 1$.

By Lemma 7.9 and (2.12), we have

$$H(\mathcal{Y}|\mathcal{X}, Q=1) \leq \log_q |\mathcal{S}_{\mathcal{X},n'}| \leq (m-n')t + \log_q 4$$

where $n' = k+t$. Choosing X uniformly from $\mathcal{T}_{n\times m,k}$, we can also make $\mathcal{Y}$ uniform within a given dimension; in particular,

$$H(\mathcal{Y}|Q=1) = \log_q \begin{bmatrix} m \\ n' \end{bmatrix}_q \geq (m-n')n'.$$

It follows that

$$\begin{aligned} I(\mathcal{X};\mathcal{Y}|Q=1) &= H(\mathcal{Y}|Q=1) - H(\mathcal{Y}|\mathcal{X},Q=1) \\ &\geq (m-n')(n'-t) - \log_q 4 \\ &\geq (m-n)(n-t-\epsilon t) - \log_q 4. \end{aligned}$$

Now, using Lemma 7.8, we obtain

$$\begin{aligned} I(\mathcal{X};\mathcal{Y}) &= I(\mathcal{X};\mathcal{Y},Q) = I(\mathcal{X};Q) + I(\mathcal{X};\mathcal{Y}|Q) \\ &\geq I(\mathcal{X};\mathcal{Y}|Q) \\ &\geq P[Q=1]I(\mathcal{X};\mathcal{Y}|Q=1) \\ &\geq I(\mathcal{X};\mathcal{Y}|Q=1) - P[Q=0]nm \\ &\geq (m-n)(n-t-\epsilon t) - \log_q 4 - \frac{2tnm}{q^{\epsilon t+1}}. \end{aligned}$$

■

Note that, differently from the results of previous sections, Theorems 7.6 and 7.7 provide only upper and lower bounds on the channel capacity. Nevertheless, it is still possible to compute exact expressions for the capacity of the AMMC in certain limiting cases.

Corollary 7.10: For $0 < \lambda = n/m \leq 1/2$ and $\tau = t/n$, we have

$$\lim_{q\to\infty} C_{\text{AMMC}} = (m-n)(n-t) \tag{7.46}$$

$$\lim_{\substack{m\to\infty \\ n=\lambda m \\ t=\tau n}} \overline{C_{\text{AMMC}}} = (1-\lambda)(1-\tau). \tag{7.47}$$

Proof: The fact that the values in (7.46) and (7.47) are upper bounds follows immediately from Theorem 7.6. The fact that (7.46) is a lower bound follows immediately from Theorem 7.7 by setting $\epsilon = 0$. To obtain (7.47) from Theorem 7.7, it suffices to choose ϵ such that $1/\epsilon$ grows sublinearly with m, e.g., $\epsilon = 1/\sqrt{m}$. ■

Differently from the MMC and the AMC, successful decoding in the AMMC does not (necessarily) allow recovery of all sources of channel uncertainty—in this case, the matrices A and W. In general, for every observable (X, Y) pair, there are many valid A and W such that $Y = A(X + W)$. Such coupling between A and W is reflected in extra term $H(W|X, Y)$ in (7.32), which provides an additional rate of roughly $(2n - t)t$ as compared to the straightforward lower bound $C_{\text{AMMC}} \geq C_{\text{MMC}} - \log_q |\mathcal{T}_{n\times m,t}| \approx (m - n)n - (n + m - t)t$.

Remark: In [23], the problem of finding the capacity of the AMMC was addressed using a specific form of transmission matrices that contained an $n \times n$ identity header. This approach, in fact, turns the channel into an AMC after stripping off the headers. It is instructive to observe that the capacity expression of [23], $\overline{C} = (1 - \lambda - \lambda\tau)(1 - \tau)$, corresponds exactly to (7.20) after accounting for the extra redundancy in the header (i.e., replacing m with $m - n$). We might, therefore, interpret that [23] has computed the capacity of the infinite-rank AMC. The statement of Proposition 7.4 is, nevertheless, (slightly) more general than that of [23]. Note that, as the input distribution assumed in [23] is not capacity-achieving, the capacity of the infinite-rank AMMC (7.47) is strictly larger than the expression obtained in [23]. □

7.4.2 A Coding Scheme

We now propose an efficient coding scheme that can asymptotically achieve (7.46) and (7.47). The scheme is based on a combination of channel sounding and error trapping strategies.

For a data matrix $U \in \mathbb{F}_q^{(n-v)\times(m-n)}$, where $v \geq t$, let the corresponding codeword be

$$X = \begin{bmatrix} 0 \\ \bar{X} \end{bmatrix} = \begin{bmatrix} 0_{v\times v} & 0_{v\times(n-v)} & 0_{v\times(m-n)} \\ 0_{(n-v)\times v} & I_{(n-v)\times(n-v)} & U \end{bmatrix}.$$

Note that the all-zero matrices provide the error traps, while the identity matrix corresponds to the pilot symbols. Clearly, the rate of this scheme is $R = (n - v)(m - n)$.

Write the noise matrix W as

$$W = BZ = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \begin{bmatrix} Z_1 & Z_2 & Z_3 \end{bmatrix}$$

where $B_1 \in \mathbb{F}_q^{v\times t}$, $B_2 \in \mathbb{F}_q^{(n-v)\times t}$, $Z_1 \in \mathbb{F}_q^{t\times v}$, $Z_2 \in \mathbb{F}_q^{t\times(n-v)}$ and $Z_3 \in \mathbb{F}_q^{t\times(m-n)}$. The

auxiliary matrix S is then given by

$$S = X + W = \begin{bmatrix} B_1 Z_1 & B_1 Z_2 & B_1 Z_3 \\ B_2 Z_1 & I + B_2 Z_2 & U + B_2 Z_3 \end{bmatrix}.$$

Similarly as in Section 7.3, we define that the error trapping is successful if $\mathsf{rank}\ B_1 Z_1 = t$. Assume that this is the case. From Section 7.3, there exists some matrix $T \in \mathcal{T}_{n \times n}$ such that

$$TS = \begin{bmatrix} B_1 Z_1 & B_1 Z_2 & B_1 Z_3 \\ 0 & I & U \end{bmatrix} = \begin{bmatrix} B_1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} Z_1 & Z_2 & Z_3 \\ 0 & I & U \end{bmatrix}.$$

Note further that

$$\mathsf{RRE}\left(\begin{bmatrix} Z_1 & Z_2 & Z_3 \\ 0 & I & U \end{bmatrix} \right) = \begin{bmatrix} \tilde{Z}_1 & 0 & \tilde{Z}_3 \\ 0 & I & U \end{bmatrix}$$

for some $\tilde{Z}_1 \in \mathbb{F}_q^{t \times v}$ in RRE form and some $\tilde{Z}_3 \in \mathbb{F}_q^{t \times (m-n)}$. It follows that

$$\begin{aligned} \mathsf{RRE}\,(S) &= \mathsf{RRE}\left(\begin{bmatrix} B_1 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} Z_1 & Z_2 & Z_3 \\ 0 & I & U \end{bmatrix} \right) \\ &= \begin{bmatrix} \tilde{Z}_1 & 0 & \tilde{Z}_3 \\ 0 & I & U \\ 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

where the bottom $v - t$ rows are all-zeros.

Since A is invertible, we have $\mathsf{RRE}\,(Y) = \mathsf{RRE}\,(S)$, from which U can be readily obtained. Thus, decoding amounts to performing Gauss-Jordan elimination on Y. It follows that the complexity of the scheme is $O(n^2 m)$ operations in $\mathbb{F}_q$.

The probability that the error trapping is not successful, i.e., $\mathsf{rank}\ B_1 Z_1 < t$, was computed in Section 7.3. Let $\hat{A}$ correspond to the first n columns of Y. Note that $\mathsf{rank}\ B_1 Z_1 = t$ if and only if $\mathsf{rank}\ \hat{A} = n - v + t$. Thus, when the error trapping is not successful, the receiver can easily detect this event by looking at $\mathsf{RRE}\,(Y)$ and then declare

a decoding failure. It follows that the scheme has zero error probability and probability of failure given by (7.28).

Theorem 7.11: The proposed coding scheme can asymptotically achieve (7.46) and (7.47).

Proof: Using (7.28) and the same argument as in the proof of Proposition 7.5, we can set a suitable v in order to achieve arbitrarily low gap to capacity while maintaining an arbitrary low probability of failure, for both cases where $q \to \infty$ or $m \to \infty$. ■

7.5 Extensions

In this section, we discuss possible extensions of the results and models presented in the previous sections.

7.5.1 Dependent Transfer Matrices

As discussed in Section 7.4, the AMMC is equivalent to a channel of the form (7.3) where $A \in \mathcal{T}_{n\times n}$ and $D \in \mathcal{T}_{n\times t}$ are chosen uniformly at random and independently from each other. Suppose now that the channel is the same, except for the fact that A and D are not independent. It should be clear that the capacity of the channel cannot be smaller than that of the AMMC. For instance, one can always convert this channel into an AMMC by employing randomization at the source. (This is, in fact, a natural procedure in any random network coding system.) Let $X = TX'$, where $T \in \mathcal{T}_{n\times n}$ is chosen uniformly at random and independent from any other variables. Then $A' = AT$ is uniform on $\mathcal{T}_{n\times n}$ and independent from D. Thus, the channel given by $Y = A'X' + DZ$ is an AMMC.

Note that our coding scheme does not rely on any particular statistics of A given X and W (except the assumption that A is invertible) and therefore works unchanged in this case.

7.5.2 Transfer Matrix Invertible but Nonuniform

The model for the AMMC assumes that the transfer matrix $A \in \mathcal{T}_{n\times n}$ is chosen uniformly at random. In a realistic network coding system, the transfer matrix may be a function of both the network code and the network topology, and therefore may not have a uniform distribution. Consider the case where A is chosen according to an arbitrary probability distribution on $\mathcal{T}_{n\times n}$. It should be clear that the capacity can only increase as compared with the AMMC, since less "randomness" is introduced in the channel. The best possible situation is to have a constant A, in which case the channel becomes exactly an AMC.

Again, note that our coding scheme for the AMMC is still applicable in this case.

7.5.3 Nonuniform Packet Errors

When expressed in the form (7.3), the models for both the AMC and the AMMC assume that the matrix Z is uniformly distributed on $\mathcal{T}_{n\times t}$. In particular, each error packet is uniformly distributed on $\mathbb{F}_q^{1\times m} \setminus \{0\}$. In a realistic situation, however, it may be the case that error packets of low weight are more likely to occur. Consider a model identical to the AMC or the AMMC except for the fact that the matrix Z is chosen according to an arbitrary probability distribution on $\mathcal{T}_{t\times m}$. Once again, it should be clear that the capacity can only increase. Note that the exact capacity in Proposition 7.4 and the upper bound of Theorem 7.6 can be easily modified to account for this case (by replacing $\log_q |\mathcal{T}_{n\times m,t}|$ with the entropy of W).

Although our coding scheme in principle does not hold in this more general case, we can easily convert the channel into an AMC or AMMC by applying a random transformation at the source (and its inverse at the destination). Let $X = X'T$, where $T \in \mathcal{T}_{m\times m}$ is chosen uniformly at random and independent from any other variables. Then

$$Y' = YT^{-1} = (AX + DZ)T^{-1} = AX' + DZ'$$

where $Z' = ZT^{-1}$. Since $\mathcal{T}_{m\times m}$ acts (by right multiplication) transitively on $\mathcal{T}_{t\times m}$, we have that Z' is uniform on $\mathcal{T}_{t\times m}$. Thus, we obtain precisely an AMMC (or AMC) and the assumptions of our coding scheme hold.

Note, however, that, depending on the error model, the capacity may be much larger than what can be achieved by the scheme described above. For instance, if the rows of Z are constrained to have weight at most s (otherwise chosen, say, uniformly at random), then the capacity would increase by approximately $\left(m - s - \log_q \binom{m}{s}\right) t$, which might be a substantial amount if s is small.

7.5.4 Error Matrix with Variable Rank ($\leq t$)

The model we considered for the AMC and the AMMC assumes an error matrix W whose rank is known and equal to t. It is useful to consider the case where $\mathsf{rank}\, W$ is allowed to vary, while still bounded by t. More precisely, we assume that W is chosen uniformly at random from $\mathcal{T}_{n\times m,R}$, where $R \in \{0, \ldots, t\}$ is a random variable with probability distribution $P[R = r] = p_r$.

Since

$$\begin{aligned} H(W) = H(W, R) &= H(R) + H(W|R) \\ &= H(R) + \sum_r p_r H(W|R = r) \\ &= H(R) + \sum_r p_r \log_q |\mathcal{T}_{n\times m,r}| \\ &\leq H(R) + \log_q |\mathcal{T}_{n\times m,t}|, \end{aligned}$$

we conclude that the capacities of the AMC and the AMMC may be reduced by at most $H(R) \leq \log_q(t + 1)$. This loss is asymptotically negligible for large q and/or large m, so the expressions (7.19), (7.20), (7.46) and (7.47) remain unchanged.

The steps for decoding and computing the probability of error trapping failure also remain the same, provided we replace t by R. The only difference is that now decoding

errors may occur. More precisely, suppose that $\mathsf{rank}\ B_1 Z_1 = t' < t$. A necessary condition for success is that $\mathsf{rank}\ B_1 Z = \mathsf{rank}\ B Z_1 = t'$. If this condition is not satisfied, then a decoding failure is declared. However, if the condition is true, then the decoder cannot determine whether $t' = R < t$ (an error trapping success) or $t' < R \leq t$ (an error trapping failure), and must proceed assuming the former case. If the latter case turns out to be true, we would have an undetected error. Thus, for this model, the expression (7.28) gives a bound on the probability that decoding is not successful, i.e., that either an error or a failure occurs.

7.5.5 Infinite Packet Length or Infinite Batch Size

We now extend our results to the infinite-packet-length AMC and AMMC and the infinite-batch-size AMC. (Note that, as pointed out in Section 7.2, there is little justification to consider an infinite-batch-size AMMC.) From the proof of Propositon 7.4 and the proof of Corollary 7.10, it is straightforward to see that

$$\lim_{m\to\infty} \overline{C_{\mathrm{AMMC}}} = \lim_{m\to\infty} \overline{C_{\mathrm{AMC}}} = (n-t)/n$$
$$\lim_{n\to\infty} \overline{C_{\mathrm{AMC}}} = (m-t)/m.$$

It is *not* straightforward, however, to obtain capacity-achieving schemes for these channels. The schemes described in Sections 7.3 and 7.4 for the infinite-rank AMC and AMMC, respectively, use an error trap whose size (in terms of columns *and* rows) grows proportionally with m (or n). While this is necessary for achieving vanishingly small error probability, it also implies that these schemes are not suitable to the infinite-packet-length channel (where $m \to \infty$ but not n) or the infinite-batch-size channel (where $n \to \infty$ but not m).

