Exploiting Interference through Algebraic Structure

Bobak Anthony Nazer

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2009-186
http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-186.html

December 18, 2009

Exploiting Interference through Algebraic Structure

by

Bobak Anthony Nazer

A dissertation submitted in partial satisfaction

of the requirements for the degree of

Doctor of Philosophy

in

Engineering—Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael C. Gastpar, Chair
Professor Kannan Ramchandran
Professor Steven N. Evans

Fall 2009

The dissertation of Bobak Anthony Nazer, titled Exploiting Interference through Algebraic Structure, is approved:

Chair ____ Date ________________

____ Date ________________

____ Date ________________

University of California, Berkeley

Exploiting Interference through Algebraic Structure

by

Bobak Anthony Nazer

Abstract

Exploiting Interference through Algebraic Structure

by

Bobak Anthony Nazer

Doctor of Philosophy in Engineering—Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Michael C. Gastpar, Chair

In a network, interference between transmitters is usually viewed as highly undesirable and clever algorithms and protocols have been devised to avoid it. Collectively, these strategies transform the physical layer into a set of reliable bit pipes which can then be used seamlessly by higher layers in the protocol stack. Unfortunately, interference avoidance results in sharply decreasing rates as the number of users increases. In this thesis, we develop a new tool, computation coding, that allows receivers to reliably decode equations of transmitted messages by harnessing the interference structure of the channel. Applied to a wireless network, this enables relays to decode linear functions of the transmitted messages with coefficients dictated by the fading realization. Relays can then forward these equations towards the destinations which simply collect enough equations to solve for their desired messages. Structured codes (such as lattices) ensure that these linear combinations can be decoded reliably at the relays, often at far higher rates than the messages individually. Through examples drawn from cooperative communication including cellular uplink, distributed MIMO and wireless network coding, we demonstrate that this compute-and-forward strategy can improve end-to-end throughput in a network. As a consequence, we find that structured codes can play an important role in approaching the capacity of networks. We also show that our techniques can result in both energy and delay savings for distributed signal processing over a sensor network. Finally, by viewing interference as implicit computation, we provide a new perspective on the interference channel with time-varying fading. We describe a simple interference alignment scheme that enables each user to achieve at least half its interference-free capacity at any signal-to-noise ratio.

Acknowledgements

It has been a true pleasure to have been one of the first graduate students of Michael Gastpar. Over the past six years, I have gained a deep respect for Michael's principles and philosophy about life and research. Michael taught me the power of a good example and to always keep searching for something bigger. Without his endless patience (and gentle prodding), I would have never found this dissertation topic. Thanks for letting me wander and letting me know when I was on the right course.

I have been quite fortunate to have Kannan Ramchandran as a mentor and a member of my thesis committee. I have enjoyed our many chats, especially since he has always been happy to share his (magnificent) view of the "big picture." I was also very lucky to have David Tse on my exam committee. David has a unique way of distilling research problems down to their essence and this has always been an inspiration to me in my research. I also owe a debt of gratitude to Steven Evans for serving as my outside thesis committee member.

I would have never found my way to this research field if not for the wonderful teaching and mentorship of my ECE professors at Rice University. I'm still enjoying the journey thanks to Behnaam Aazhang, Rich Baraniuk, Don Johnson, and Rob Nowak: I really did learn all I needed to know in 241!

Several of the results in this thesis were obtained in collaboration with some wonderful co-authors including Alex Dimakis, Uri Erez, Syed Jafar, Amichai Sanderovich, Anand Sarwate, Shlomo Shamai, Sriram Vishwanath, and Jiening Zhan. Thanks for all of your kindness, patience, and hard work! I would also like to acknowledge Krish Eswaran and Galen Reeves for much appreciated help in a couple of proofs along the way.

It is hard to imagine life at Berkeley without Wireless Foundations, the common office space that the communications faculty, postdocs, and students share. I enjoyed both the quiet times and the "less" quiet times. Special thanks to Salman Avestimehr, Guy Bresler, Cheng Chang, Alex Dimakis, Stark Draper, Lara Dolecek, Krish Eswaran, Raul Etkin, Allie Fletcher, Amin Gohari, Lenny Grokop, Pulkit Grover, Dan Hazen, Mark Johnson, Mohammad Ali Maddah-Ali, Paolo Minero, Sahand Negahban, Dapo Omidiran, Hari Palaiyanur, Vinod Prabhakaran, Galen Reeves, Prasad Santhanam, Anand Sarwate, Dan Schonberg, Rahul Tandra, Parv Venkitasubramaniam, Aaron Wagner, June Wang, Wei Wang, and Jiening Zhan for all the memories.

Thanks to Sheila Ross for the chance to teach a (huge) discussion section for EE20N. I also had a great time in EE120 with Michael Gastpar and Alex Dimakis.

Like many other EECS graduate students, I owe a lot to Ruth Gjerde for her constant help in navigating the Berkeley bureaucracy. Thanks also to Amy Ng and Kim Kail who were the glue that kept Wireless Foundations together as well.

I spent nearly every Friday looking forward to the Battlestar Galactica Dinner Group. Without Drew Carlson, Krish Eswaran, Dave Gorin, David Hembry, Craig Hetherington, Stephanie Hsia, Jennifer Imamura, Amy Jewel, Darcy Krasne, Alex Rennet, Prasad Santhanam, and Anand Sarwate this could not have been the event that it was. Thanks also to Ryan Aipperspach for the Bay Area Rice Alumni Happy Hour and Jay Taneja for Beer Night.

I'll always remember the long conversations with Krish Eswaran and the inside jokes. Alex Dimakis was there for a laugh, no matter the time of day (or time to the deadline). Thanks to Hari Palaiyanur for the countless Costco social hour food trips and the endless pursuit of the perfect M&M cookie. Anand Sarwate is a true Renaissance man and I enjoyed many dinners, concerts, and plays with him. I never found myself alone, racing towards the last minute, thanks to Rahul Tandra.

I have been very lucky to have a wonderful family. My brother Ramin sent me new humor and music whenever the need was great. My cousin Ali was just across the bay and introduced me to some fantastic restaurants. My grandparents have always been there with kindness and generosity. I am forever grateful to my parents Nassrin and Yahya for their constant love, support, and encouragement. They gave me their love of learning and instilled a passion for science in me at an early age. Thanks for letting me twist the stereo knobs all those years ago.

This dissertation would have never been finished without the amazing love and support of Stephanie Hsia. She gave me the gift of time when I needed it most (even in Hawai'i) and has made every day fun in its own way.

To my parents and Stephanie,

Contents

Chapter 1

Introduction

Modern communication systems are designed and analyzed through a ubiquitous currency of information: *the bit*. This is due to Claude Shannon's groundbreaking 1948 treatise which showed that communication of an information source over a noisy channel can be completely understood in terms of bits [142]. Specifically, it is optimal to convert the noisy channel into a reliable bit pipe (through a channel code), compress the information source into a representation in bits (through a source code), and then connect these two components together. Thus, from a theoretical perspective, bits can serve as a universal interface between a single transmitter and a single receiver. In practice, this digital interface has fueled the successful deployment of far larger systems consisting of many transmitters and many receivers, such as cellular telephone networks and the Internet.