In these situations, the proposed schemes can be adapted by replacing the data matrix and part of the error trap with an MRD code. Consider first an infinite-packet-length

AMC. Let the transmitted matrix be given by

$$X = \begin{bmatrix} 0_{n\times v} & x \end{bmatrix} \tag{7.54}$$

where $x \in \mathbb{F}_q^{n\times(m-v)}$ is a codeword of a matrix code $\mathcal{C}$. If (column) error trapping is successful then, under the terminology of Section 5.2.2, the decoding problem for $\mathcal{C}$ amounts to the correction of t *erasures*. As we know from Section 5.2.3, for $m - v \geq n$, an MRD code $\mathcal{C} \subseteq \mathbb{F}_q^{n\times(m-v)}$ with rate $(n-t)/n$ can correct exactly t erasures (with zero probability of error) [27]. Thus, decoding fails if and only if column trapping fails.

Similarly, for an infinite-batch-size AMC, let the transmitted matrix be given by

$$X = \begin{bmatrix} 0_{v\times m} \\ x \end{bmatrix}$$

where $x \in \mathbb{F}_q^{(n-v)\times m}$ is a codeword of a matrix code $\mathcal{C}$. If (row) error trapping is successful then, under the terminology of Section 5.2.2, the decoding problem for $\mathcal{C}$ amounts to the correction of t *deviations*. As we know from Section 5.2.3, for $n - v \geq m$, an MRD code $\mathcal{C} \subseteq \mathbb{F}_q^{(n-v)\times m}$ with rate $(m-t)/m$ can correct exactly t deviations (with zero probability of error). Thus, decoding fails if and only if row trapping fails.

Finally, for the infinite-packet-length AMMC, it is sufficient to prepend to (7.54) an identity matrix, i.e.,

$$X = \begin{bmatrix} I_{n\times n} & 0_{n\times v} & x \end{bmatrix}.$$

The same reasoning as for the infinite-packet-length AMC applies here, and the decoder in Chapter 5 is also applicable in this case.

For more details on the decoding of an MRD code combined with an error trap, we refer the reader to [67]. The decoding complexity is in $O(tn^2m)$ and $O(tm^2n)$ (whichever is smaller), and could be further simplified using the approach of Chapter 6.

In all cases, the schemes have probability of error upper bounded by t/q^{1+v-t} and therefore are capacity-achieving.

Chapter 8

Secure Network Coding

In this chapter, we divert our attention from the error control problem and instead focus on the security issues that may arise when using network coding. We are interested in addressing the problem of securing a linear network coding communication system against a wiretapper adversary.

The scenario considered here has its roots in the wiretap channel II of Ozarow and Wyner [19]. In this channel, a source transmits n symbols to a destination, μ of which are observed by a wiretapper. The problem is how to encode a message into the n transmitted symbols in such a way that the wiretapper gets no information about the message (information-theoretic security). An optimal solution, proposed by Ozarow and Wyner, is to use a coset coding scheme. The idea is to use the message to select a coset of an MDS code, and transmit a random word within that coset. The randomization does not affect the destination, but ensures that a wiretapper who observes only $\mu < n$ symbols will not be able to uniquely identify the coset.

The wiretap channel II can be generalized to a network coding scenario [2–4] by regarding the symbols as packets flowing through a network that implements linear network coding. In this case, the wiretapper observes μ linear combinations of the original packets. This problem, which became known as secure network coding, was first studied

by Cai and Yeung [20], and further investigated by Feldman *et al.* [21]. The connection with the wiretap channel II was observed by Rouayheb and Soljanin [22].

In all of these works, the proposed solutions are essentially analogous to Ozarow-Wyner's coset coding scheme. However, the analogy breaks down in the fact that the network may "undo" some of the randomization performed by the source, and actually help the wiretapper. Thus, the network code and the outer code must be jointly designed to prevent this "de-randomization." A side-effect of this approach is that the field size for network coding is required to be significantly large.

In Section 8.2, we propose a *universal* approach to network coding security that allows a completely independent design. The idea (essentially a vector-linear approach) is to use coset coding over an extension field of the field used for linear network coding operations. This allows a complete separation between a "black-box" implementing network coding and an outer security scheme. In particular, the network code need not be known, so a our scheme is compatible with random network coding. A consequence of this approach is that the field size for network coding can return to its usual values, i.e., as if security were not a concern. Our scheme makes crucial use of the properties of MRD codes and is shown to be optimal in a certain sense.

A drawback of enforcing security on a wiretap network is that the communication rate must be reduced from n to $n-\mu$, i.e., one "randomization" packet has to be spent for each packet observed by the wiretapper. Hoping to overcome this rate barrier, Bhattad and Narayanan [24] have defined a weaker notion of security that is practically very appealing. Roughly speaking, a system is considered weakly secure if the wiretapper gets no information about each packet individually, while still potentially obtaining "meaningless" information about mixtures of packets. Under these more relaxed security requirements, the maximum rate of n packets per message can be obtained. The approach of [24], however, suffers from the same drawback as [20–22] in that the outer code has to be carefully designed, and the field size required for network coding may be significantly

large. In Section 8.3, we show that a universal (i.e., network-code-independent) weakly secure scheme can be obtained via an approach similar to that of Section 8.2.

A third contribution of this chapter, described in Section 8.4, is the design of a universal scheme that can provide not only security but also protection against errors. More precisely, we assume that the wiretapper is able not only to eavesdrop, but also to inject up to t erroneous packets anywhere in the network. In this case, we can guarantee security and reliability as long as $\mu + 2t < n$. Other works have attained the same goals under even looser conditions [16, 68], but these schemes were not universal. Once again, advantages of our approach include the smaller field size and the simplicity of the design.

Finally, Section 8.5 discusses implementation aspects of our proposed schemes, which are shown to be computationally efficient for both encoding and decoding.

8.1 The Wiretap Channel II

Let F be an arbitrary alphabet. Consider a communication system consisting of a source, a destination and a wiretapper. The source produces a message $S = (S_1, S_2, \ldots, S_k)$, with symbols $S_1, \ldots, S_k$ drawn from F, and encodes this message as a word $X = (X_1, \ldots, X_n)$, $X_i \in F$. This word is transmitted over a noiseless channel and received by the destination. The wiretapper has access to μ symbols of X, represented as the word $W = (X_i,\, i \in \mathcal{I})$, where $\mathcal{I} \subseteq \{1, \ldots, n\}$ has cardinality μ. The goal of the system is for the source to communicate the message to the destination in such a way that the wiretapper cannot obtain any information about S from any possible set of μ intercepted symbols. More precisely, the conditions for secure communication are

$$H(S|X) = 0 \tag{8.1}$$

$$I(S; W) = 0, \quad \forall \mathcal{I} \colon |\mathcal{I}| = \mu. \tag{8.2}$$

The objective is to design a (necessarily probabilistic) encoding of S into X such that conditions (8.1) and (8.2) are satisfied.

It can be shown that the maximum number of symbols that can be securely communicated is upper bounded by $H(S) \leq n - \mu$ (where the log is taken to the base $|F|$). This maximum rate can be achieved, if F is a finite field of sufficiently large cardinality, by using Ozarow-Wyner coset coding scheme [19]. The scheme can be described as follows.

Let $k = n - \mu$ and let $\mathcal{C}$ be a linear (n, μ) MDS code over F with parity-check matrix $H \in F^{k \times n}$. Encoding is performed by choosing uniformly at random some $X \in F^n$ such that $S = HX$, where S and X are taken as column vectors. In other words, each message is viewed as a syndrome specifying a coset of $\mathcal{C}$, and the transmitted vector is randomly chosen among the elements of that coset. Upon reception of X, decoding is performed by simply computing the syndrome $S = HX$.

It is useful to review the proof that this scheme is secure. By expanding $I(S, X; W)$, we have

$$\begin{aligned} I(S;W) &= I(X;W) + I(S;W|X) - I(X;W|S) \\ &= H(W) - H(W|X,S) - H(X|S) + H(X|S,W) \\ &= H(W) - H(X|S) + H(X|S,W) && (8.3) \\ &= H(W) - \mu + H(X|S,W) && (8.4) \\ &\leq H(X|S,W) && (8.5) \end{aligned}$$

where (8.3) follows since W is a function of X, (8.4) follows since X is chosen uniformly at random among $|F|^\mu$ possibilities, and (8.5) follows since $H(W) \leq \mu$. The last step is to show that $H(X|S,W) = 0$, i.e., that a word X can be determined given its coset, S, and μ "non-erased" symbols, W. But this follows from the fact that $\mathcal{C}$, as well as all of its cosets, is an (n, μ) MDS code.

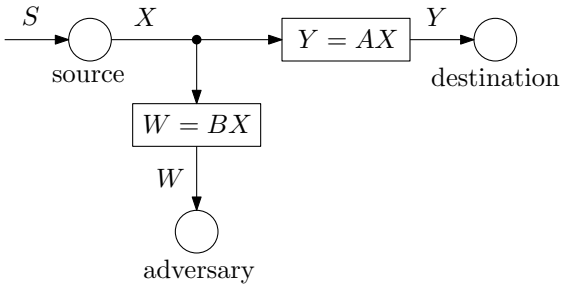

Figure 8.1: Linear network coding channel with a wiretapper adversary.

8.2 Security for Wiretap Networks

For the remainder of this chapter we will use the following notation and terminology. Packets transmitted over a network implementing linear network coding are drawn from a set F, which is assumed to be an m-dimensional vector space over a finite field $\mathbb{F}_q$. A network using a network code that is feasible for the transmission of n packets from source to (each) destination is called an $(n, m, \mathbb{F}_q)$-*linear multicast network*.

8.2.1 Wiretap Networks

The wiretap problem of Section 8.1 can be generalized to a linear network coding scenario by introducing a wiretapper in the setup of Section 3.1. The wiretapper can eavesdrop on μ links, collectively represented by the set $\mathcal{I}$; in this case, we will say that the network *is subject to μ observations*. Since linear network coding is used, the packets observed by the wiretapper are given by $W = BX \in F^\mu$, where $B \in \mathbb{F}_q^{\mu \times n}$ is the matrix whose rows are the global coding vectors associated with the edges in $\mathcal{I}$. Additionally, we assume that the "true" source message is given by $S = \begin{bmatrix} S_1 & S_2 & \cdots & S_k \end{bmatrix}^T$, $S_i \in F$, which is then encoded into X for transmission over the network. The system is depicted in Fig. 8.1. Similarly as before, the conditions for secure communication are given by

$$H(S|Y) = 0 \tag{8.6}$$

$$I(S; W) = 0, \quad \forall \mathcal{I} \colon |\mathcal{I}| = \mu. \tag{8.7}$$

The problem, now, is how to design an encoding from S to X *and* a linear network code such that (8.6) and (8.7) are satisfied.

Note that, since making a row of B linear dependent of the others cannot possibly increase $I(S;W)$, we have that a necessary and sufficient (and therefore equivalent) condition for (8.7) is given by

$$I(S;W) = 0, \quad \forall \mathcal{I}\colon |\mathcal{I}| = \mu, \, \mathsf{rank}\, B = \mu. \tag{8.8}$$

It is reasonable to assume that the network code used is feasible, i.e., that X can be recovered from Y. If this is the case, then $H(X|Y) = 0$ and thus condition (8.6) can be replaced by (8.1). The problem then becomes very similar to that of Section 8.1, the only difference being that the matrix B consists of row vectors over $\mathbb{F}_q$ rather then rows from an identity matrix.

8.2.2 Security via Linear MDS Codes

The similarity mentioned above suggests that the same techniques that are useful for the wiretap channel could also be useful for a wiretap network; namely, the Ozarow-Wyner coset coding scheme discussed in Section 8.1. Note, however, that for that scheme to be applicable, F must be a field. The case $F = \mathbb{F}_q$ was studied by Rouayheb and Soljanin [22], who showed that secure communication at the maximum rate $k = n - \mu$ can be achieved with a coset coding scheme if the network code is chosen to satisfy certain constraints.

Let H be the parity-check matrix of a linear (n, μ) MDS code over F, and suppose that a coset coding scheme based on H is used. Note that (8.1) follows immediately from the fact that $S = HX$. In order to satisfy (8.8), we will make use of the following proposition. We state the result in somewhat general terms since it will be useful later on.

Proposition 8.1: Let F be a field, $H \in F^{k\times n}$ and $B \in F^{\mu\times n}$. Let $\mathcal{S} = \{Hx\colon x \in F^n\}$ and $\mathcal{X}_s = \{x \in F^n\colon s = Hx\}$. Let $S \in \mathcal{S}$, $X \in F^n$ and $W = BX$ be random variables.

1) If X is uniform over $\mathcal{X}_S$ given S, then

$$\mathsf{rank}\begin{bmatrix} H \\ B \end{bmatrix} = \mathsf{rank}\, H + \mathsf{rank}\, B \implies I(S;W) = 0.$$

2) If S is uniform, then

$$I(S;W) = 0 \implies \mathsf{rank}\begin{bmatrix} H \\ B \end{bmatrix} = \mathsf{rank}\, H + \mathsf{rank}\, B.$$

Proof: See the Appendix. ∎

A lightly weaker version of item 1) of Proposition 8.1 was proved in [22]. As we can see from Proposition 8.1, condition (8.8) will be satisfied if the following condition holds:

$$\mathsf{rank}\begin{bmatrix} H \\ B \end{bmatrix} = n, \quad \forall \mathcal{I}\colon |\mathcal{I}| = \mu,\, \mathsf{rank}\, B = \mu. \tag{8.11}$$

Equivalently, we must ensure that no linear combination of at most μ global coding vectors belongs to the row space of H.

It follows from this result of Rouayheb and Soljanin that secure multicast communication can be achieved in two steps: first, designing a coset coding scheme based on an MDS code, and then designing a linear network code so as to satisfy the above constraint.

Note that there is still a potentially undesirable coupling between the design of the coset coding scheme and the design of the network code, as the latter must take H into account. Moreover, the scheme is not compatible with random network coding, since the network code must be carefully chosen so as to satisfy a large number of global constraints.

8.2.3 Universal Security via MRD Codes

Dealing with the coupling mentioned above motivates the following definition of a universal scheme.

Definition 8.1: A secure communication scheme for a linear multicast network is called *universal* if it provides security and reliability for *any* choice of a feasible linear network code. That is, the scheme must satisfy conditions

$$H(S|X) = 0 \tag{8.12}$$

$$I(S;W) = 0, \quad \forall B. \tag{8.13}$$

By definition, a universal scheme is an outer code that can be designed independently and applied on top of any (feasible) network code. The specific wiretapping matrix B is irrelevant as long as at most μ observations are made by the wiretapper. As a consequence, any such scheme is naturally compatible with random network coding.

We now present our construction of a universal secure communication scheme. Our crucial assumption is that m, the dimension of F as a vector space over $\mathbb{F}_q$, is at least n; in other words, each packet must consist of $m \geq n$ symbols from $\mathbb{F}_q$.

Our scheme is based on two main ideas: first, we regard F as an extension field of $\mathbb{F}_q$ of degree m, denoted $\mathbb{F}_{q^m}$; second, we replace the MDS code in Ozarow-Wyner coset coding scheme by an MRD code over $\mathbb{F}_{q^m}$. Note that, since coset encoding/decoding is performed only at source/destination nodes, setting F to be an extension field of $\mathbb{F}_q$ does not interfere with the underlying linear network coding operations, which are still performed over the base field $\mathbb{F}_q$.

Let $k = n - \mu$ and let H be the parity-check matrix of a linear (n, μ) MRD code over $F = \mathbb{F}_{q^m}$. Encoding and decoding of the source message S is performed as described in

Section 8.1. With respect to security, we must ensure that

$$\mathsf{rank} \begin{bmatrix} H \\ B \end{bmatrix} = n, \quad \forall B\colon \mathsf{rank}\, B = \mu.$$

Note that, differently from Section 8.2.2, the matrix H is now defined over $\mathbb{F}_{q^m}$, while the matrix B still has all its entries in $\mathbb{F}_q$. This is the fundamental distinction of our approach, which allows a complete decoupling of the outer code and the network code.

Our main result in this section is a consequence of the following theorem.

Theorem 8.2: Let $\mathcal{C}$ be a linear (n, μ) code over $\mathbb{F}_{q^m}$ with parity-check matrix $H \in \mathbb{F}_{q^m}^{(n-\mu)\times n}$. Then $\mathcal{C}$ is an MRD code with $m \geq n$ if and only if the matrix

$$\begin{bmatrix} H \\ B \end{bmatrix} \tag{8.15}$$

is nonsingular for any full-rank $B \in \mathbb{F}_q^{\mu\times n}$.

Proof: Let M denote the matrix in (8.15). We will show that $d_{\mathrm{R}}(\mathcal{C}) \leq n-\mu$ if and only if there exists some full-rank B such that M is singular.

First, assume that $d_{\mathrm{R}}(\mathcal{C}) \leq n-\mu$. Then there exists some nonzero $x \in \mathcal{C}$ such that $\mathsf{rank}(x) \leq n-\mu$. This implies that there exists some full-rank $B \in \mathbb{F}_q^{\mu\times n}$ such that $Bx = 0$. Thus, $Mx = 0$ and M is singular.

Conversely, assume that M is singular for some full-rank $B \in \mathbb{F}_q^{\mu\times n}$. Then there must be some nonzero $x \in F^n$ such that $Mx = 0$. But this implies that $Hx = 0$, i.e., $x \in \mathcal{C}$, and $Bx = 0$, i.e., $\mathsf{rank}(x) \leq n-\mu$. Thus, $d_{\mathrm{R}}(\mathcal{C}) \leq n-\mu$. ∎

We can now state the main result of this section.

Theorem 8.3: Consider an $(n, m, \mathbb{F}_q)$ linear multicast network subject to μ observations. Let $H \in \mathbb{F}_{q^m}^{(n-\mu)\times n}$ be the parity-check matrix of a linear (n, μ) code over $\mathbb{F}_{q^m}$. Universal secure communication at the maximum rate $k = n-\mu$ can be achieved with a

coset coding scheme based on H if and only if the code defined by H is an MRD code with $m \geq n$.

Proof: The achievability has been proved. The necessity part follows by combining item 2) of Proposition 8.1 with Theorem 8.2. ■

We now proceed to proving that our scheme is optimal, i.e., that there is no universal scheme with $m < n$. First we need a couple of technical lemmas.

Lemma 8.4: Let $S \in \mathcal{S}$, $W \in \mathcal{W}$ and $X \in \mathcal{X}$ be discrete random variables with $S = f(X)$ and $W = g(X)$. Suppose S is uniform and $|\{x \in \mathcal{X} \colon g(x) = w\}| = |\mathcal{S}|$, $\forall w \in \mathcal{W}$. Then $I(S;W) = 0$ implies $H(X|S,W) = 0$.

Proof: See the Appendix. ■

Lemma 8.5: Let $\mathcal{C} \subseteq F^n$ be a rank-metric code over $F/\mathbb{F}_q$. For $B \in \mathbb{F}_q^{\mu \times n}$, let $g_B \colon \mathcal{C} \to F^\mu$ be given by $x \mapsto Bx$. Then g_B is injective for all full-rank B if and only if $d_R(\mathcal{C}) \geq n - \mu + 1$.

Proof: Suppose $d_{\mathrm{R}}(\mathcal{C}) \leq n-\mu$. Then there exist distinct $x, y \in \mathcal{C}$ such that $\mathsf{rank}(y-x) \leq n - \mu$. This implies that there exists some full-rank $B \in \mathbb{F}_q^{\mu \times n}$ such that $B(y - x) = 0$, i.e., $Bx = By$. Thus, g_B is not injective for this choice of B.

Conversely, suppose g_B is not injective for some full-rank $B \in \mathbb{F}_q^{\mu \times n}$. Then there exist distinct $x, y \in \mathcal{C}$ such that $B(y - x) = 0$. This implies that $\mathsf{rank}(y - x) \leq n - \mu$. Thus, $d_{\mathrm{R}}(\mathcal{C}) \leq n - \mu$. ■

We are now ready to prove our converse result.

Theorem 8.6: Consider an $(n, m, \mathbb{F}_q)$ linear multicast network subject to μ observations. Universal secure communication at the maximum rate $k = n - \mu$ is possible only if $m \geq n$.

Proof: Assume that a universal secure communication scheme is used that achieves the maximum rate. Then $H(S|Y) = 0$, which implies that $H(S|X) = 0$ (since Y is a function of X). Thus, we may assume that $S = f(X)$.

The fact that maximum rate is achieved implies that $H(S) = k = n - \mu$, so S must be uniform over F^k. Also, for any full-rank B, we have

$$|\{x \in F^n \colon Bx = w\}| = |F|^{n-\mathsf{rank}\ B} = |F|^{n-\mu} = |F|^k$$

for all $w \in F^\mu$. The conditions of Lemma 8.4 hold and, since $I(S;W) = 0$, we conclude that $H(X|S,W) = 0$.

Now, let $\mathcal{X}_s = \{x \in F^n \colon f(x) = s\}$, for all $s \in F^k$. The condition $H(X|S,W) = 0$ implies that X must be uniquely determined given $W = BX$ and the indication that $X \in \mathcal{X}_S$. Note that this must hold for any full-rank B. According to Lemma 8.5, this requires each $\mathcal{X}_s$ to be a rank-metric code with $d_{\mathrm{R}}(\mathcal{X}_s) \geq n - \mu + 1$. On the other hand, the collection of $\mathcal{X}_s$ form a $|F^k|$-partition of F^n, with average cardinality $|F^n|/|F^k| = q^{m\mu}$. Thus, there must be at least one $\mathcal{X}_s$ with cardinality at least $q^{m\mu}$. From the Singleton bound (2.23), we see that this can only happen if $m \geq n$. ∎

The existence of universal schemes implies that, to incorporate end-to-end security in an existing linear network coding system, the field on which the internal nodes operate does not need to be enlarged. This is in sharp contrast with the previous approaches. Although we still require the network to transport packets of sufficiently large size ($m \geq n$), this requirement is usually easily satisfied in practice.

The following example illustrates the results obtained in this section.

Example 8.1: Let $q = 2$, $m = n = 3$, $\mu = 2$ and $k = n - \mu = 1$. Let $F = \mathbb{F}_{2^3}$ be generated by a root of $p(x) = x^3 + x + 1$, which we denote by α. According to [37], one possible (n, μ) MRD code over $\mathbb{F}_{q^m}$ has parity-check matrix $H = \begin{bmatrix} 1 & \alpha & \alpha^2 \end{bmatrix}$.

To form X, we can choose $X_2, X_3 \in \mathbb{F}_{q^m}$ uniformly at random and set X_1 to satisfy

$$S = HX = X_1 + \alpha X_2 + \alpha^2 X_3.$$

Note that X can be transmitted over any network that uses a feasible linear network code. The specific network code used is irrelevant as long as each destination node is able to recover X.

Now, suppose that the wiretapper intercepts $W = BX$, where

$$B = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Then

$$\begin{aligned} W = B \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} S + \alpha X_2 + \alpha^2 X_3 \\ X_2 \\ X_3 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} S + \begin{bmatrix} \alpha & 1+\alpha^2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} X_2 \\ X_3 \end{bmatrix}. \end{aligned}$$

This is a linear system with 3 variables and 2 equations over $\mathbb{F}_{q^m}$. Note that, given S, there is exactly one solution for (X_2, X_3) for each value of W. Thus, $\mathsf{Pr}(W|S) = 1/8^2$, $\forall S, W$, from which follows that S and W are independent. □

8.3 Weak Security for Wiretap Networks

The notion of security described in Section 8.2 is *strong* in the sense that the wiretapper gets no information whatsoever about the message, but it comes at the expense of rate. If we insist on transmitting at a higher rate, the wiretapper will certainly obtain *some* information about the message. However, it may still be the case that this information is completely meaningless to the wiretapper. For instance, if $n = 2$ and $\mu = 1$, then the

maximum rate for secure communication is $k = n - \mu = 1$. If we insist on using $k = 2$ (and assuming that $S = HX$), then the wiretapper will be able to obtain some linear combination of the message packets, say $W = BH^{-1}S = a_1S_1 + a_2S_2$. As long as both a_1 and a_2 are nonzero, the wiretapper would still be confused about the value of each message packet individually.

More formally, the amount of *meaningful information* that the wiretapper obtains about S is defined as $\sum_{i=1}^{k} I(S_i; W)$. A communication scheme in which the wiretapper gets no meaningful information about the message is called *weakly secure* [24]. The conditions for weakly secure communication are

$$H(S|Y) = 0 \tag{8.19}$$

$$I(S_i; W) = 0, \quad \forall i, \forall \mathcal{I} \colon |\mathcal{I}| = \mu. \tag{8.20}$$

It was shown in [24] that weakly secure communication at the maximum rate of $k = n$ packets can be achieved if $\mu < n$ and q is sufficiently large.

One main idea behind weak security is that, from the point of view of a specific message packet S_i, the remaining message packets function as random data that confuse the wiretapper (similarly to the role of X_R in Section 8.5). Thus, for the security to be most effective, we should require $S_1, \ldots, S_k$ to be independent and uniformly distributed (although this is not strictly necessary in all cases).

The work of [24] also investigated a notion of security against guessing. The motivation is that, under weak security, the wiretapper may, in some cases, discover many of the message packets by simply guessing (correctly) a single message packet. Ideally, we would want the wiretapper to obtain at most 1 unit of meaningful information per guess. In the following, we assume that the wiretapper is able to guess (or obtain through some side channel) at most g of the message packets. Denote the set of indices of the guessed packets by $\mathcal{G} \subset \{1, \ldots, k\}$ and let $S_{\mathcal{G}} = (S_i,\, i \in \mathcal{G})$. Then, for weak security against g

guesses, condition (8.20) is replaced by

$$I(S_i; W|S_{\mathcal{G}}) = 0, \quad \forall i, \forall \mathcal{G}\colon |\mathcal{G}| \leq g, \forall \mathcal{I}\colon |\mathcal{I}| = \mu. \tag{8.21}$$

It was shown in [24] that the maximum rate of $k = n$ packets can also be achieved under weak security against g guesses, provided $\mu + g < n$ and q is sufficiently large.

The results of [24] on weak security are possible if the network code is known and the outer code is designed accordingly. In this section, we wish to address the problem of *universal* weak security against g guesses. More precisely, we replace condition (8.21) with

$$I(S_i; W|S_{\mathcal{G}}) = 0, \quad \forall i, \forall \mathcal{G}\colon |\mathcal{G}| \leq g, \forall B. \tag{8.22}$$

We start by proving some general results on (not necessarily universal) weak security against guessing.

Proposition 8.7: Let $\mathcal{G}, \mathcal{G}' \subseteq \{1, \ldots, n\}$. Then

$$I(S_i; W|S_{\mathcal{G}}) = 0, \quad \forall i, \forall \mathcal{G}\colon |\mathcal{G}| \leq g \tag{8.23}$$

if and only if

$$I(S_{\mathcal{G}'}; W) = 0, \quad \forall \mathcal{G}'\colon |\mathcal{G}'| \leq g + 1. \tag{8.24}$$

Proof: Assume (8.23) is true. Choose any $\mathcal{G}'$ with $|\mathcal{G}'| \leq g + 1$, and let $\mathcal{G} = \mathcal{G}' \setminus \{i\}$ for some $i \in \mathcal{G}'$. Then

$$\begin{aligned} I(S_{\mathcal{G}'}; W) &= I(S_i, S_{\mathcal{G}}; W) \\ &= I(S_{\mathcal{G}}; W) + I(S_i; W|S_{\mathcal{G}}) \\ &= I(S_{\mathcal{G}}; W). \end{aligned}$$

Since the statement is clearly true for $g = 0$, by induction we obtain the desired result.

Conversely, assume (8.24) is true. Choose any i and any $\mathcal{G}$ such that $|\mathcal{G}| \leq g$. Let $\mathcal{G}' = \mathcal{G} \cup \{i\}$. Then

$$I(S_i; W|S_{\mathcal{G}}) = I(S_{\mathcal{G}'}; W) - I(S_{\mathcal{G}}; W) \leq I(S_{\mathcal{G}'}; W) = 0. \qquad \blacksquare$$

Proposition 8.7 shows that weak security against g guesses is equivalent to strong security of a "message" $S' = S_{\mathcal{G}'}$ for any $\mathcal{G}'$ with $|\mathcal{G}'| \leq g+1$. Thus, we can use the analysis of the previous section to study this problem. The following converse result is immediate.