When designing a communication network, the usual approach is to establish reliable bit pipes between pairs of users and then route information along the resulting graph of bit pipes. This is a natural extension of the approach developed by Shannon for a single noisy channel. However, in a wireless network, a transmission from a single node is heard not only by the intended receiver, but also by all other nearby nodes; by analogy, any receiver not only captures the signal from its designated transmitter, but from all other nearby transmitters. The signal observed at each receiver can be modeled as a function of the transmitted signals corrupted by some random noise process. For instance, a good approximation to a narrowband wireless channel is that receivers observe a linear combination of the transmitted signals plus Gaussian noise. Given this noisy function, the receiver attempts to extract its desired signal while treating the other signals as unwanted *interference*. Thus, as the number of active users increases, the signal-to-interference-and-noise ratio decreases and the data rate available to each user plummets. Another alternative is for each user to take a turn transmitting while the rest stay silent but this again results in decreasing data rates as more users join the network.

In this thesis, we take an alternative view of interference as *implicit computation*. Rather than trying to extract one message from the noisy combination of transmitted signals, the receiver attempts to reliably decode a function of the transmitted messages. We will show that if the desired function is close enough to the function naturally provided by the channel, then it can be recovered very efficiently. This approach is clearly quite useful in network scenarios, such as sensor networks, where the objective is not to gather all the data but just a function thereof. Our primary goal is to describe its applicability to data networks where each user only wants to reliably recover one or more messages from other users. The basic architecture is quite simple: users decode equations and pass them along towards the intended destinations which, given sufficiently many equations, can recover their desired messages. We will show that this *compute-and-forward* strategy can improve end-to-end bit rates in wireless networks. As we will see, one fascinating by-product is that one must pay close attention to the algebraic structure of codes in network information theory. Below, we set forth the main themes of this thesis and our primary contributions towards them.

Themes

- **Computation over Noisy Channels.** Consider a destination that wants to compute a function of data from several sources. If these users are separated by noisy, interfering links, then the standard approach is to encode the data for reliable communication using a channel code and send it all to the destination which then computes the desired function. Even though in some cases the channel may naturally provide a noisy version of the desired function, it has been tacitly assumed that this could not help the destination compute in an error-free fashion. Here, we show that noisy functions can in fact be harnessed for reliable computation, provided the channel codes are structured appropriately. We will investigate the fundamental limits of computing a function over a channel, the *computation capacity*, and, though the problem is very difficult in general, we find achievable strategies and outer bounds that match in some special cases.

- **Cooperative Communication.** The basic idea underlying cooperative strategies is the following: users in a network help each other achieve their respective objectives. While it is not surprising that this is superior to a non-interference policy, the real advantage lies in synergistic gains. In some important network scenarios, these gains can be extremely large. Examples include:
  - *Distributed beam-forming*: Helpful users in a wireless network listen for a transmission and then simultaneously transmit their (noisy) observations. The resulting coherent combining of the original waveform provides a significant boost in the receiver's signal-to-noise ratio [55; 114; 97].
  - *Distributed MIMO (multiple-input and multiple-output)*: Users in an ad hoc network team up to form distributed multiple-antenna arrays to benefit from the well-known MIMO antenna gain [36; 130; 118].

  - *Cooperative diversity*: Two users in a wireless network learn each other's messages and then transmit both messages jointly. If one user is in a deep fade, the other may still be able to relay its message to the receiver [83; 137].
  - *Network coding*: Users collect packets and, rather than simply forwarding them, mix them according to a linear equation and forward the outcome. This increases throughput rates by satisfying many demands concurrently [4; 76; 69].

  At a relatively small scale, it is becoming increasingly clear how to implement these cooperative techniques in practice [87; 98]. What is unclear is how to weave these strategies into the fabric of a larger network. Substantial research advocates using a cross-layer design that dispenses with the illusion of bit pipes and gives higher layers in the network stack direct access to the physical medium. However, this would negate many of the advantages of a modular design [70]. The strategies developed in this thesis provide a natural solution to this problem by permitting a slight revision of the physical layer. Instead of forcing the wireless medium to be a set of reliable bit pipes, we can transform it to a system of reliable linear equations. Through several case studies, we will show we can reap many of the advantages of cooperative strategies while retaining a notion of modularity.

- **Network Information Theory.** The maximum rate of reliable communication across a point-to-point channel (also referred to as the capacity) was completely characterized by Shannon [142]. For the past several decades, researchers have attempted to generalize this result to find the *capacity region* of networks with several transmitters and receivers. There have been several successes including the multiple-access channel (many-to-one) [3; 86] and the stochastically degraded broadcast channel (one-to-many) [31]. However, in general, the problem is still wide open. We will argue that this is, in part, due to a "missing ingredient" in the standard achievability proofs: algebraic structure.

- **Distributed Signal Processing.** There is an emerging body of work on how to implement centralized signal processing algorithms in a decentralized fashion over a communication network. Many of these studies use a bit pipe abstraction of the physical layer and then develop quantization schemes to run the algorithm over the network. Using the tools developed in this thesis, some of the required computations can be carried out directly on the channel. In some cases, this leads to new distributed signal processing methods that significantly reduce the total energy and delay costs.

Contributions

- **Computation Codes.** Standard channel codes are designed to keep users' messages separated as they pass through a common channel. We propose a new set of coding techniques, *computation codes*, that provide error protection while harnessing the natural computation performed by the channel. Take the case of M transmitters that communicate across a noisy adder to a receiver that wants the error-free sum of the messages. In this example, our computation coding strategy assigns the same linear code to all transmitters. Since the sum of codewords in a linear code is itself a codeword, the sum of the messages will be afforded protection against noise as it traverses the channel. For this example, this leads to an M-fold gain in rate over standard strategies which require sending all the data to the receiver. We develop computation codes for discrete alphabet channels in Chapter 3 and additive white Gaussian noise channels (AWGN) in Chapter 4.

- **Compute-and-Forward.** As mentioned above, cooperative communication schemes have users in a network act as relays to send messages from sources to destinations. Usually, the relay is given the choice of either decoding some part of the message (decode-and-forward) or working directly with the analog observations of messages from the wireless channel (compress-and-forward, amplify-and-forward). Our contribution is that using computation codes relays can also choose to decode a linear function of the messages and send these towards the destination. This provides a digital framework, *compute-and-forward*, for implementing cooperative schemes. In Chapters 3 and 4, we provide the foundations of compute-and-forward for discrete and continuous alphabet channels. One key feature is that channel state information at the transmitters is not required. We explore applications of compute-and-forward in Chapter 5 including network coding over wireless networks, distributed MIMO, and cellular uplink.

- **The Role of Structured Codes.** In order to show the existence of codes that can approach the capacity region of a given channel, most proofs (starting with Shannon's) have relied on the probabilistic method [7]. By evaluating the performance of a random codebook in expectation, one can easily show that at least one good code must exist. The key is to choose the right ensemble of codebooks to randomize over. For a point-to-point channel, it suffices to use a codebook comprised of independent and identically distributed (i.i.d.) codewords [142]. Given a capacity theorem, it is often of interest to demonstrate the existence of a capacity-achieving linear code as this is one step towards a practical implementation of the scheme. However, the conditions for algebraically structured random codes[1] to be capacity-achieving are often more restrictive than those for unrestricted random codes. For instance, linear codes achieve capacity for point-to-point channels only when the noise is symmetric [35; 2]. Thus, it is tempting to believe that i.i.d.. random codes are a strictly more powerful tool for proving

[1] For brevity, we will often shorten "algebraically structured random codes" to "structured random codes" or "structured codes."

capacity theorems. However, an elegant multiterminal problem developed by Körner and Marton showed that purely random code constructions are not always sufficient [77]; structured random codes may be required on the achievability side of the proof. In their problem, a decoder wishes to reconstruct the parity of two correlated binary sources seen by separate encoders. Here, we generalize this observation to show that in a network setting, even if we are only interested in communicating bits from one end to another, structured random codes (especially linear and lattice codes) can be more powerful than purely random codes.