Corollary 8.8: Consider an $(n, m, \mathbb{F}_q)$ linear multicast network subject to μ observations. Communication at the maximum rate $k = n$ with weak security against g guesses is possible only if $g + 1 + \mu \leq n$. For universal weak security against g guesses, it is necessary, in addition, that $m \geq n$.

Before proceeding to our achievability results, we mention that the notion of weak security proposed in [24] is in fact slightly stronger than the one presented above; namely, the guesses are allowed to be any $\mathbb{F}_q$-linear combinations of the message packets. We can strengthen this condition even further and require that every $\mathbb{F}_q$-linear combination of the message packets is also secure against guesses. More precisely, we can replace (8.22) with

$$I(P_0 S; W | P_1 S, \ldots, P_g S) = 0, \quad \forall P_0, \ldots, P_g \in \mathbb{F}_q^{1\times k}, \forall B.$$

Following the same steps as in Proposition 8.7 yields the equivalent condition

$$I(PS; W) = 0, \quad \forall P \in \mathbb{F}_q^{(g+1)\times k}, \forall B \tag{8.27}$$

which can be interpreted as universal strong security of $S' = PS$ for any $P \in \mathbb{F}_q^{(g+1)\times k}$. Unless otherwise mentioned, we will use this strengthened version of weak security in the remaining results of this section.

In the next theorem, we assume that S is uniformly distributed on F^k. We also assume that $k \geq n - \mu$, since only in this case is weak security appealing over strong security.

Theorem 8.9: Consider an $(n, m, \mathbb{F}_q)$ linear multicast network subject to μ observations. A coset coding scheme based on a linear $(n, n-k)$ code over $\mathbb{F}_{q^m}$ with parity-check

matrix $H \in \mathbb{F}_{q^m}^{k\times n}$ provides universal weak security against the maximum number of $g = n - \mu - 1$ guesses if and only if the code defined by the parity-check matrix PH is an MRD code with $m \geq n$ for every full-rank $P \in \mathbb{F}_q^{(g+1)\times k}$. Equivalently, H must be such that the matrix PHT is nonsingular for all full-rank $P \in \mathbb{F}_q^{(g+1)\times k}$ and all full-rank $T \in \mathbb{F}_q^{n\times(g+1)}$.

Proof: Let $S' = PS$. Since $X \in F^n$ is chosen uniformly at random such that $S = HX$, it is easy to see that the distribution of X conditioned on S' is uniform on the values such that $S' = PHX$. Using Proposition 8.1, we obtain that (8.27) is satisfied if and only if

$$\mathsf{rank} \begin{bmatrix} PH \\ B \end{bmatrix} = g + 1 + \mu = n$$

for all full-rank $P \in \mathbb{F}_q^{(g+1)\times k}$ and all full-rank $B \in \mathbb{F}_q^{\mu\times n}$. The result now follows by applying Theorem 8.2. The equivalent condition is an immediate consequence of Theorem 2.1. ■

Theorem 8.9 establishes the conditions for a coset coding scheme to provide universal weak security that is maximally secure against guessing. Note that we may choose $k < n$ if we are interested in a tradeoff between weak and strong security. That is, up to $n - k$ packets may be eavesdropped without leaking any information, and up to $n - g - 1$ packets may be eavesdropped without leaking any meaningful information.

The next result guarantees the existence of a universal weakly secure communication scheme if the packet length m is sufficiently large.

Theorem 8.10: There exists some matrix $H \in \mathbb{F}_{q^m}^{k\times n}$ satisfying the conditions of Theorem 8.9 for all $0 \leq g \leq k - 1$ if $m \geq (n + k)^2/8 + \log_q 16k$.

Proof: The proof follows the same argument as in [4]. Consider the kn entries of H as indeterminates $\xi_1, \ldots, \xi_{kn}$, forming the set $\boldsymbol{\xi}$. For convenience, let $r = g + 1$. For

$P \in \mathbb{F}_q^{r\times k}$ and $T \in \mathbb{F}_q^{n\times r}$, let

$$f_{P,T}(\boldsymbol{\xi}) = \det(PHT)$$

and let

$$f(\boldsymbol{\xi}) = \prod_P \prod_T f_{P,T}(\boldsymbol{\xi}) \tag{8.30}$$

where the first (second) product is over all full-rank matrices in reduced row (column) echelon form. Then, there exists some assignment of $\xi_1, \ldots, \xi_{kn} \in \mathbb{F}_{q^m}$ such that H satisfies the conditions of Theorem 8.9 if and only if the polynomial $f(\boldsymbol{\xi})$ is not identically zero as an element of $\mathbb{F}_{q^m}[\boldsymbol{\xi}]$. Note that we only need to consider matrices P (T) in reduced row (column) echelon form, since performing any elementary row (column) operations does not change the rank of PHT.

It is known that a multivariate polynomial over a finite field cannot be identically zero if the maximum degree of a variable is smaller than the cardinality of the field. Thus, a solution must exist if m is sufficiently large. To obtain an upper bound on the required m, we simply need to compute δ, the maximum degree of a variable in $f(\boldsymbol{\xi})$. Since

$$(PHT)_{i,j} = \sum_{\ell=1}^{k}\sum_{t=1}^{k} P_{i,\ell}H_{\ell,t}T_{t,j}$$

we have that the maximum degree of a variable in $f_{P,T}(\boldsymbol{\xi})$ is at most r. There are precisely $\begin{bmatrix}k\\r\end{bmatrix}_q\begin{bmatrix}n\\r\end{bmatrix}_q$ factors in (8.30), where $\begin{bmatrix}n\\r\end{bmatrix}_q$ denotes the number of r-dimensional subspaces of $\mathbb{F}_q^n$. Thus, $\delta \le r\begin{bmatrix}k\\r\end{bmatrix}_q\begin{bmatrix}n\\r\end{bmatrix}_q$. Using the upper bound (2.12), we have that

$$\begin{aligned}\delta &\le r\begin{bmatrix}k\\r\end{bmatrix}_q\begin{bmatrix}n\\r\end{bmatrix}_q\\ &\le r4q^{r(k-r)}4q^{r(n-r)} = 16rq^{r(n+k-2r)}\\ &\le 16kq^{2r(n+k-2r)/2}\\ &\le 16kq^{(n+k)^2/8}.\end{aligned}$$

Thus, a solution always exists if $m \ge (n+k)^2/8 + \log_q 16k$. ∎

While Theorem 8.10 provides an existence result for m on the order of n^2, we remark that this bound is not tight, i.e., smaller values of m are sufficient in many special cases.

At this point we return to the first notion of weak security against guessing, condition (8.21). Due to the weaker nature of this condition, some results are easier to prove. Note that Theorem 8.9 still holds under condition (8.21) if we constrain P to be a submatrix of a permutation matrix. The proof of Theorem 8.10 also follows unchanged except for $\begin{bmatrix} k \\ r \end{bmatrix}_q$ being replaced by $\binom{k}{r}$; thus, an existence result can be obtained for $m \geq n^2/4 + \log_q n \binom{k}{\lfloor k/2 \rfloor}$. For the special case of 0 or 1 guess, a suitable matrix H can be found explicitly, provided m is equal to n or slightly larger. The latter result is shown in the next proposition.

Proposition 8.11: Let $m \geq n$ and let $h_1, \ldots, h_n \in \mathbb{F}_{q^m}$ be elements that are linearly independent over $\mathbb{F}_q$. Let $H = [H_{ij}] \in \mathbb{F}_{q^m}^{k \times n}$ be given by $H_{i,j} = h_i^{q^{j-1}}$, for $1 \leq i \leq k$ and $1 \leq j \leq n$. Then every row of H defines an MRD code. Moreover, if m is prime, then every pair of rows of H define an MRD code.

Proof: The first statement follows immediately from the fact that $h_1^{q^i}, \ldots, h_n^{q^i}$ are linearly independent over $\mathbb{F}_q$, for all i (otherwise $h_1, \ldots, h_n$ would not be so).

For the second statement, we must show that any two rows of HT are linearly independent over $\mathbb{F}_q$, for any full-rank matrix $T \in \mathbb{F}_q^{n \times 2}$. Any two such rows, say the $(i+1)$th and the $(j+1)$th rows, can be written as $\begin{bmatrix} \alpha_1^{q^i} & \alpha_2^{q^i} \end{bmatrix}$ and $\begin{bmatrix} \alpha_1^{q^j} & \alpha_2^{q^j} \end{bmatrix}$. Note that $\alpha_1, \alpha_2 \in \mathbb{F}_{q^m}$ must be linearly independent over $\mathbb{F}_q$, otherwise T would not be full-rank.

These two rows are linearly dependent only if $(\alpha_2/\alpha_1)^{q^i} = (\alpha_2/\alpha_1)^{q^j}$, i.e., $(\alpha_2/\alpha_1)^{q^{i-j}} = (\alpha_2/\alpha_1)$. This implies that α_2/α_1 is a root of $x^{q^{i-j}} - x$ and therefore lies in a proper subfield of $\mathbb{F}_{q^m}$. Since m is prime, there are no subfields between $\mathbb{F}_q$ and $\mathbb{F}_{q^m}$. Thus, α_2/α_1 must lie in $\mathbb{F}_q$, which means that α_2 and α_1 are linearly dependent over $\mathbb{F}_q$. This is a contradiction. Hence, the two rows must be linearly independent. ∎

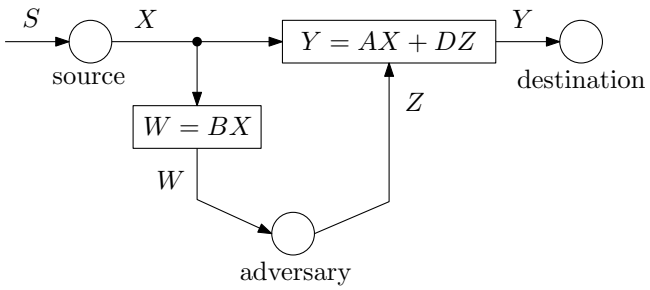

Figure 8.2: Linear network coding channel with a wiretapper/jammer adversary.

8.4 Extension: A Wiretapper-Jammer Adversary

In this section, we extend the model of the previous sections to the case where the wiretapper is also allowed to inject erroneous packets into the network in order to disrupt communication.

Consider the system depicted in Fig. 8.2. As before, F is a vector space of dimension m over $\mathbb{F}_q$. The source node encodes a message $S \in F^k$ into a tuple $X \in F^n$ for transmission over the network. The adversary observes $W = BX$, where $B \in \mathbb{F}_q^{\mu \times n}$, and injects into the network a tuple $Z \in F^t$ corresponding to the error packets $Z_1, \ldots, Z_t \in F$. In this case, we will say that the network *is subject to t errors and μ observations*. The tuple $Y \in F^N$ received by the destination is given by

$$Y = AX + DZ$$

where $A \in \mathbb{F}_q^{N \times n}$ is the transfer matrix from the source to the destination, and $D \in \mathbb{F}_q^{n \times t}$ is the transfer matrix from the adversary to the destination (see Chapter 3).

The goal of the system designer is, again, to ensure secure communication. Note that the communication requirement is now made more difficult due to the presence of error packets.

In the remainder of this section, we will move directly to our main goal, which is to achieve *universal* secure communication; namely, we want to find an encoding from S to X that can provide secure communication for any choice of a feasible linear network

code.

More formally, we make the following assumptions: A is an arbitrary matrix with full column-rank known to both the adversary and the destination; B and D are arbitrary matrices known only to the adversary; and S, X, W, Z and Y are random variables with the Markov chain structure shown in Fig. 8.2, i.e., the joint probability mass function factors as

$$p(s, x, w, z, y) = p(s)p(x|s)p(w|x)p(z|w)p(y|x, z).$$

The conditions for universal secure communication are then given by

$$H(S|Y) = 0, \quad \forall A, D, p(z|w) \tag{8.34}$$

$$I(S; W) = 0, \quad \forall B. \tag{8.35}$$

Our objective is to design $p(x|s)$ such that (8.34) and (8.35) are satisfied.

We will now show that this goal can be achieved if $k \leq n - 2t - \mu$ and m is sufficiently large. Our approach will be to first "clean" the channel using an (inner) error-correcting scheme, and then use an (outer) coset coding scheme to provide security. We start with the following lemma.

Lemma 8.12: Assume that F is a vector space over $\mathbb{F}_{q^n}$ and that $\ell = n - 2t > 0$. Let $G \in \mathbb{F}_{q^n}^{\ell \times n}$ be a generator matrix of a linear (n, ℓ) MRD code over $\mathbb{F}_{q^n}/\mathbb{F}_q$. Suppose X is given by $X = G^T U$, for some random variable $U \in F^\ell$. Then $H(U|Y) = 0$.

Proof: From Theorem 2.2, the set $\mathcal{C} = \{G^T U, U \in F^\ell\}$ is a rank-metric code over $F/\mathbb{F}_q$ with $d_{\mathrm{R}}(\mathcal{C}) = n - \ell + 1 = 2t + 1$. Let $A^* \in \mathbb{F}_q^{n \times N}$ be such that $A^* A = I$. Let

$$\bar{Y} = A^* Y = X + A^* D Z = X + \bar{Z}$$

where $\bar{Z} = A^* D Z$. Note that, regardless of A, D and $p(z|w)$, we have $\mathsf{rank}\, \bar{Z} \leq t$. Now, since $\mathcal{C}$ can correct any t rank errors (see Sections 2.3 and 4.2), it follows that $H(X|Y) = 0$ and therefore $H(U|Y) = 0$. ∎

Remark: The results of Section 4.3 show that Lemma 8.12 also holds if the matrix A is unknown (as in random network coding), provided that an $n \times n$ identity matrix is prepended to each transmitted matrix X. □

Lemma 8.12 shows that, using an appropriate error-correcting scheme, we can guarantee universal communication of an auxiliary variable $U \in F^\ell$ from the source to the destination. Thus, we can regard the resulting system as a feasible linear network subject only to wiretapping. The wiretapper observation is given by

$$W = BX = BG^T U = \tilde{B}U$$

where $\tilde{B} = BG^T \in \mathbb{F}_{q^n}^{\mu \times \ell}$.