- **Physical-Layer Network Coding.** The seminal paper of Ahlswede et al. demonstrated that mixing packets at relays or *network coding* is required to achieve the multicast capacity of wired networks [4]. For wireless networks, the gain from network coding may be even larger than the gain for wired ones due to the broadcast and multiple-access phenomena. Specifically, when a user transmits a signal, it is seen (multiplied by some amplitude and phase) by all nearby users. Thus, users are automatically given access to many packets. Moreover, due to the superposition of electromagnetic transmissions and the multiple paths to each receiver, simultaneous transmissions are observed by receivers as linear combinations. At a high level, one could say that the wireless medium is performing network coding on the transmitted signals by combining them in a random, linear fashion. However, if we simply operate the network in an uncoded fashion to take advantage of this natural network coding, noise builds up as signals are received and retransmitted and can significantly decrease the overall rate. The tools developed in this thesis can be viewed as error-correcting codes for wireless network coding. Specifically, the codes in Chapter 4 show how to use the complex-valued operations of the wireless channel for computation over a finite field (as required in most network coding schemes). We derive the multicast capacity of finite field multiple-access networks in Section 5.3. We also develop an achievable strategy for multicasting over AWGN networks and show that it performs strictly better than first decoding the packets and then performing network coding.

- **Sensor Network Strategies.** In a sensor network, our goal is to obtain some function of the sensor observations (e.g., the average temperature) either at a fusion center or at individual sensor nodes. Since these nodes communicate across the wireless medium to make their estimates of the desired function, they can exploit the channel to save in both energy and delay. For this purpose, we develop source-channel computation codes that are suited for communicating linear functions of continuous-valued sources (instead of bits). We consider two specific examples. In Section 6.1, sensors have noisy observations of a source that we wish to estimate at a fusion center. We show that if the channel bandwidth is larger than the source bandwidth, computation coding provides substantial energy savings. In Section 6.3, we consider a scenario where each sensor must acquire an estimate of the global average. Gossip algorithms are a robust, distributed strategy for computing averages (and other functions) in sensor networks. Here, we develop a gossip algorithm that exploits interference for faster convergence.

- **Interference Alignment via Computation.** Consider M transmitter-receiver pairs that communicate over a time-varying wireless channel (often referred to as the M-user fast fading interference channel). Each receiver observes a superposition of all transmitted messages from which it must extract its desired message. Standard strategies include orthogonalizing transmissions and treating undesired messages as additive noise. However, these approaches can only provide a capacity per user that scales like $\frac{1}{M}$. Surprisingly, through a new technique known as interference alignment, each user can attain $\frac{1}{2}$ its interference-free capacity as the signal-to-noise ratio (SNR) tends to infinity [21]. In Chapter 7, we give a new perspective on interference alignment based on computation. This results in a simpler scheme that allows each user to achieve at least 1/2 its interference-free capacity at any SNR.

Outline

We begin with a review of i.i.d. random codes and random linear codes as they apply to point-to-point channels in Chapter 2. We also include a full discussion of the Körner-Marton problem [77]. This is the first example of structured codes outperforming i.i.d. random codes and is the inspiration for the main results of this thesis.

Chapter 3 develops computation codes that are optimal for computing linear functions over finite field channels. For general functions and channels, we provide an achievable strategy with examples in Section 3.4. Chapter 4 generalizes computation codes for AWGN networks. Specifically, we use nested lattice codes to perform finite field computations over channels that operate on the complex field. These tools form the underpinnings of our compute-and-forward strategy for networks with interference.

We demonstrate the effectiveness of compute-and-forward for communicating bits across a network in Chapter 5. We begin with an idealized cellular uplink model in Section 5.1. Next, we consider a distributed MIMO problem under slow fading in Section 5.2. Then we delve into network coding by determining the multicast capacity of networks comprised of finite field multiple-access channels in Section 5.3.3.1. Afterwards, we use our lattice computation codes to give achievable rates for AWGN multiple-access networks in Section 5.3.3.2. Finally, in Section 5.4, we take a look at the two-way relay channel with fading and use compute-and-forward to achieve higher outage rates.

Chapter 6 looks at applications of computation coding to distributed signal processing in wireless networks. We show how to exploit the wireless channel when one central terminal wants to estimate a source from many noisy observations in Section 6.1. We also show these gains can be conferred to a completely distributed scenario wherein every sensor must obtain an estimate of the global average without any centralized processing. Specifically, we develop a gossip algorithm that exploits the broadcast and multiple-access properties of the wireless channel for faster convergence in Section 6.3.

In Chapter 7, we develop a new scheme for interference alignment over time-varying channels. The main result is that for the fast fading M-user Gaussian interference channel, each user can achieve at least half its interference-free capacity at any SNR. We also investigate more general message sets such as when each receiver wants messages from more than one transmitter. We end with a look at a finite field interference channel for which we can derive the capacity region and its implication for general interference channels.

Finally, in Chapter 8, we summarize our results and discuss open problems and new directions for this line of work.

Chapter 2

Preliminaries

We now give a brief overview of some key results in information theory and the tools used to derive them. We begin with the capacity theorems for the point-to-point channel and the multiple-access channel. These results are usually derived using i.i.d. random codes but in some special cases random linear codes are also sufficient. Next, we look at lossless compression for a single source and multiple, distributed sources (the Slepian-Wolf problem). These problems are solvable both using i.i.d. random codes and random linear codes. Finally, we discuss the Körner-Marton problem, a distributed compression problem for which the proof relies on random linear codes. This sets the stage for the results in subsequent chapters which all rely on some algebraic structure in the coding scheme.

For completeness, we recall the standard definitions of entropy, conditional entropy, and mutual information from [29, Ch.2]. These quantities have a precise operational meaning in the context of reliable communication systems which we describe below. Note that throughout this thesis, log specifies the logarithm in base 2. We use uppercase letters to denote random variables and superscripts to denote vectors of them. For example, $U^k = (U[1], U[2], \ldots, U[k])$ and $X_j^n = (X_j[1], X_j[2], \ldots, X_j[n])$. We may also denote a vector with a bold, lowercase version of the random variable, where the length can always be inferred from context. For example, $\mathbf{u} = (U[1], U[2], \ldots, U[k])$. Bold, uppercase letters will be used for matrices.

Definition 1 (Entropy)**.** Let $X \in \mathcal{X}$ be a discrete random variable with probability mass function (pmf) $p_X(x)$. Its *entropy* is

$$H(X) = -\sum_{x \in \mathcal{X}} p_X(x) \log p_X(x). \tag{2.1}$$

Definition 2 (Conditional Entropy)**.** Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be discrete random variables with joint pmf $p_{XY}(x,y)$. The *conditional entropy* of X given Y is

$$H(X|Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x,y)\log p_{X|Y}(x|y). \tag{2.2}$$

Definition 3 (Mutual Information)**.** Let $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be discrete random variables with joint pmf $p_{XY}(x,y)$. The *mutual information* of X and Y is

$$I(X;Y) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x,y)\log\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}. \tag{2.3}$$

We also recall the notion of joint typicality.

Definition 4 (Joint Typicality)**.** Given a joint pmf $p_{XY}(x,y)$, the set of *jointly typical* sequences $A_\epsilon^{(n)}$ is the set of all pairs (x^n, y^n) whose empirical entropies are close to the true entropies:

$$\begin{aligned} A_\epsilon^{(n)} = \{&(x^n,y^n) : & (2.4)\\ &\left|-\frac{1}{n}\log p(x^n) - H(X)\right| < \epsilon & (2.5)\\ &\left|-\frac{1}{n}\log p(y^n) - H(Y)\right| < \epsilon & (2.6)\\ &\left|-\frac{1}{n}\log p(x^n,y^n) - H(X,Y)\right| < \epsilon\Big\} & (2.7) \end{aligned}$$

where $p(x^n,y^n) = \prod_{i=1}^n p_{XY}(x_i,y_i)$.

Lemma 1. *Assume that (X^n,Y^n) is generated according to $p(x^n,y^n) = \prod_{i=1}^n p_{XY}(x_i,y_i)$ for some $p_{XY}(x,y)$. The set $A_\epsilon^{(n)}$ satisfies the following three properties:*

1. *$Pr((X^n,Y^n) \in A_\epsilon^{(n)}) \to 1$ as $n \to \infty$.*

2. *$|A_\epsilon^{(n)}| \le 2^{n(H(X,Y)+\epsilon)}$*

3. *If two independent sequences $\tilde{X}^n$ and $\tilde{Y}^n$ have the same marginal distributions as X^n and Y^n, then $Pr((\tilde{X}^n,\tilde{Y}^n) \in A_\epsilon^{(n)}) \le 2^{-n(I(X;Y)-3\epsilon)}$.*

See [29, Theorem 7.6.1] for a proof.

2.1 Point-to-Point Channels

Consider a transmitter and a receiver that wish to communicate over a noisy channel (see Figure 2.1). The goal is to send messages reliably at the highest possible rate. This is the channel's *capacity* and was defined and completely characterized by Shannon [142].

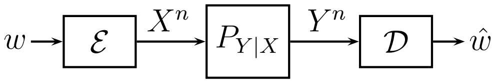

Figure 2.1: Reliable communication across a noisy channel.

More formally, each channel has a finite input alphabet $\mathcal{X}$ and finite output alphabet $\mathcal{Y}$. The probability that an input symbol $x \in \mathcal{X}$ results in an output symbol $y \in \mathcal{Y}$ is specified by $p_{Y|X}(y|x)$. Assume that we are given $n \in \mathbb{Z}_+$ channel uses and that the channel is memoryless so that the output at time i only depends on the input from time i:

$$p(y^n|x^n) = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i) \tag{2.8}$$

The transmitter has a *message* chosen randomly and uniformly from the set $\{1, 2, \ldots, 2^{nR}\}$ for some rate $R > 0$. This message is mapped onto a length n codeword through a fixed encoding function $\mathcal{E} : \{1, 2, \ldots, 2^{nR}\} \to \mathcal{X}^n$. The collection of all possible codewords is called the codebook

$$\mathcal{C} = \Big\{X^n : X^n = \mathcal{E}(w) \text{ for some } w \in \{1, 2, \ldots, 2^{nR}\}\Big\}. \tag{2.9}$$

Once a message is encoded into a codeword X^n, the symbols are sent over the channel. The resulting sequence of channel outputs Y^n is then fed into a decoding function $\mathcal{D} : \mathcal{Y}^n \to \{1, 2, \ldots, 2^{nR}\}$.

The *average probability of error* P_e of a codebook $\mathcal{C}$ is

$$P_e = 2^{-nR} \sum_{w=1}^{2^{nR}} \Pr\left(\mathcal{D}(Y^n) \neq w | X^n = \mathcal{E}(w)\right) \tag{2.10}$$

We say that a rate R is *achievable* if for all $\epsilon > 0$ and n large enough, there exists a deterministic encoding function $\mathcal{E}$ with rate R and associated deterministic decoding function $\mathcal{D}$ with probability of error no greater than ϵ. Thus, as n tends to infinity, the probability of error can be driven to zero. Finally, we define the *capacity* to be the supremum of all achievable rates.

Theorem 1 (Shannon)**.** *The capacity of a discrete memoryless channel is given by*

$$C = \max_{p_X(x)} I(X;Y). \tag{2.11}$$

Proof. To prove this result, we need to show the existence of a sequence of good codebooks whose rates approach C. This is usually called an *achievability* proof. One popular proof technique is known as an i.i.d. random coding argument. We now list some of the main steps and refer the interested reader to [29, Theorem 7.7.1] for more details. Choose some $\epsilon > 0$ and let $p_X(x)$ be the probability mass function that maximizes (2.11). Each of the 2^{nR} codewords is generated i.i.d. according to the product distribution

$$p(x^n) = \prod_{i=1}^{n} p_X(x_i). \tag{2.12}$$

Now assume that a message w is mapped into its codeword $X^n(w)$ and sent over the channel, resulting in channel output Y^n. If given Y^n, there is exactly one X^n in the codebook that is jointly typical, then the decoder outputs the message associated with that codeword. Otherwise, if there is no jointly typical codeword or more than one, it declares an error. Thus, there is an error if $(X^n(w), Y^n)$ is not jointly typical, which by Property 1 of Lemma 1 can be made smaller than $\frac{\epsilon}{2}$ for n large enough. There is also an error if another message's codeword $X^n(\tilde{w})$ for $\tilde{w} \neq w$ is jointly typical with Y^n. Since the codewords are i.i.d. then we can apply Property 3 from Lemma 1 to get:

$$Pr\left((X^n(\tilde{w}), Y^n) \in A_\epsilon^{(n)}\right) \leq 2^{-n(I(X;Y)-3\epsilon)} \tag{2.13}$$

Using the union bound, we bound the average probability of error:

$$\begin{aligned} P_e &\leq Pr\left((X^n(w), Y^n) \notin A_\epsilon^{(n)}\right) + \sum_{\tilde{w}\neq w} Pr\left((X^n(\tilde{w}), Y^n) \in A_\epsilon^{(n)}\right) && (2.14)\\ &< \frac{\epsilon}{2} + \sum_{\tilde{w}\neq w} 2^{-n(I(X;Y)-3\epsilon)} && (2.15)\\ &< \frac{\epsilon}{2} + 2^{-n(I(X;Y)-R-3\epsilon)} && (2.16) \end{aligned}$$

Thus, for n large enough, we can drive the probability of error down to ϵ for any $R < I(X;Y) - 3\epsilon$. Note that this is the probability of error averaged over the randomness in the codebook and we require fixed encoding and decoding functions in our definition of an achievable rate. However, since the average probability of error is at most ϵ then at least one fixed codebook must have probability of error at most ϵ (or else the average would be larger). By taking an appropriate sequence of codebooks, we approach the capacity C arbitrarily closely.

Using Fano's inequality, we can also show the *converse*: no rate greater than C is possible with a vanishing probability of error (see [29, Theorem 7.7.1]). □

We now show that codes with linear structure can, in some special cases, approach the capacity of a point-to-point channel. Let $\mathbb{F}$ denote a finite field.

Definition 5. Assume each codeword $\mathbf{x}$ is a length n vector in a finite field, $\mathbf{x} \in \mathbb{F}^n$. We say a codebook is $\mathcal{C} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{2^{nR}}\}$ is *linear* if the sum of any two codewords is itself a codeword, $\mathbf{x}_a + \mathbf{x}_b \in \mathcal{C} \;\; \forall \mathbf{x}_a, \mathbf{x}_b \in \mathcal{C}$.

Theorem 2 (Elias)**.** *For any discrete memoryless point-to-point channel, any rate $R \leq I(X;Y)$ for X drawn uniformly from $\mathcal{X}$ is achievable using linear codes.*

This result follows naturally by combining the lemma below (generalized from the binary case in [49, §6.2]) and the union bound argument used for the proof of Theorem 1.