Now, let $Q = q^n$ and let r be the dimension of F as a vector space over $\mathbb{F}_Q$. Since $\tilde{B}$ has entries in $\mathbb{F}_Q$, the resulting network with input variable U can be regarded as a feasible $(\ell, r, \mathbb{F}_Q)$ linear network subject to μ observations. If $\mu < \ell$ and $r \geq \ell$, then we can use Theorem 8.3 to achieve universal secure communication at a rate $k = \ell - \mu$. We thus have proved the following theorem.

Theorem 8.13: Consider an $(n, m, \mathbb{F}_q)$ linear multicast network subject to t errors and μ observations. Assume $\mu + 2t < n$, $m = nr$ and $r \geq \ell$, where $\ell = n - 2t$. Let $G_{\text{in}} \in \mathbb{F}_{q^n}^{\ell \times n}$ be a generator matrix of a linear (n, ℓ) MRD code over $\mathbb{F}_{q^n}/\mathbb{F}_q$ and let $H_{\text{out}} \in \mathbb{F}_{q^m}^{(\ell-\mu)\times \ell}$ be a parity-check matrix of a linear (ℓ, μ) MRD code over $\mathbb{F}_{q^m}/\mathbb{F}_{q^n}$. Then, universal secure communication at rate $k = n - 2t - \mu$ can be achieved by using a coset coding scheme based on H_{out} concatenated with an error-correcting scheme based on G_{in}.

The system in Theorem 8.13 is depicted in Fig. 8.3. For completeness, we also show the complementary result where the error-correcting scheme is used outside the coset coding scheme.

Theorem 8.14: Consider an $(n, m, \mathbb{F}_q)$ linear multicast network subject to t errors and μ observations. Assume $\mu + 2t < n$, $m = nr$ and $r \geq \ell$, where $\ell = n - \mu$. Let $H_{\text{in}} \in \mathbb{F}_{q^n}^{\ell \times n}$

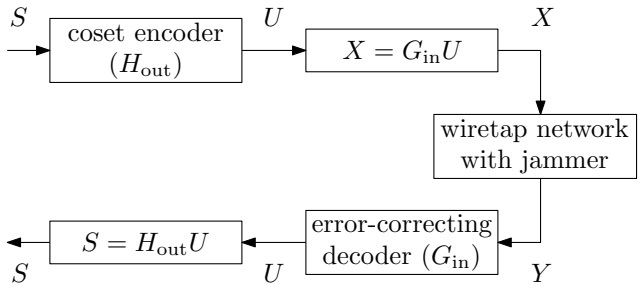

Figure 8.3: Concatenated coding scheme for a linear network with a wiretapper/jammer adversary.

be a parity-check matrix of a linear (n, μ) MRD code over $\mathbb{F}_{q^n}/\mathbb{F}_q$ and let $G_{\text{out}} \in \mathbb{F}_{q^m}^{(\ell-2t)\times\ell}$ be a generator matrix of a linear $(\ell, \ell-2t)$ MRD code over $\mathbb{F}_{q^m}/\mathbb{F}_{q^n}$. Then, universal secure communication at rate $k = n-2t-\mu$ can be achieved by using an error-correcting scheme based on G_{out} concatenated with a coset coding scheme based on H_{in}.

Proof: The security of the scheme is clear. We just have to prove that the message S can be obtained at the destination node.

Let $U \in F^\ell$ be an auxiliary variable such that $U = G^T S$ and $U = H_{\text{in}}X$. Let

$$\tilde{Y} = H_{\text{in}}A^{-1}Y = H_{\text{in}}X + H_{\text{in}}A^{-1}DZ = U + \tilde{D}Z$$

where $\tilde{D} = H_{\text{in}}A^{-1}D \in \mathbb{F}_{q^n}^{n\times t}$. Since U is a codeword of a rank-metric code over $\mathbb{F}_{q^m}/\mathbb{F}_{q^n}$ with minimum rank distance $2t+1$, it follows that U can be perfectly recovered from $\tilde{Y}$. Thus, $H(S|Y) = 0$. ■

It follows from Theorems 8.13 and 8.14 that universal secure communication can be achieved with a packet of size $m = n\ell$, where $\ell = \min\{n-2t, n-\mu\}$.

Remark: The results of this section show that, for sharing a secret between source and destination, it is sufficient that the network satisfies the condition $\mu + 2t < n$. In [16], the same condition was obtained using different methods that require both q and m to grow to infinity. Later, an improved condition of $\mu + t < n$ was obtained in [68] using

the same approach, i.e., letting $q, m \to \infty$. Note that, due to the necessity of choosing a large q, these schemes may not be suitable for practical applications. It remains an open question whether the improved condition of [68] can be obtained for a universal scheme with moderate packet length and field size. □

We conclude this section by mentioning that, following the same tandem approach, it is straightforward to generalize the results of Section 8.3 to the case of a wiretapper-jammer adversary.

8.5 Practical Considerations

As described in Sections 8.2 and 8.3 encoding and decoding of the source message (for both strong and weak security) require operations to be performed in the extension field $\mathbb{F}_{q^m}$. While decoding corresponds to a matrix multiplication $S = HX$, encoding (at least for strong security) corresponds to choosing a random word within a specified coset. It is convenient in practice to have an encoder that performs similarly to the decoder, i.e., that also amounts to matrix multiplication. Given a matrix $H \in \mathbb{F}_{q^m}^{k\times n}$ (where $k \leq n$), let $\tilde{H} \in \mathbb{F}_{q^m}^{(n-k)\times n}$ be any matrix such that $T = \begin{bmatrix} H \\ \tilde{H} \end{bmatrix}$ is nonsingular. Then, encoding of a message $S \in \mathbb{F}_{q^m}^{k}$ may be performed by first selecting a "noise vector" $R \in \mathbb{F}_{q^m}^{(n-k)}$ uniformly at random and independently from S, and then computing $X = T^{-1}\begin{bmatrix} S \\ R \end{bmatrix}$. Thus, both encoding and decoding can be performed with $O(n^2)$ operations in $\mathbb{F}_{q^m}$.

If $m = rn$, with $r > 1$, then the encoding and decoding complexity can be further reduced by using the construction in Theorem 2.2. Namely, take $F = \mathbb{F}_{q^n}^{1\times r}$, which makes $S \in \mathbb{F}_{q^n}^{k\times r}$ and $X \in \mathbb{F}_{q^n}^{n\times r}$, and turns $S = HX$ into a matrix-by-matrix multiplication. In this case, using the same approach as above, encoding and decoding can be performed in $O(rn^2)$ operations in $\mathbb{F}_{q^n}$, which may be a substantial reduction depending on how

extension field arithmetic is implemented.

Now, if extension field arithmetic is performed using an optimal or low-complexity normal basis, then the complexity can be even more dramatically reduced—at least for strong security and weak security under the construction of Proposition 8.11. Let $T = [T_{ij}] \in \mathbb{F}_{q^n}^{n \times n}$ be given by $T_{i,j} = \alpha^{[i-1+j-1]}$, for $1 \leq i, j \leq n$, where α is a normal element in $\mathbb{F}_{q^n}$. Then T is invertible, and the matrix H induced by taking the first $k \leq n$ rows of T satisfies the conditions for strong security and weak security (with 0 guesses, or 1 guess if n is prime). Taking α to be generator of the normal basis for extension field arithmetic (as in the approach of Section 6.3), allows the decoding to be performed with $O(rn^3)$ operations in the base field $\mathbb{F}_q$ (rather than $O(rn^4)$ as before). Moreover, if q has characteristic 2, then the decoding complexity is given by $n^2(C(\alpha) - 1)$ *additions* in $\mathbb{F}_q$, where $C(\alpha)$ is the complexity of the normal basis generated by α. Although the encoding complexity is still comparatively high, the situation changes if the normal basis is self-dual. In this case, we have precisely $T^{-1} = T$, that is, encoding and decoding are essentially identical.

Thus, if $m = rn$, q is a power of 2, and an optimal self-dual normal basis over $\mathbb{F}_q$ is used for extension field arithmetic, then encoding and decoding can be performed with $2rn^3 = 2n^2m$ additions in $\mathbb{F}_q$—that is, only XORs. Note that this complexity is even (much) smaller than performing Gaussian elimination on the received matrix.

Finally, it is worth mentioning that our security scheme can be seamlessly integrated with random network coding. We simply require that each packet transports a header of length n containing the global coding vector associated with the packet; thus, the total packet length must be $n + m \geq 2n$ symbols in $\mathbb{F}_q$. Note that the linear network code not being feasible affects communication but has no impact on the security of the scheme.

Chapter 9

Conclusion

This thesis has investigated an end-to-end approach to forward error control and information-theoretic security in network coding. By placing all the error control and security schemes at the endpoints, we allow a perfectly layered design of a network-coding-based communication system. The designer of the inner network coding system need not care about error propagation or security issues, while the designer of the outer security and error control schemes needs to know only basic parameters of the inner system such as packet length and field size. We believe that this layered approach, which has many similarities with the design of the Internet and even of classical communication systems [69], can potentially help network coding become commonplace in the design of communication networks.

The contributions of this thesis are summarized as follows. We propose a worst-case model for a network coding system (either coherent or noncoherent) under the threat of an adversary (a jammer) that can introduce a bounded number of error packets. In both cases we propose a distance function that elegantly and succinctly describes the correction capability of a code: the rank metric, for coherent network coding, and the injection metric, for noncoherent network coding. A consequence of our results is that optimal (for the coherent case) and near-optimal (for the noncoherent case) codes can be

easily constructed based on rank-metric codes. Remarkably, in both cases, the decoding problem can be treated as a single unified decoding problem, which is shown to be a generalization of the conventional rank-metric decoding problem. This result enables us to use the existing techniques for rank-metric codes (with some adaptations) to decode our end-to-end codes for network coding.

We also delve into the domain of abstract algebra (the topics of extension field arithmetic and linearized polynomials) to propose two new encoding and two new decoding algorithms for Gabidulin codes that are faster in their respective competencies than any previously proposed algorithms. One decoding algorithm makes use of normal basis theory and is useful for high-rate codes; the other decoding algorithm adds a transform-domain twist and is useful for low-rate codes; and the two encoding algorithms are useful for high-rate systematic codes and nonsystematic codes, respectively.

In addition, we investigate a more information-theoretic scenario where the network coding system is subject not to the threat of a jammer but to the occurrence of random packet errors. We compute upper and lower bounds on the channel capacity and propose a simple, reliable and computationally efficient coding scheme that can achieve capacity asymptotically in the packet length or field size. The scheme is based on a combination of two strategies that we call channel sounding and error trapping. For finite parameters, the scheme has lower probability of failure than any other previously proposed schemes and, surprisingly, decoding amounts simply to the standard Gaussian elimination that is usually already embedded in a typical network coding system.

Finally, we propose the first universal coding schemes for network coding that can provide strong and/or weak information-theoretic security against a wiretapper adversary. Once again, our schemes rely on rank-metric codes and can be implemented very efficiently (faster than Gaussian elimination) if certain normal bases are available for extension field arithmetic. We also address the problem of providing both security and error control when the adversary is not only a wiretapper but also a jammer. Our contribution

here is the final test of our layered approach: we show that, under an appropriate interface, our proposed error control and security schemes can be simply concatenated—even the order does not matter.

9.1 Open Problems

At this point, several potential questions may arise in the reader's mind. What if the network is subject not to packet errors but only erasures? At least for generation-based (one-shot) codes, this question falls into the framework of the LNCC (Chapter 3), since packet erasures are captured by a rank deficiency of the transfer matrix. A nontrivial problem to study is when the multiple destinations experience different erasure levels. A solution is provided by priority encoding transmission [70], which is adapted to network coding in [71]. As one might expect, we simply need to replace conventional MDS codes with MRD codes. It is an open question, however, how to provide reliability under a multi-shot scenario, i.e., when the rank of the transfer matrix to the same receiver is a random variable (and therefore coding must be done over multiple generations).

Even in the case that the transfer matrix is assumed to be always nonsingular, it might be worth investigating the error correction problem under the perspective of multi-shot codes. Although both models of Chapters 4 and 7 can be easily extended to cover multiple channel uses, designing good codes for these models (i.e., better than the trivial one-shot extensions) still remain an intriguing possibility. In that regard, the recent work of [72] seems to be a promising approach.

Chapter 4 deals with error control against an omniscient adversary, and concludes that the maximum achievable rate is $n-2t$ packets per transmission. One might wonder what is the ultimate rate achievable under a non-omniscient adversary. The question is elusive, but the answer can generally be regarded as $n-t$. This is shown in [16, 73] using a non-universal scheme and recently in [67] using a universal scheme; the trick, in both

schemes, is that source and destination share a secret key. A question is then how to share this key using the network itself. Assuming that the adversary can eavesdrop at most μ packets, the work of [16] proposes a non-universal scheme to share a secret provided $n-2t-\mu>0$, while Section 8.4 proposes a universal scheme that works under the same conditions. The work of [68], however, improves this condition to $n-t-\mu>0$ using a non-universal scheme. As mentioned in Section 8.4, it is an open question whether the same condition can be obtained using a universal scheme.

Roughly speaking, Chapter 4 considers a threat model where the adversary can arbitrarily choose both the row and column spaces of the error matrix (each of bounded dimension t), and provides a scheme achieving rate $n-2t$. Chapter 7, one the other hand, assumes that both row and column error spaces are selected uniformly at random, and provides a scheme achieving rate $n-t$. It is implicit in [67] that if the adversary can choose only the column space of the error matrix (but not the row space, which is assumed to be random), than error trapping can be combined with MRD coding to provide a scheme achieving rate $n-t$. The question is: what can be said when the adversary can choose only the row space of the error matrix, while its column space is random? Can any scheme be devised achieving a higher rate than $n-2t$? The question is hard because it involves a mixed adversarial-probabilistic characterization of the channel. On the other hand, this situation is perhaps the most realistic model for random network coding: the adversary can arbitrarily choose what packets to inject, but cannot influence the actual network code.

The model of Chapter 7 considers error packets essentially drawn uniformly at random from $\mathbb{F}_q^m$. There are certain situations (e.g., [9]) where one should expect that error packets would rather have low weight; say, a symbol would get corrupted with a very small probability. While the results in [67] show that the approach of Chapter 7 is still valid in this case (if we first apply an inner "randomization" scheme), a capacity analysis would show that even better rates should be achievable. It is an open problem to design

coding schemes that are matched to these specific types of error patterns.