Lemma 2. *Let $\mathbf{G} \in \mathbb{F}^{k \times n}$ be a random matrix with i.i.d. uniform entries and $\mathbf{v}$ a vector drawn independently and uniformly over $\mathbb{F}^n$. Then, for any $\mathbf{w}_1 \in \mathbb{F}^k$, $\mathbf{w}_1\mathbf{G} \oplus \mathbf{v}$ is uniformly distributed in $\mathbb{F}^n$ and, for any $\mathbf{w}_2 \neq \mathbf{w}_1 \in \mathbb{F}^k$, $\mathbf{w}_1\mathbf{G} \oplus \mathbf{v}$ and $\mathbf{w}_2\mathbf{G} \oplus \mathbf{v}$ are independent.*

Proof. Given $\mathbf{w}_1$ and $\mathbf{G}$ there is exactly one $\mathbf{v}$ that will map $\mathbf{w}_1$ to each possible $\mathbf{x}_1$. Thus, the probability of mapping $\mathbf{w}_1$ to some $\mathbf{x}_1$ is $P(\mathbf{x}_1) = |\mathbb{F}|^{-n}$.

Since $\mathbf{w}_2 \neq \mathbf{w}_1$, they must differ in at least one position, say the k^{th}. Then, for any $\mathbf{r} = \mathbf{x}_1 + \mathbf{x}_2 \in \mathbb{F}^n$ and any $\mathbf{G}$ with all rows specified except $\mathbf{g_k}$, there is exactly one choice of $\mathbf{g_k}$ such that $\mathbf{w}_1\mathbf{G} + \mathbf{w}_2\mathbf{G} = \mathbf{r}$. For this $\mathbf{G}$, there is only one $\mathbf{v}$ that maps $\mathbf{w}_1\mathbf{G} + \mathbf{v}$ to a given $\mathbf{x}_1$. From these facts, we can show that the probability of mapping $(\mathbf{w}_1, \mathbf{w}_2)$ to a pair $(\mathbf{x_1}, \mathbf{x_2})$ is $P(\mathbf{x_1}, \mathbf{x_2}) = |\mathbb{F}|^{-n(k+1)}|\mathbb{F}|^{n(k-1)} = |\mathbb{F}|^{-2n}$. Thus, input sequences are mapped to output sequences in a pairwise independent fashion. □

Note that in the probability of error analysis in the proof of Theorem 1, we employ a union bound in (2.14) that only requires pairwise independence between codewords. Thus, we can use a random matrix with a random shift to generate our codewords and achieve any rate available under a uniform input distribution. Note that if we do not use the shift, the resulting code has the same performance since this just amounts to a reindexing of the messages. From here, we can show that there exist linear codes that can approach any rate up to $I(X;Y)$ for X drawn from a uniform distribution. Recall the following definition from [49, p.94].

Definition 6. We say that $p_{Y|X}$ is *symmetric* if the output symbols in $\mathcal{Y}$ can be placed into subsets such that for each subset the probability transition matrix satisfies the following two conditions:

1. Each row is a permutation of every other row.

2. Each column is a permutation of every other column.

It is well-known that if $p_{Y|X}$ is symmetric then the uniform input distribution is capacity-achieving. Otherwise, a result due to Ahlswede shows that linear codes cannot achieve capacity for asymmetric channels [2]. Thus, for point-to-point channels, linear codes can, in some cases, approach the capacity but, in general, they cannot. This suggests that i.i.d. random codes are a more versatile tool for proving capacity theorems. In fact, one can use generalizations of the i.i.d. random coding argument to derive the capacity regions of the multiple-access channel [3; 86], stochastically degraded broadcast channel [31], and physically degraded relay channel [32], just to name a few. In all of these situations, linear codes only reach capacity if the noise satisfies certain symmetry conditions.

2.2 Slepian-Wolf Problem

We now review the Slepian-Wolf distributed compression problem.

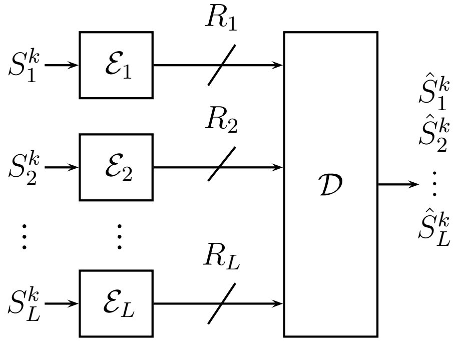

Figure 2.2: Slepian-Wolf Problem

There are L encoders, each of which observes a length k source vector $S_\ell^k \in \mathcal{S}_\ell^k$ (see Figure 2.2). The L-tuple of source vectors $(S_1^k, S_2^k, \ldots, S_L^k)$ is generated i.i.d. according to $p_{S_1 S_2 \cdots S_L}(s_1, s_2, \ldots, s_L)$. Each encoder maps its observation into nR_ℓ bits: $\mathcal{E}_\ell : \mathcal{S}_\ell^k \rightarrow \{1, 2, \ldots, 2^{nR_\ell}\}$. It then sends these bits to the decoder which makes an estimate $(\hat{S}_1^k, \hat{S}_2^k, \ldots, \hat{S}_L^k)$ using its decoding function:

$$\mathcal{D} : \{1, 2, \ldots, 2^{nR_1}\} \times \cdots \times \{1, 2, \ldots, 2^{nR_L}\} \rightarrow \mathcal{S}_1^k \times \cdots \times \mathcal{S}_L^k. \tag{2.17}$$

The probability of error P_e is just the probability that the estimate is not equal to the original L-tuple of source vectors. We say that the rates $(R_1, R_2, \ldots, R_L)$ are achievable if for all

$\epsilon > 0$ and k large enough, if there exist deterministic encoding functions $\mathcal{E}_\ell$ and decoding function $\mathcal{D}$ with probability of error no greater than ϵ. The goal is to compress the sources using the lowest possible achievable rates.

Let $S_\mathcal{I}$ denote the subset of $(S_1, S_2, \ldots, S_L)$ with indices in the set $\mathcal{I} \subseteq \{1, 2, \ldots, L\}$. The following result of Slepian and Wolf shows that even though the sources are encoded separated, there is no difference in sum rate from a centralized solution [145].

Theorem 3 (Slepian-Wolf)**.** *A rate tuple is achievable if and only if the following constraints are satisfied:*

$$\sum_{\ell \in \mathcal{I}} R_\ell > H(S_\mathcal{I} | S_{\mathcal{I}^C}) \quad \forall \mathcal{I} \subseteq \{1, 2, \ldots, L\} \tag{2.18}$$

Proof. One elegant proof of this theorem is the random binning argument (first used by Cover in [30]). Each encoder ℓ randomly and independently assigns each possible source sequence s_ℓ^k to an index $w_\ell \in \{1, 2, \ldots, 2^{nR_\ell}\}$ according to a uniform distribution. When it observes S_ℓ^k, it simply transmits the assigned index to the decoder (which is aware of the encoders' codebooks). Given all these indices, the decoder has a list of possible L-tuples of source vectors. It now determines which of these L-tuples is jointly typical. If there is only one, it outputs this as its estimate; otherwise, it declares an error. Thus, there is an error if $(S_1^k, S_2^k, \ldots, S_L^k)$ is not jointly typical or there is more than one jointly typical sequence assigned indices $w_1, \ldots, w_L$. First, recall that by Lemma 1, for k large enough, $(S_1^k, S_2^k, \ldots, S_L^k)$ is jointly typical with probability greater than $1 - \frac{\epsilon}{2}$. Now, consider a jointly typical L-tuple $(\tilde{S}_1^k, \tilde{S}_2^k, \ldots, \tilde{S}_L^k)$ with $\tilde{S}_\ell^k \neq S_\ell^k$ for $\ell \in \mathcal{I}$ and $\tilde{S}_\ell^k = S_\ell^k$ for $\ell \in \mathcal{I}^C$. From [29, Theorem 15.2.2], the number of such sequences is upper bounded by $2^{n(H(S_\mathcal{I}|S_{\mathcal{I}^C})+2\epsilon)}$. The probability that sequence is assigned the same indices is just $2^{-n\sum_{\ell \in \mathcal{I}} R_\ell}$. By the union bound, we see that the probability of error can be driven to zero for k large enough so long as the rate constraints in (2.18) are satisfied. Since this random assignment has a low average probability of error, there must exist at least one good fixed assignment with low probability of error. The converse argument follows easily and we refer the interested reader to [29, Theorem 15.4.1] for more details. □

We note that the Slepian-Wolf problem can also be solved optimally using linear codes as shown by Csiszar [62].

Theorem 4 (Csiszar)**.** *There exist linear codes that can approach the rate region of the Slepian-Wolf problem.*

All we need for the random binning argument is that sequences are assigned to indices in a uniform fashion and that these mappings are pairwise independent. From Lemma 2, a matrix whose entries are drawn i.i.d. uniform over a finite field has exactly these properties.

2.3 Körner-Marton Problem

We now turn to a distributed compression problem for which a random linear coding argument yields the optimal rate region yet a random binning argument does not. Although to date there is no proof that linearity is strictly necessary, this example shows that algebraically structured codes are useful for proving capacity theorems.

There are two encoders which each observe a source vector $S_\ell^k \in \{0,1\}^k$. The sources are generated i.i.d. from the following joint pmf:

$$\begin{aligned} \Pr(S_1 = 0, S_2 = 0) &= \Pr(S_1 = 1, S_2 = 1) = \frac{1-p}{2} \\ \Pr(S_1 = 0, S_2 = 1) &= \Pr(S_1 = 1, S_2 = 0) = \frac{p}{2} \end{aligned} \tag{2.19}$$

A simple calculation will show that S_1 and S_2 have uniform marginal distributions. The goal is to reconstruct the mod-2 sum, $U = S_1 \oplus S_2$, at the decoder with vanishing probability of error (see Figure 2.3).

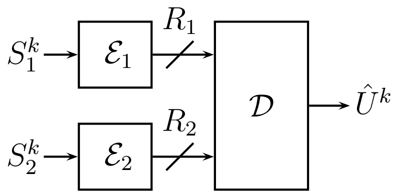

Figure 2.3: Körner-Marton Problem

Each encoder maps its observation S_ℓ^k into nR_ℓ bits:

$$\mathcal{E}_\ell : \{0,1\}^k \rightarrow \{1, 2, \dots, 2^{nR_\ell}\} \quad \ell = 1, 2 \tag{2.20}$$
$$\tag{2.21}$$

and the decoder makes an estimate $\hat{U}^k$ of the mod-2 sum using its decoding function:

$$\mathcal{D} : \{1, 2, \dots, 2^{nR_1}\} \times \{1, 2, \dots, 2^{nR_1}\} \rightarrow \{0,1\}^k \tag{2.22}$$

The probability of error is given by $P_e = Pr(\hat{U}^k \neq U^k)$. We say that the rate pair (R_1, R_2) is acheivable if for all $\epsilon > 0$ and k large enough, if there exist deterministic encoding functions $\mathcal{E}_\ell$ and decoding function $\mathcal{D}$ with probability of error no greater than ϵ. Let $h_B(p)$ be the binary entropy function:

$$h_B(p) = -p \log p - (1-p) \log (1-p) \tag{2.23}$$

and note that $H(U) = h_B(p)$.

Theorem 5 (Körner-Marton)**.** *The rate region for distributed compression of $U = S_1 \oplus S_2$ is given by the following constraints:*

$$R_1 > h_B(p) \tag{2.24}$$

$$R_2 > h_B(p). \tag{2.25}$$

Proof. (*Achievability.*) Choose a linear source code, $\mathbf{G} \in \{0,1\}^{n \times nR}$ with rate $R > h_B(p)$ that is sufficient for losslessly compressing U. Have each encoder apply this code to its observed source vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ to get $\mathbf{w}_1 = \mathbf{s}_1\mathbf{G}$ and $\mathbf{w}_2 = \mathbf{s}_2\mathbf{G}$. These codewords are sent to the decoder which computes $\mathbf{w}_1 \oplus \mathbf{w}_2 = \mathbf{s}_1\mathbf{G} \oplus \mathbf{s}_2\mathbf{G} = \mathbf{u}\mathbf{G}$. Since $\mathbf{G}$ was chosen for recovering U, decoding is successful.

(*Converse.*) Consider the relaxation where the decoder has full knowledge of S_2 and we would like to jointly encode S_1 and U to losslessly reconstruct U at the decoder. Note that any scheme that accomplishes this also gives the decoder a lossless reconstruction of S_1. Thus, it can be shown that for joint encoding, $R \geq H(S_1, U|S_2) = H(U|S_2) = H(U) = h_B(p)$ is required for a vanishing probability of error. This implies that for separate encoding of S_1 and U, $R_1 + R_U \geq h_B(p)$. Similarly, we can get that $R_2 + R_U \geq h_B(p)$. Setting $R_U = 0$ gives the desired result. □

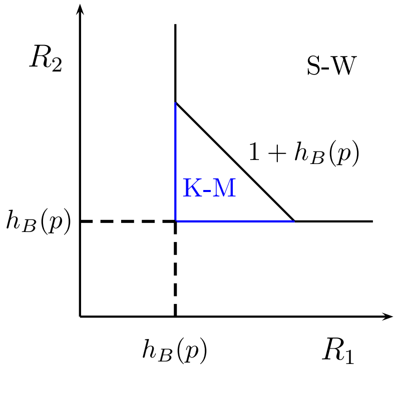

Figure 2.4: Körner-Marton and Slepian-Wolf rate regions for the distributed compression of the parity of two dependent sources.

For random binning with rates satisfying:

$$R_1 > h_B(p) \tag{2.26}$$

$$R_2 > h_B(p) \tag{2.27}$$

$$R_1 + R_2 > 1 + h_B(p) \tag{2.28}$$

it is easy to show that the decoder can reconstruct the source vectors S_1^k and S_2^k and U^k follows by taking the mod-2 sum (see Figure 2.4 for a comparison of the Körner-Marton and Slepian-Wolf rate regions for this problem). We now argue that with random binning the mod-2 sum cannot be recovered at smaller rates. Suppose that $R_1 + R_2 < 1 + h_B(p)$. We can correctly decode the sum with high probability if all typical pairs assigned to a particular pair bin indices have the same mod-2 sum. This ensures that the decoder will not get confused between several possible sums. There are approximately $2^{n(1+h_B(p))}$ typical pairs but there are at most 2^n pairs with the same mod-2 sum (even including atypical sequences). Thus, two typical pairs assigned to the same bin indices only have the same mod-2 sum with vanishing probability. As $R_1 + R_2 < 1 + h_B(p)$, we will definitely have many typical pairs assigned to the same bins and these will almost certainly have different mod-2 sums. As a result, we cannot recover the mod-2 sum correctly at the decoder.

The Körner-Marton problem demonstrates that there exist problems for which purely random coding arguments are insufficient. However, the gains depend on the source dependencies; for independent sources, there is no advantage to linear codes. A similar phenomenon has been discovered for correlated Gaussian sources by Krithivasan and Pradhan [80]. There if the sources are positively correlated and we only demand the difference at the decoder then lattice coding can be helpful. In the next two chapters, we show that for computation over noisy channels, structured codes offer significant benefits even if the sources are independent. Then, in Chapter 5, we show that linear and lattice codes can help prove new achievability results even when we are only interested in sending independent messages across a network.