Another open problem, which may be specially appealing to mathematicians, is the design of optimal subspace codes for the injection metric. Several recent results have proposed upper and lower bounds on the size of optimal codes, as well as constructions of codes that are superior to liftings of MRD codes. Although most of these results consider either constant-dimension codes or general subspace codes under the subspace metric [52, 74–78], some of the most recent ones have started considering bounds and constructions for the injection metric [77, 79].

Another problem that might be interesting to mathematicians concerns the explicit construction of matrices that allow weak security against the maximum number of guesses. The proof of Theorem 8.10 not only is an existential argument (which requires exponential design complexity [4]) but also is loose in many cases (as Proposition 8.11 evidences). New algebraic tools on extensions of finite fields and corresponding linear maps could enable the efficient design of matrices for weak security requiring much smaller packet sizes.

Another potential research area that we might suggest concerns the efficient implementation of Gabidulin encoders and decoders—either in hardware or in GPU-based software. As discussed in Chapter 6, the complexity bottleneck for the standard (time-domain) Gabidulin decoder is now the Berlekamp-Massey algorithm and Gabidulin's algorithm. For high-rate codes, these could perhaps be efficiently implemented by "hard-coding" the equations involved. Also, as mentioned in Section 6.6, for Cartesian-type Gabidulin codes, the received words in the same received matrix may share the same error location patterns; exploiting this situation could significantly speed up the decoding. In any case, we believe that parallelization, which has become commonplace in today's hardware and software implementations, can provide dramatic gains that are not revealed by time-complexity expressions such as those presented in Chapter 6.

Appendix A

Detection Capability

When dealing with communication over an adversarial channel, there is little justification to consider the possibility of error detection. In principle, a code should be designed to be unambiguous (in which case error detection is not needed); otherwise, if there is any possibility for ambiguity at the receiver, then the adversary will certainly exploit this possibility, leading to a high probability of decoding failure (detected error). Still, if a system is such that (a) sequential transmissions are made over the same channel, (b) there exists a feedback link from the receiver to the transmitter, and (c) the adversary is not able to fully exploit the channel, then it might be worth using a code with a lower correction capability (but higher rate) that has some ability to detect errors.

Following classical coding theory, we consider error detection in the presence of a bounded error-correcting decoder. More precisely, define a *bounded-discrepancy decoder with correction radius* t, or simply a *t-discrepancy-correcting decoder*, by

$$\hat{x}(y) = \begin{cases} x & \text{if } \Delta(x,y) \leq t \text{ and } \Delta(x',y) > t \text{ for all } x' \neq x,\ x' \in \mathcal{C} \\ f & \text{otherwise.} \end{cases}$$

Of course, when using a t-discrepancy-correcting decoder we implicitly assume that the code is t-discrepancy-correcting. The discrepancy correction capability of a code (under a t-discrepancy-correcting decoder) is the maximum value of discrepancy for which the

decoder above is guaranteed to return f.

For $t \in \mathbb{N}$, let the function $\sigma^t \colon \mathcal{X} \times \mathcal{X} \to \mathbb{N}$ be given by

$$\sigma^t(x, x') \triangleq \min_{y \in \mathcal{Y} \colon \Delta(x', y)} \Delta(x, y) - 1. \tag{A.2}$$

Proposition A.1: The discrepancy-detection capability of a code $\mathcal{C}$ is given exactly by $\sigma^t(\mathcal{C})$. That is, under a t-discrepancy-correction decoder, any discrepancy of magnitude s can be detected if and only if $s \leq \sigma^t(\mathcal{C})$.

Proof: Let $t < s \leq \sigma^t(\mathcal{C})$. Suppose that $x \in \mathcal{X}$ is transmitted and $y \in \mathcal{Y}$ is received, where $t < \Delta(x, y) \leq s$. We will show that $\Delta(x', y) > t$, for all $x' \in \mathcal{C}$. Suppose, by way of contradiction, that $\Delta(x', y) \leq t$, for some $x' \in \mathcal{C}$, $x' \neq x$. Then $\sigma^t(\mathcal{C}) \leq \Delta(x, y) - 1 \leq s - 1 < s \leq \sigma^t(\mathcal{C})$, which is a contradiction.

Conversely, assume that $\sigma^t(\mathcal{C}) < s$, i.e., $\sigma^t(\mathcal{C}) \leq s - 1$. We will show that an undetected error may occur. Since $\sigma^t(\mathcal{C}) \leq s - 1$, there exist $x, x' \in \mathcal{C}$ such that $\sigma^t(x, x') \leq s - 1$. This implies that there exists some $y \in \mathcal{Y}$ such that $\Delta(x', y) \leq t$ and $\Delta(x, y) - 1 \leq s - 1$. By assumption, $\mathcal{C}$ is t-discrepancy-correcting, so $\hat{x}(y) = x'$. Thus, if x is transmitted and y is received, an undetected error will occur, even though $\Delta(x, y) \leq s$. ∎

The result above has also been obtained in [50], although with a different notation (in particular, treating $\sigma^0(x, x') + 1$ as a "distance" function). Below, we characterize the detection capability of a code in terms of the Δ-distance.

Proposition A.2: For any code $\mathcal{C}$, we have $\sigma^t(\mathcal{C}) \geq \delta(\mathcal{C}) - t - 1$.

Proof: For any $x, x' \in \mathcal{X}$, let $y \in \mathcal{Y}$ be a solution to the minimization in (A.2), i.e., y is such that $\Delta(x', y) \leq t$ and $\Delta(x, y) = 1 + \sigma^t(x, x')$. Then $\delta(x, x') \leq \Delta(x, y) + \Delta(x', y) \leq 1 + \sigma^t(x, x') + t$, which implies that $\sigma^t(x, x') \leq \delta(x, x') - t - 1$. ∎

Theorem A.3: Suppose that $\Delta(\cdot, \cdot)$ is normal. For every code $\mathcal{C} \subseteq \mathcal{X}$, we have $\sigma_t(\mathcal{C}) = \delta(\mathcal{C}) - t - 1$.

Proof: We just need to show that $\sigma^t(\mathcal{C}) \leq \delta(\mathcal{C}) - t - 1$. Take any $x, x' \in \mathcal{X}$. Since $\Delta(\cdot, \cdot)$ is normal, there exists some $y \in \mathcal{Y}$ such that $\Delta(x', y) = t$ and $\Delta(x, y) = \delta(x, x') - t$. Thus, $\sigma^t(x, x') \leq \Delta(x, y) - 1 = \delta(x, x') - t - 1$. ■

Appendix B

Omitted Proofs

B.1 Proofs for Chapter 4

Before proving Lemma 4.15, we need the following result [27].

Lemma B.1: Let $X, Y \in \mathbb{F}_q^{N \times M}$. Then

$$\mathsf{rank}\,(X - Y) \geq \max\{\mathsf{rank}\ X, \mathsf{rank}\ Y\} - \mathsf{dim}\,(\langle X \rangle \cap \langle Y \rangle).$$

Proof: We have

$$\mathsf{rank} \begin{bmatrix} X \\ Y \end{bmatrix} = \mathsf{rank} \begin{bmatrix} X \\ Y - X \end{bmatrix} \leq \mathsf{rank}\,(Y - X) + \mathsf{rank}\ X$$

$$\mathsf{rank} \begin{bmatrix} X \\ Y \end{bmatrix} = \mathsf{rank} \begin{bmatrix} Y - X \\ Y \end{bmatrix} \leq \mathsf{rank}\,(Y - X) + \mathsf{rank}\ Y.$$

Thus,

$$\mathsf{rank} \begin{bmatrix} X \\ Y \end{bmatrix} \leq \mathsf{rank}\,(Y - X) + \min\{\mathsf{rank}\ X, \mathsf{rank}\ Y\}.$$

The result now follows by applying (2.7). ∎

Proof (of Lemma 4.15): Using Lemma B.1 and (2.5), we have

$$\begin{aligned}\mathsf{rank}\,(AX - BY) &\geq \max\{\mathsf{rank}\,AX,\, \mathsf{rank}\,BY\} - \mathsf{dim}\,(\langle AX\rangle \cap \langle BY\rangle) \\ &\geq \max\{\mathsf{rank}\,X - \rho,\, \mathsf{rank}\,Y - \sigma\} - \mathsf{dim}\,(\langle X\rangle \cap \langle Y\rangle).\end{aligned}$$

We will now show that this lower bound is achievable. Our approach will be to construct A as $A = A_1A_2$, where $A_1 \in \mathbb{F}_q^{L\times(L+\rho)}$ and $A_2 \in \mathbb{F}_q^{(L+\rho)\times n}$ are both full-rank matrices. Then (2.5) guarantees that $\mathsf{rank}\,A \geq n - \rho$. The matrix B will be constructed similarly: $B = B_1B_2$, where $B_1 \in \mathbb{F}_q^{L\times(L+\sigma)}$ and $B_2 \in \mathbb{F}_q^{(L+\sigma)\times N}$ are both full-rank.

Let $k = \mathsf{rank}\,X$, $s = \mathsf{rank}\,Y$, and $w = \mathsf{dim}\,(\langle X\rangle \cap \langle Y\rangle)$. Let $W \in \mathbb{F}_q^{w\times m}$ be such that $\langle W\rangle = \langle X\rangle \cap \langle Y\rangle$, let $\tilde{X} \in \mathbb{F}_q^{(k-w)\times m}$ be such that $\langle W\rangle + \langle \tilde{X}\rangle = \langle X\rangle$ and let $\tilde{Y} \in \mathbb{F}_q^{(s-w)\times m}$ be such that $\langle W\rangle + \langle \tilde{Y}\rangle = \langle Y\rangle$. Then, let A_2 and B_2 be such that

$$A_2X = \begin{bmatrix} W \\ \tilde{X} \\ 0 \end{bmatrix} \quad \text{and} \quad B_2Y = \begin{bmatrix} W \\ \tilde{Y} \\ 0 \end{bmatrix}.$$

Now, choose any $\bar{A} \in \mathbb{F}_q^{i\times(k-w)}$ and $\bar{B} \in \mathbb{F}_q^{j\times(s-w)}$ that have full row rank, where $i = [k - w - \rho]^+$ and $j = [s - w - \sigma]^+$. For instance, we may pick $\bar{A} = \begin{bmatrix} I & 0 \end{bmatrix}$ and $\bar{B} = \begin{bmatrix} I & 0 \end{bmatrix}$. Finally, let

$$A_1 = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & \bar{A} & 0 & 0 \\ 0 & 0 & I & 0 \end{bmatrix} \quad \text{and} \quad B_1 = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & \bar{B} & 0 & 0 \\ 0 & 0 & I & 0 \end{bmatrix}$$

where, in both cases, the upper identity matrix is $w \times w$.

We have

$$\begin{aligned}
\mathsf{rank}\,(AX - BY) &= \mathsf{rank}\,(A_1A_2X - B_1B_2Y) \\
&= \mathsf{rank}\,(\begin{bmatrix} W \\ \bar{A}\tilde{X} \\ 0 \end{bmatrix} - \begin{bmatrix} W \\ \bar{B}\tilde{Y} \\ 0 \end{bmatrix}) \\
&= \max\{i,\, j\} \\
&= \max\{k - w - \rho,\, s - w - \sigma,\, 0\} \\
&= [\max\{k - \rho,\, s - \sigma\} - w]^+ .
\end{aligned}$$

■

B.2 Proofs for Chapter 5

Before proving Proposition 5.2, note that Lemma 4.15 (together with (2.9)) implies that, for all $X \in \mathbb{F}_q^{n\times m}$, $Y \in \mathbb{F}_q^{N\times m}$ and $Z \in \mathbb{F}_q^{n\times M}$, we have

$$\min_{A\in\mathbb{F}_q^{N\times n}} \mathsf{rank}\,(Y - AX) = \mathsf{rank}\,\begin{bmatrix} Y \\ X \end{bmatrix} - \mathsf{rank}\,X \tag{B.5}$$

$$\min_{B\in\mathbb{F}_q^{m\times M}} \mathsf{rank}\,(Z - XB) = \mathsf{rank}\,\begin{bmatrix} Z & X \end{bmatrix} - \mathsf{rank}\,X. \tag{B.6}$$

Proof (of Proposition 5.2): Let

$$\epsilon' = \min_{V^{(1)},L^{(2)}} \mathsf{rank}\,(e - \hat{L}V^{(1)} - L^{(2)}\hat{V}).$$

We first show the equivalence of 1) and 2). From (B.5), we have

$$\min_{L^{(2)}} \mathsf{rank}\,(e - \hat{L}V^{(1)} - L^{(2)}\hat{V}) = \mathsf{rank}\,\begin{bmatrix} e - \hat{L}V^{(1)} \\ \hat{V} \end{bmatrix} - \mathsf{rank}\,\hat{V}.$$

Similarly, from (B.6) we have

$$\begin{aligned}\min_{V^{(1)}} \mathsf{rank} \begin{bmatrix} e - \hat{L}V^{(1)} \\ \hat{V} \end{bmatrix} &= \min_{V^{(1)}} \mathsf{rank} \left(\begin{bmatrix} e \\ \hat{V} \end{bmatrix} - \begin{bmatrix} \hat{L} \\ 0 \end{bmatrix} V^{(1)} \right) \\ &= \mathsf{rank} \begin{bmatrix} \hat{L} & e \\ 0 & \hat{V} \end{bmatrix} - \mathsf{rank}\, \hat{L}.\end{aligned}$$

Thus,

$$\epsilon' = \mathsf{rank} \begin{bmatrix} \hat{L} & e \\ 0 & \hat{V} \end{bmatrix} - \mu - \delta$$

and the equivalence is shown.