Chapter 3

Compute-and-Forward: Discrete Alphabets

Computation and communication are often viewed as distinct problems. A communications engineer, tasked to design a multi-user system for performing computations while facing communication constraints, would almost certainly employ a version of the "separation principle." The system would employ a (distributed) source code to compress the sources into bits and a channel code to losslessly convey these bits over the noisy channel. The perceived reason for this design choice is two-fold. First, the abstraction of the sources and channel to bits lends itself to a universal, modular design. Second, it seems that the only gain from a joint source-channel design stems from exploiting the correlations between the sources as in [28].

In this chapter, we study the problem of computing functions over multiple-access channels (MACs) and show that in many cases of interest, a joint design can exploit a match between the structure of the channel and the function to be computed. This *structural gain* does not hinge on the correlations between the sources and, with a perfect matching, increases the *computation rate* proportionally to the number of users. Furthermore, our underlying schemes are modular and depend primarily on coding techniques originally developed for their lower complexity.

Instead of fighting the interference caused by other users, our codes exploit channel collisions to compute functions efficiently. This can be thought of as a form of passive cooperation between transmitting terminals. More precisely, in the standard literature, cooperation is often considered in terms of the correlations (and more generally, dependence) it creates between transmitted signals, thus permitting it to outperform the communication performance attainable without cooperation. It should be clear that correlated signals only result in improved performance if the correlation between the signals is appropriately matched to the *structure* of the multiple-access channel. In our considerations, the goal is no longer to communicate messages, but a function thereof. By using appropriate codes, the transmitters cooperate to realize an enhanced communication performance. Again, it should be clear that this only results in a gain if the desired function is appropriately matched to the *structure* of the multiple-access channel. In this chapter, we provide two strategies for computing functions over multiple-access channels.[1]

Summary of Results

In this chapter, we focus on sources and channels that take values over finite alphabets. Our constructions are based on linear codes over finite fields and, as a result, our schemes work best when the desired function is linear. For non-linear functions, we find a one-to-one map onto a linear function which is transmitted over the channel and mapped back to the desired non-linear function.

First, we will bound the performance of separation-based schemes in Section 3.2. For many functions, if the sources are independent, then the best a separation-based scheme can do is have each encoder send its source in its entirety.

Next, in Section 3.3, we propose a strategy that we call *linear computation coding*. Essentially, each encoder employs the same linear code. If the channel is a noisy linear function of its inputs, then the receiver will observe the codeword for the desired linear function corrupted by noise and this scheme is optimal. More generally, so long as the mutual information between a linear function of the channel inputs and the output is non-zero, we can compute at a non-zero rate.

We take a different approach in Section 3.4 which is useful when the channel is not a linear combination of its inputs. First, the encoders transmit their sources in an uncoded fashion and the receiver collects this as side information. Then, using a separation-based scheme, the encoders send a few extra bits to the decoder to help it decode the desired function. This scheme is called *systematic computation coding*.

Finally, in Section 3.A we give upper bounds on the computation capacity.

Related Work

Shannon showed in his landmark paper that separate source and channel code design is asymptotically optimal in a point-to-point setting [142, Theorem 21]. This insight has fueled a design philosophy based completely on bits. Although in many cases of interest, such an approach is optimal, it is well-known that in certain scenarios separation fails. For instance, Cover, El Gamal, and Salehi demonstrated that separation is suboptimal for transmitting correlated sources over a MAC in [28]. Their joint source-channel scheme uses the source correlations to create channel input probability distributions unavailable to a separation-based scheme. Exploiting the source correlations in this fashion is sometimes known as

[1] The material in this chapter is drawn from [104].

collaborative gain. Ahlswede and Han continued work on the problem of sending correlated sources over a MAC in [5]. In particular, they considered a variant of the problem in which only one of the sources had to be recovered.

In [54; 52], an uncoded joint source-channel scheme is shown to be optimal (and significantly better than separation) for estimating a remote source from multiple observations. Although at a first glance, the scheme seems to benefit only from the correlations between the observations, it also exploits an ideal structural match between the channel, a Gaussian MAC, and the sufficient statistic, the sum of the observations. This uncoded transmission framework has been extended to more general sensor network estimation problems in [93; 12].

In [56], function properties are used to reduce the amount of required communication in a large sensor network. For many functions, the sensors can process incoming data before sending it along to the fusion center, thus reducing the communications overhead. Related work has developed upper and lower bounds on the network computation problem wherein multiple source nodes communicate over a graph of bit pipes to a single receiver that wants a function of the sources [9].

Reliable distributed computation has been studied from the source coding perspective as well. The general problem is still open and seems prohibitively difficult with current techniques. Körner and Marton found the rate region for distributed compression of the parity of two correlated uniform binary sources in [77]. Their proof relies on random *linear* codes and their gains come entirely from the correlation between the sources. The seeming necessity of linear codes for this simple problem implies that random coding techniques are inadequate for the general problem. Doshi et al. showed that in special cases where a "zig-zag" condition is satisfied, graph coloring combined with Slepian-Wolf coding is sufficient [37].

In [117], Orlitsky and Roche determined the required rate for sending X to a decoder with side information Y that must reliably compute $f(X, Y)$. This is essentially a generalization of the Körner-Marton parity problem to any function except that the decoder gets Y for free. The basic result is that in most cases of interest, we must send X in its entirety to the decoder; further compression is only possible if for some x and x', $f(x, Y) = f(x', Y)$ with probability 1. In many cases, the gains enabled by requiring only a function of the sources at the decoder versus the sources themselves are marginal.

Earlier work by Yamamato established the rate-distortion function for sending X to a decoder that must reconstruct $f(X, Y)$ up to a given fidelity given Y as side-information [159]. In [45], the authors extend the rate-distortion function to the case where only a noisy version of X is available at the encoder.

Ma and Ishwar solved an interactive distributed source coding problem related to the side information problems above [89]. One encoder observes X and another observes Y. These encoders exchange messages in a multi-round protocol in an attempt to compute $f(X, Y)$ to a given fidelity. Further results appear in [90].

Krithivasan and Pradhan have developed a framework for distributed source coding using nested group codes [82]. In their considerations, the decoder is interested in a general function of the sources and they are able to include many known source coding results as special cases.

3.1 Problem Statement

We begin with several transmitters that wish to communicate a function to a single receiver across a multiple-access channel (see Figure 3.1). The goal is to reliably reconstruct the function at the receiver at the highest possible rate.

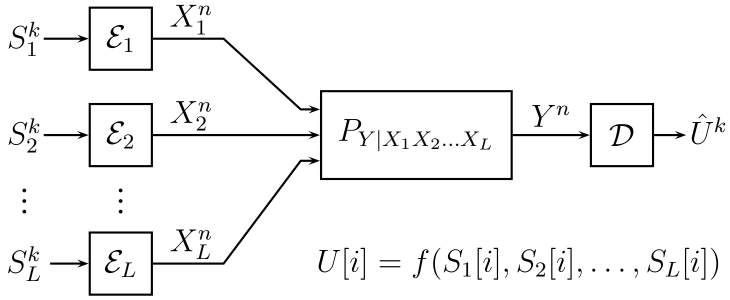

Figure 3.1: Reliable Computation over a MAC. The decoder only reconstructs a function of the sources.