Now, observe that the statement in 3) is equivalent to the statement that $\tau^* - \mu - \delta$ is the minimum value of ϵ for which there exist $V^{(1)} \in \mathbb{F}_q^{\mu \times m}$, $L^{(2)} \in \mathbb{F}_q^{n \times \delta}$, $L^{(3)} \in \mathbb{F}_q^{n \times \epsilon}$ and $V^{(3)} \in \mathbb{F}_q^{\epsilon \times m}$ satisfying

$$e = \hat{L}V^{(1)} + L^{(2)}\hat{V} + L^{(3)}V^{(3)}.$$

To show the equivalence of 2) and 3), we will show that $\epsilon' = \epsilon''$, where

$$\epsilon'' = \min_{\substack{\epsilon, V^{(1)}, L^{(2)}, L^{(3)}, V^{(3)}: \\ e = \hat{L}V^{(1)} + L^{(2)}\hat{V} + L^{(3)}V^{(3)}}} \epsilon.$$

We can rewrite ϵ'' as

$$\begin{aligned}\epsilon'' &= \min_{V^{(1)}, L^{(2)}} \min_{\substack{\epsilon, L^{(3)}, V^{(3)}: \\ e - \hat{L}V^{(1)} - L^{(2)}\hat{V} = L^{(3)}V^{(3)}}} \epsilon \\ &= \min_{V^{(1)}, L^{(2)}} \mathsf{rank}(e - \hat{L}V^{(1)} - L^{(2)}\hat{V}) \qquad \text{(B.11)} \\ &= \epsilon'.\end{aligned}$$

where (B.11) follows from (2.1). This shows the equivalence between 2) and 3). ∎

Proof (of Theorem 5.4): Let us write $X = \begin{bmatrix} I & x \end{bmatrix}$, $Y = \begin{bmatrix} \hat{A} & y \end{bmatrix}$, and $Z = \begin{bmatrix} W & z \end{bmatrix}$, where $x \in \mathbb{F}_q^{n \times m}$, $\hat{A} \in \mathbb{F}_q^{N \times n}$, $y \in \mathbb{F}_q^{N \times m}$, $W \in \mathbb{F}_q^{t \times n}$, and $z \in \mathbb{F}_q^{t \times m}$.

($\Longrightarrow$) From (3.4), we have

$$\hat{A} = A + DW \quad \text{and} \quad A = \hat{A} - DW.$$

Applying (2.4) to the above equations, we obtain

$$\mathsf{rank}\, A - t \leq \mathsf{rank}\, \hat{A} \leq \mathsf{rank}\, A + t$$

which implies that $\rho - t \leq \mu \leq \rho + t$. Similarly, observe that

$$Y = (\hat{A} - DW)X + DZ = \hat{A}X + D(Z - WX). \tag{B.14}$$

Applying (2.4) to (3.4) and (B.14), it follows that

$$\mathsf{rank}\, Y \leq \min\{\mathsf{rank}\, A + t,\, \mathsf{rank}\, \hat{A} + t\}$$

i.e., $\delta = \mathsf{rank}\, Y - n + \mu \leq \min\{t - \rho + \mu,\, t\}$. Now, let us determine the possible values of ϵ. From (3.4), we know that

$$\Delta_\rho(X, Y) \leq \Delta_A(X, Y) \leq t.$$

Thus, according to Theorem 5.1 and Theorem 4.16,

$$\begin{aligned}
\epsilon &= d_{\mathrm{I}}(\langle X \rangle, \langle Y \rangle) + \min\{\mu, \delta\} - \mu - \delta \\
&= \max\{\mathsf{rank}\, X,\, \mathsf{rank}\, Y\} - \mathsf{dim}\,(\langle X \rangle \cap \langle Y \rangle) - \max\{\mu, \delta\} \\
&= \max\{n,\, n - \mu + \delta\} + \Delta_\rho(X, Y) - \max\{n - \rho,\, n - \mu + \delta\} - \max\{\mu, \delta\} \\
&= \max\{\mu,\, \delta\} + \Delta_\rho(X, Y) - \max\{\mu - \rho,\, \delta\} - \max\{\mu, \delta\} \\
&= \Delta_\rho(X, Y) - \max\{\mu - \rho,\, \delta\} \\
&\leq t - \max\{\mu - \rho,\, \delta\}.
\end{aligned}$$

The remaining inequalities for μ, δ and ϵ are trivially satisfied.

($\Longleftarrow$) Let us choose $z = \bar{z} + Wx$ for some $\bar{z} \in \mathbb{F}_q^{t\times m}$. From (B.14), note that $y = \hat{A}x + D(z - Wx) = \hat{A} + D\bar{z}$. We have

$$\begin{aligned}
\mathsf{rank} \begin{bmatrix} X \\ Y \end{bmatrix} &= \mathsf{rank} \begin{bmatrix} X \\ AX + DZ \end{bmatrix} \\
&= \mathsf{rank} \begin{bmatrix} X \\ DZ \end{bmatrix} \\
&= \mathsf{rank} \begin{bmatrix} I & x \\ DW & Dz \end{bmatrix} \\
&= \mathsf{rank} \begin{bmatrix} I & x \\ 0 & Dz - DWx \end{bmatrix} \\
&= n + \mathsf{rank}\ D\bar{z}.
\end{aligned}$$

Suppose that $\mathsf{rank}\ \hat{A} = n - \mu$ (this will be obtained later). Then there exists some full-rank matrix $T_2 \in \mathbb{F}_q^{(N-n+\mu)\times N}$ such that $T_2\hat{A} = 0$. Let $T_1 \in \mathbb{F}_q^{(n-\mu)\times N}$ be some matrix such that $T = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix}$ is nonsingular. Note that $\langle T_1 \rangle$ has trivial intersection with the left null space of $\hat{A}$. Then

$$\begin{aligned}
TY &= \begin{bmatrix} T_1 Y \\ T_2 Y \end{bmatrix} \\
&= \begin{bmatrix} T_1\hat{A} & T_1 y \\ T_2\hat{A} & T_2 y \end{bmatrix} \\
&= \begin{bmatrix} T_1\hat{A} & T_1 y \\ 0 & T_2 D\bar{z} \end{bmatrix} \\
&= \mathsf{rank}\ T_1\hat{A} + \mathsf{rank}\ T_2 D\bar{z} \\
&= n - \mu + \mathsf{rank}\ T_2 D\bar{z}.
\end{aligned}$$

Let $A = PQ$ be a full-rank decomposition, where $P \in \mathbb{F}_q^{N\times(n-\rho)}$ and $Q \in \mathbb{F}_q^{(n-\rho)\times n}$. We will now proceed to finding D, W and $\bar{z}$ such that $Y = AX + DZ$ satisfies (5.29)–(5.31). Two cases must be considered.

First, consider the case that $\mu - \rho \geq 0$. In this case, the rank of A must be *decreased* by $\mu - \rho$ units when adding DW to form $\hat{A}$. Let us write $P = \begin{bmatrix} P_1 & P_2 \end{bmatrix}$ and $Q = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix}$, where $P_1 \in \mathbb{F}_q^{N\times(\mu-\rho)}$, $P_2 \in \mathbb{F}_q^{N\times(n-\mu)}$, $Q_1 \in \mathbb{F}_q^{(\mu-\rho)\times n}$, and $Q_2 \in \mathbb{F}_q^{(n-\mu)\times n}$. Let $D = \begin{bmatrix} D_1 & D_2 & D_3 \end{bmatrix}$ and $W = \begin{bmatrix} W_1 & W_2 & W_3 \end{bmatrix}$, where we choose

$$\begin{array}{ll} D_1 = P_1 & W_1 = -Q_1 \\ D_2 \in \mathbb{F}_q^{N\times[\delta-\mu+\rho]^+} \text{ such that } \begin{bmatrix} P & D_2 \end{bmatrix} \text{ is full-rank} & W_2 = 0_{[\delta-\mu+\rho]^+\times n} \\ D_3 \text{ as the first } t - \max\{\mu-\rho,\delta\} \text{ columns of } P_2 & W_3 = 0_{(t-\max\{\mu-\rho,\delta\})\times n}. \end{array}$$

Note that this is always possible according to our assumptions. Also note that the resulting D is full-rank. It follows that

$$\hat{A} = P_1Q_1 + P_2Q_2 + D_1W_1 + D_2W_2 + D_3W_3 = P_2Q_2.$$

Thus, $\mathsf{rank}\ \hat{A} = n - \mu$. Moreover, since $T_2P_2 = 0$, we have

$$T_2D = T_2 \begin{bmatrix} D_1 & D_2 & D_3 \end{bmatrix} = \begin{bmatrix} T_2D_1 & T_2D_2 & 0 \end{bmatrix}$$

and, by construction, $T_2 \begin{bmatrix} D_1 & D_2 \end{bmatrix}$ is full-rank. Let us write $\bar{z} = \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \end{bmatrix}$, where $\bar{z}_1$ and $\bar{z}_2$ have $\max\{\mu-\rho,\delta\}$ and $t - \max\{\mu-\rho,\delta\}$ rows, respectively. It follows that $\mathsf{rank}\, Y = n - \mu + \mathsf{rank}\, \bar{z}_1$. By choosing $\bar{z}_1$ with rank δ and $\bar{z}_2$ such that $\mathsf{rank}\, \bar{z} = \delta + \epsilon$ (which is possible according to our assumptions), we obtain the desired result.

Now, let us consider the case that $\rho - \mu \geq 0$. In this case, the addition of DW should have the effect of *increasing* the rank of A by $\rho - \mu$ units. As before, let $D =$

$\begin{bmatrix} D_1 & D_2 & D_3 \end{bmatrix}$ and $W = \begin{bmatrix} W_1 & W_2 & W_3 \end{bmatrix}$, but now choose

$$
\begin{aligned}
&D_1 \in \mathbb{F}_q^{N \times (\rho - \mu)} \text{ such that } \begin{bmatrix} P & D_1 \end{bmatrix} \text{ is full-rank} \\
&W_1 \in \mathbb{F}_q^{(\rho-\mu) \times n} \text{ such that } \begin{bmatrix} Q \\ W_1 \end{bmatrix} \text{ is full-rank} \\
&D_2 \in \mathbb{F}_q^{N \times \delta} \text{ such that } \begin{bmatrix} D_1 & D_2 \end{bmatrix} \text{ is full-rank} \\
&W_2 = 0_{\delta \times n} \\
&D_3 \text{ as the first } t - \delta - \rho + \mu \text{ columns of } P \\
&W_2 = 0_{(t-\delta-\rho+\mu) \times n}.
\end{aligned}
$$

Note that this is always possible according to our assumptions. It follows that

$$\hat{A} = PQ + D_1 W_1 + D_2 W_2 + D_3 W_3 = \begin{bmatrix} P & D_1 \end{bmatrix} \begin{bmatrix} Q \\ W_1 \end{bmatrix}.$$

Thus, $\mathsf{rank}\ \hat{A} = n - \mu$. Moreover, since $T_2 \begin{bmatrix} P & D_1 \end{bmatrix} = 0$, we have

$$T_2 D = T_2 \begin{bmatrix} D_1 & D_2 & D_3 \end{bmatrix} = \begin{bmatrix} 0 & T_2 D_2 & 0 \end{bmatrix}$$

and, by construction, $T_2 D_2$ is full-rank. Let us write $\bar{z} = \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \\ \bar{z}_3 \end{bmatrix}$, where $\bar{z}_1$, $\bar{z}_2$, $\bar{z}_3$ have $\rho - \mu$, δ, and $t - \delta - \rho + \mu$ rows, respectively. It follows that $\mathsf{rank}\ Y = n - \mu + \mathsf{rank}\ \bar{z}_2$. By choosing $\bar{z}_2$ with rank δ and $\bar{z}_1$ and $\bar{z}_3$ such that $\mathsf{rank}\ \bar{z} = \delta + \epsilon$ (which is possible according to our assumptions), we obtain the desired result. ∎

B.3 Proofs for Chapter 8

Proof (of Proposition 8.1): To prove 1), let $\mathcal{W} = \{Bx \colon x \in F^n\}$ and

$$\mathcal{X}_{s,w} = \left\{ x \in F^n \colon \begin{bmatrix} s \\ w \end{bmatrix} = \begin{bmatrix} H \\ B \end{bmatrix} x \right\}.$$

Observe that

$$H(W) \leq \log_{|F|} |\mathcal{W}| = \mathsf{rank}\ B$$

$$H(X|S) = \log_{|F|} |\mathcal{X}_S| = \mathsf{dim}\ \ker(H) = n - \mathsf{rank}\ H$$

$$H(X|S,W) \leq \log_{|F|} |\mathcal{X}_{S,W}| = n - \mathsf{rank} \begin{bmatrix} H \\ B \end{bmatrix}.$$

From (8.3), it follows that

$$I(S;W) \leq \mathsf{rank}\ H + \mathsf{rank}\ B - \mathsf{rank} \begin{bmatrix} H \\ B \end{bmatrix}.$$

Now, to prove 2), first note that

$$\mathsf{dim}\,(\langle H \rangle \cap \langle B \rangle) = \mathsf{rank} \begin{bmatrix} H \\ B \end{bmatrix} - \mathsf{rank}\ H - \mathsf{rank}\ B.$$

Suppose $\mathsf{dim}\,(\langle H \rangle \cap \langle B \rangle) = t > 0$. Then there exist full-rank matrices $T_1 \in F^{t \times \mu}$ and $T_2 \in F^{t \times k}$ such that $T_1 B = T_2 H$. This implies that

$$T_1 W = T_1 B X = T_2 H X = T_2 S.$$

Since S is uniform, we have that $I(S;W) \geq t > 0$. ∎

Proof (of Lemma 8.4): Assume $I(S;W) = 0$. Using (8.3), it follows that

$$H(X|S,W) = H(X|S) - H(W).$$

We will now show that $H(W) - H(X|S) \geq 0$, which implies that $H(X|S,W) = 0$.

We have

$$\begin{aligned} H(W) - H(X|S) &= E\left[-\log p_W(W) + \log p_{X|S}(X|S)\right] \\ &= E\left[\log \frac{p_{X|S}(X|S)}{p_W(W)}\right] \\ &= E\left[\log \frac{p_{S|X}(S|X)p_X(X)}{p_W(W)p_S(S)}\right] \\ &= E\left[\log \frac{p_{S|X}(f(X)|X)p_X(X)}{p_W(g(X))p_S(f(X))}\right] \\ &= E\left[\log \frac{p_X(X)}{p_W(g(X))p_S(f(X))}\right] \\ &= D\big(p_X(x) \parallel p_W(g(x))p_S(f(x))\big) \end{aligned}$$

where $D(\cdot \parallel \cdot)$ denotes the relative entropy [64].

It remains to prove that $p_W(g(x))p_S(f(x))$ is indeed a probability mass function on $\mathcal{X}$, i.e., that

$$\sum_{x\in\mathcal{X}} p_W(g(x))p_S(f(x)) = 1. \tag{B.26}$$

Then the result will follow from the nonnegativity of the relative entropy.

Since S is uniform, we have that $p_S(f(x)) = 1/|\mathcal{S}|$, $\forall x$. Moreover,

$$\begin{aligned} \sum_{x\in\mathcal{X}} p_W(g(x)) &= \sum_{w\in\mathcal{W}} \sum_{\substack{x\in\mathcal{X}:\\ g(x)=w}} p_W(g(x)) \\ &= \sum_{w\in\mathcal{W}} p_W(w) \sum_{\substack{x\in\mathcal{X}:\\ g(x)=w}} 1 \\ &= \sum_{w\in\mathcal{W}} p_W(w)\,|\{x\in\mathcal{X}\colon g(x)=w\}| \\ &= |\mathcal{S}|. \end{aligned}$$

Thus, (B.26) holds and the proof is complete. ∎

Bibliography

[1] M. Effros, R. Koetter, and M. Médard, "Breaking network logjams," *Scientific American*, vol. 6, pp. 78–85, Jun. 2007.

[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] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE Int. Symp. Information Theory*, Yokohama, Japan, Jun. 29–Jul. 4, 2003, p. 442.

[6] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[7] L. L. Peterson and B. S. Davie, *Computer Networks: A Systems Approach*, 4th ed. San Francisco, CA: Morgan Kaufmann, 2007.

[8] "The network coding homepage." [Online]. Available: http://www.ifp.uiuc.edu/ koetter/NWC/Bibliography.html

[9] S. Katti, D. Katabi, H. Balakrishnan, and M. Médard, "Symbol-level network coding for wireless mesh networks," in *ACM SIGCOMM*, Seattle, WA, Aug. 2008.

[10] N. Cai and R. W. Yeung, "Network coding and error correction," in *Proc. 2002 IEEE Inform. Theory Workshop*, Bangalore, India, Oct. 20–25, 2002, pp. 119–122.

[11] R. W. Yeung and N. Cai, "Network error correction, part I: Basic concepts and upper bounds," *Commun. Inform. Syst.*, vol. 6, no. 1, pp. 19–36, 2006.

[12] N. Cai and R. W. Yeung, "Network error correction, part II: Lower bounds," *Commun. Inform. Syst.*, vol. 6, no. 1, pp. 37–54, 2006.

[13] Z. Zhang, "Network error correction coding in packetized networks," in *Proc. 2006 IEEE Inform. Theory Workshop*, Chengdu, China, Oct. 22–26, 2006, pp. 433–437.

[14] ——, "Linear network error correction codes in packet networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 209–218, 2008.

[15] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, and M. Médard, "Resilient network coding in the presence of Byzantine adversaries," in *Proc. 26th IEEE Int. Conf. on Computer Commun.*, Anchorage, AK, May 2007, pp. 616–624.

[16] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Médard, and M. Effros, "Resilient network coding in the presence of Byzantine adversaries," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2596–2603, Jun. 2008.

[17] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," in *Proc. IEEE Int. Symp. Information Theory*, Nice, France, Jun. 24–29, 2007, pp. 791–795.

[18] 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.

[19] L. H. Ozarow and A. D. Wyner, "Wire-tap channel II," in *Proc. EUROCRYPT 84 workshop on Advances in cryptology: theory and applicationof cryptographic techniques*. New York, NY, USA: Springer-Verlag New York, Inc., 1985, pp. 33–51.

[20] N. Cai and R. W. Yeung, "Secure network coding," in *Proc. IEEE Int. Symp. Information Theory*, Lausanne, Switzerland, Jun. 30–Jul. 5, 2002, p. 323.

[21] J. Feldman, T. Malkin, C. Stein, and R. A. Servedio, "On the capacity of secure network coding," in *Proc. 42nd Annual Allerton Conf. on Commun., Control, and Computing*, Sep. 2004.

[22] S. Y. E. Rouayheb and E. Soljanin, "On wiretap networks II," in *Proc. IEEE Int. Symp. Information Theory*, Nice, France, Jun. 24–29, 2007, pp. 551–555.

[23] A. Montanari and R. Urbanke, "Iterative coding for network coding," 2007, submitted for publication. [Online]. Available: http://arxiv.org/abs/0711.3935

[24] K. Bhattad and K. R. Narayanan, "Weakly secure network coding," in *Proc. NetCod 2005*, Riva del Garda, Italy, Apr. 2005.

[25] D. Silva and F. R. Kschischang, "Using rank-metric codes for error correction in random network coding," in *Proc. IEEE Int. Symp. Information Theory*, Nice, France, Jun. 24–29, 2007, pp. 796–800.

[26] D. Silva, F. R. Kschischang, and R. Koetter, "A rank-metric approach to error control in random network coding," in *Proc. IEEE Information Theory Workshop on Information Theory for Wireless Networks*, Bergen, Norway, Jul. 1–6, 2007.

[27] D. Silva, F. R. Kschischang, and R. Kötter, "A rank-metric approach to error control in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 3951–3967, 2008.

[28] D. Silva and F. R. Kschischang, "Adversarial error correction for network coding: Models and metrics," in *Proc. 46th Annual Allerton Conf. on Commun., Control, and Computing*, Monticello, IL, Sep. 23-26, 2008, pp. 1246–1253.

[29] ——, "On metrics for error correction in network coding," *IEEE Trans. Inf. Theory*, 2008, to be published. [Online]. Available: http://arxiv.org/abs/0805.3824

[30] ——, "Fast encoding and decoding of Gabidulin codes," in *Proc. IEEE Int. Symp. Information Theory*, Seoul, Korea, Jun. 28–Jul. 3, 2009, pp. 2858–2862.

[31] D. Silva, F. R. Kschischang, and R. Kötter, "Capacity of random network coding under a probabilistic error model," in *Proc. 24th Biennial Symp. Communications*, Kingston, Ontario, Canada, Jun. 24–26, 2008, pp. 9–12.

[32] ——, "Communication over finite-field matrix channels," *IEEE Trans. Inf. Theory*, 2008, to be published. [Online]. Available: http://arxiv.org/abs/0807.1372

[33] D. Silva and F. R. Kschischang, "Security for wiretap networks via rank-metric codes," in *Proc. IEEE Int. Symp. Information Theory*, Toronto, Canada, Jul. 6–11, 2008, pp. 176–180.

[34] ——, "Universal weakly secure network coding," in *Proc. Inform. Theory Workshop on Networking and Inform. Theory*, Volos, Greece, Jun. 10-12, 2009, pp. 281–285.

[35] ——, "Universal secure network coding via rank-metric codes," *IEEE Trans. Inf. Theory*, 2008, submitted for publication. [Online]. Available: http://arxiv.org/abs/0809.3546

[36] A. R. Rao and P. Bhimasankaram, *Linear Algebra*, 2nd ed. New Delhi, India: Hindustan Book Agency, 2000.

[37] E. M. Gabidulin, "Theory of codes with maximum rank distance," *Probl. Inform. Transm.*, vol. 21, no. 1, pp. 1–12, 1985.

[38] R. Lidl and H. Niederreiter, *Finite Fields*. Reading, MA: Addison-Wesley, 1983.

[39] S. H. Friedberg, A. J. Insel, and L. E. Spence, *Linear Algebra*, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2003.

[40] R. M. Roth, "Maximum-rank array codes and their application to crisscross error correction," *IEEE Trans. Inf. Theory*, vol. 37, pp. 328–336, 1991.

[41] P. Delsarte, "Bilinear forms over a finite field, with applications to coding theory," *J. of Comb. Theory. Series A*, vol. 25, pp. 226–241, 1978.

[42] P. Loidreau, "Étude et optimisation de cryptosystèmes à clé publique fondés sur la théorie des codes correcteurs," Ph.D. Dissertation, École Polytechnique, Paris, France, May 2001.

[43] M. Gadouleau and Z. Yan, "Packing and covering properties of rank metric codes," *IEEE Trans. Inf. Theory*, vol. 54, no. 9, pp. 3873–3883, 2008.

[44] S. Gao, "Normal bases over finite fields," Ph.D. dissertation, University of Waterloo, Department of Combinatorics and Optimization, 1993.

[45] S. Gao, J. von zur Gathen, D. Panario, and V. Shoup, "Algorithms for exponentiation in finite fields," *J. Symbolic Computation*, vol. 29, pp. 879–889, 2000.

[46] J. von zur Gathen and J. Gerhard, *Modern Computer Algebra*, 2nd ed. New York: Cambridge University Press, 2003.

[47] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. Allerton Conf. on Comm., Control, and Computing*, Monticello, IL, Oct. 2003, pp. 40–49.

[48] S. Yang and R. W. Yeung, "Characterizations of network error correction/detection and erasure correction," in *Proc. NetCod 2007*, San Diego, CA, Jan. 2007.

[49] S. Jaggi, P. Sanders, P. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1973–1982, Jun. 2005.

[50] S. Yang, R. W. Yeung, and Z. Zhang, "Weight properties of network codes," *European Transactions on Telecommunications*, vol. 19, no. 4, pp. 371–383, 2008.

[51] S. Yang and R. W. Yeung, "Refined coding bounds for network error correction," in *Proc. 2007 IEEE Information Theory Workshop*, Bergen, Norway, Jul. 1–6, 2007, pp. 1–5.

[52] S.-T. Xia and F.-W. Fu, "Johnson type bounds on constant dimension codes," *Designs, Codes and Cryptography*, vol. 50, no. 2, pp. 163–172, Feb. 2009.

[53] H. Wang, C. Xing, and R. Safavi-Naini, "Linear authentication codes: bounds and constructions," *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 866–872, 2003.

[54] E. M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov, "Rank errors and rank erasures correction," in *Proc. 4th Int. Colloq. Coding Theory, Dilijan, Armenia, 1991*, Yerevan, 1992, pp. 11–19.

[55] E. M. Gabidulin and N. I. Pilipchuk, "A new method of erasure correction by rank codes," in *Proc. IEEE Int. Symp. Information Theory*, Jun. 29–Jul. 4, 2003, p. 423.

[56] G. Richter and S. Plass, "Error and erasure decoding of rank-codes with a modified Berlekamp-Massey algorithm," in *Proc. ITG Conf. on Source and Channel Coding*, Erlangen, Germany, Jan. 2004, pp. 249–256.

[57] ——, "Fast decoding of rank-codes with rank errors and column erasures," in *Proc. IEEE Int. Symp. Information Theory*, Jun. 27–Jul. 2, 2004, pp. 398–398.

[58] E. M. Gabidulin and N. I. Pilipchuk, "Error and erasure correcting algorithms for rank codes," *Designs, Codes and Cryptography*, vol. 49, no. 1-3, pp. 105–122, Dec. 2008.

[59] R. E. Blahut, "Transform techniques for error control codes," *IBM J. Res. Develop.*, vol. 23, pp. 299–315, May 1979.

[60] ——, *Algebraic codes for data transmission*. Cambridge, UK: Cambridge Univ. Press, 2003.

[61] V. Skachek and R. M. Roth, "Probabilistic algorithm for finding roots of linearized polynomials," *Designs, Codes and Cryptography*, vol. 46, no. 1, pp. 17–23, 2008, also in Comput. Sci. Dept., Technion, Tech. Rep. CS-2004-08, Jun. 2004.

[62] E. R. Berlekamp, *Algebraic Coding Theory*. New York: McGraw-Hill, 1968.

[63] M. Gadouleau and Z. Yan, "Complexity of decoding Gabidulin codes," in *Proc. Annual Conf. Inform. Sciences and Syst.*, Princeton, NJ, Mar. 19–21, 2008, pp. 1081–1085.

[64] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. John Wiley & sons, 1991.

[65] M. Siavoshani, C. Fragouli, and S. Diggavi, "Noncoherent multisource network coding," in *Proc. IEEE Int. Symp. Information Theory*, Toronto, Canada, Jul. 6–11, 2008, pp. 817–821.

[66] D. S. Dummit and R. M. Foote, *Abstract Algebra*, 3rd ed. John Wiley & Sons, 2004.

[67] D. Silva and F. R. Kschischang, "A key-based error control scheme for network coding," in *Proc. 11th Canadian Workshop Inform. Theory*, Ottawa, Canada, May 13-15, 2009, pp. 5–8.

[68] S. Jaggi and M. Langberg, "Resilient network codes in the presence of eavesdropping Byzantine adversaries," in *Proc. IEEE Int. Symp. Information Theory*, 24–29 June 2007, pp. 541–545.

[69] R. G. Gallager, *Principles of Digital Communication*. Cambridge, UK: Cambridge Univ. Press, 2008.

[70] A. Albanese, J. Blömer, J. Edmonds, M. Luby, and M. Sudan, "Priority encoding transmission," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1737–1744, Nov. 1996.

[71] D. Silva and F. R. Kschischang, "Rank-metric codes for priority encoding transmission with network coding," in *Proc. 10th Canadian Workshop Inform. Theory*, Edmonton, Alberta, Canada, Jun. 6–8, 2007, pp. 81–84.

[72] R. W. Nobrega and B. F. Uchoa-Filho, "Multishot codes for network coding: Bounds and a multilevel construction," in *Proc. IEEE Int. Symp. Information Theory*, Seoul, Korea, Jun. 28–Jul. 3, 2009, pp. 428–432.

[73] L. Nutman and M. Langberg, "Adversarial models and resilient schemes for network coding," in *Proc. IEEE Int. Symp. Information Theory*, Toronto, Canada, Jul. 6–11, 2008, pp. 171–175.

[74] T. Etzion and N. Silberstein, "Error-correcting codes in projective spaces via rank-metric codes and Ferrers diagrams," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 2909–2919, July 2009.

[75] T. Etzion and A. Vardy, "Error-correcting codes in projective space," in *Proc. IEEE Int. Symp. Information Theory*, Toronto, Canada, Jul. 6–11, 2008, pp. 871–875.

[76] A. Kohnert and S. Kurz, "Construction of large constant dimension codes with a prescribed minimum distance," *Lecture Notes in Computer Science*, vol. 5393, pp. 31–42, 2008.

[77] M. Gadouleau and Z. Yan, "Packing and covering properties of subspace codes," in *Proc. IEEE Int. Symp. Information Theory*, Seoul, Korea, Jun. 28–Jul. 3, 2009, pp. 2867–2871.

[78] ——, "Construction and covering properties of constant-dimension codes," in *Proc. IEEE Int. Symp. Information Theory*, Seoul, Korea, Jun. 28–Jul. 3, 2009, pp. 2221–2225.

[79] A. Khaleghi and F. R. Kschischang, "Projective space codes for the injection metric," in *Proc. 11th Canadian Workshop Inform. Theory*, Ottawa, Canada, May 13–15, 2009, pp. 9–12.