Remark 1. We assume that time is discrete. This can be justified by the well-known fact that any continuous-time system with finite bandwidth can be reduced to a discrete-time system using the sampling theorem [113; 78; 143]. In nearly any practical setting, finite bandwidth is assured.

Definition 7 (Sources)**.** Let $(S_1^k, S_2^k, \ldots, S_L^k)$ be an L-tuple of length k source vectors drawn i.i.d. according to $p_{S_1S_2\cdots S_L}(s_1, s_2, \ldots, s_L)$. Each source S_ℓ takes values in the finite alphabet $\mathcal{S}_\ell$.

Definition 8 (Desired Function)**.** Let $\mathcal{U}$ be a finite alphabet and f the *desired function* of the sources:

$$f : \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \mathcal{S}_L \to \mathcal{U}. \tag{3.1}$$

In some cases, we may want the receiver to recover several functions $f_1, \ldots, f_N$ of the sources in alphabets $\mathcal{U}_1, \ldots, \mathcal{U}_N$, respectively.

Definition 9 (MAC)**.** The *multiple-access channel* is specified by a conditional pmf

$$p_{Y|X_1X_2\cdots X_L}(y|x_1,x_2,\ldots,x_L) \tag{3.2}$$

with channel inputs $x_\ell \in \mathcal{X}_\ell$ and channel output $y \in \mathcal{Y}$.

Definition 10 (Computation Code)**.** A (k,n,ϵ) *computation code* is specified by M *encoders*:

$$\mathcal{E}_\ell : \mathcal{S}_\ell^k \to \mathcal{X}_\ell^n, \tag{3.3}$$

for $\ell = 1,2,\ldots,L$, as well as a *decoder*:

$$\mathcal{D} : \mathcal{Y}^n \to \mathcal{U}^k, \tag{3.4}$$

such that:

$$\begin{aligned} X_\ell^n &= \mathcal{E}_\ell(S_\ell^k) \\ \hat{U}^k &= \mathcal{D}(Y^n) \\ \Pr(\hat{U}^k &\neq U^k) \leq \epsilon. \end{aligned} \tag{3.5}$$

Definition 11 (Computation Rate)**.** We say a *computation rate*, R_{COMP}, is achievable if $\forall\epsilon \in (0,1)$ and n large enough there exists a (k,n,ϵ) code satisfying:

$$\frac{k}{n} \geq R_{\text{COMP}}. \tag{3.6}$$

Definition 12 (Computation Capacity)**.** The *computation capacity*, C_{COMP}, is the supremum of all achievable computation rates.

3.2 Separation-Based Computation

If we want to reliably send a single source over a noisy channel, then it is known that compressing the source into bits and communicating these reliably over the channel is optimal [142]. This result, due to Shannon, is known as the separation theorem. The separation-based approach generalizes naturally to a network setting. Essentially, each transmitter compresses its source into bits and a channel code communicates these bits to the receivers which then attempt to recover the sources (or functions thereof). Although in many special cases, such an approach is optimal, it is well-known that in certain networks separation fails. For instance, Cover, El Gamal, and Salehi demonstrated that separation is suboptimal for transmitting dependent sources over a MAC in [28]. Their joint source-channel scheme uses the source dependencies to create channel input probability distributions unavailable to a

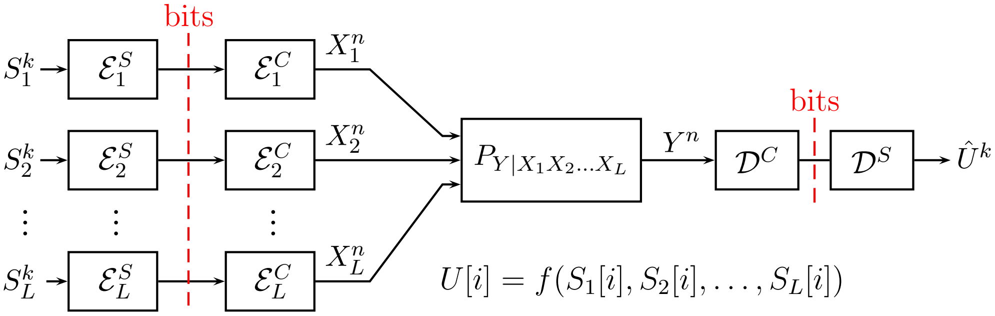

Figure 3.2: Separation-based computation over a multiple-access channel.

separation-based scheme. Our results show that the separation-based approach is suboptimal for sending functions over a channel, even if the sources are independent.

In this section, we formally define what we mean by a separation-based scheme for computation over a MAC (see Figure 3.2). This will give us a baseline against which we can evaluate the performance of our computation codes.

Definition 13. The *distributed compression rate region*, $\mathbf{R}_f$, is the set of all rate vectors $(R_1, R_2, \dots, R_L)$ such that for all $\epsilon > 0$ and k large enough there are L source encoders and a source decoder of the form:

$$\mathcal{E}_\ell^S : \mathcal{S}_\ell^k \rightarrow \{0,1\}^{kR_\ell} \tag{3.7}$$
$$\mathcal{D}^S : \{0,1\}^{kR_1} \times \dots \times \{0,1\}^{kR_L} \rightarrow \mathcal{U}^k, \tag{3.8}$$

for $\ell = 1, 2, \dots, L$ such that the desired function $U = f(S_1, S_2, \dots, S_L)$ can be recovered with probability of error at most ϵ:

$$\hat{U}^k = \mathcal{D}^S(\mathcal{E}_1^S(S_1^k), \dots, \mathcal{E}_L^S(S_L^k))$$
$$\Pr(\hat{U}^k \neq U^k) < \epsilon. \tag{3.9}$$

Unfortunately, as of the writing of this thesis, the distributed compression problem remains unsolved. Körner and Marton solved the special case where there are two correlated, uniform, binary sources and we want to recover their parity [77] (see Section 2.3 for more details). Recall also that the rate region for complete recovery of the sources was characterized by Slepian and Wolf in [145] (see Section 2.2). Orlitsky and Roche solved the special case where all but one of the sources are given to the decoder as side information [117]. The required rate is given by a graph entropy characterization and is reviewed in detail in Section 3.A.1. We will use their result to establish the distributed compression rate region for a restricted class of functions with independent sources as inputs. Essentially, if no input symbols can be merged without incurring errors and the sources are independent, then the sources must be sent in their entirety.

Lemma 3. *Assume that the sources are independent and the desired function, f, is chosen such that for each pair of possible source symbols at an encoder, $s_\ell, s_\ell^* \in \mathcal{S}_\ell$, there is a choice of $s_1, s_2, \ldots, s_{\ell-1}, s_{\ell+1}, \ldots, s_L$ such that:*

$$Pr(f(s_1, \ldots, s_\ell, \ldots, s_L) \neq f(s_1, \ldots, s_\ell^*, \ldots, s_L)) > 0.$$

Then, the rate required for each decoder for distributed compression of f is $R_\ell \geq H(S_\ell)$.

See Appendix 3.A.1 for a proof.

Example 1. Let $S_1, S_2, \ldots, S_L$ be independent sources drawn uniformly from the same alphabet. Then, real addition, $U_1 = \sum_{\ell=1}^{L} S_\ell$, and multiplication, $U_2 = S_1 \cdot S_2 \cdot \ldots \cdot S_L$, satisfy the conditions of Lemma 3.

After the sources are compressed into bits, then these bits must be reliably communicated over the multiple-access channel. Ahlswede and Liao concurrently determined the rate region for the MAC [3; 86].

Definition 14. Let each encoder's message w_ℓ be independently and uniformly drawn from $\{0,1\}^{nR_\ell}$. The *multiple-access rate region*, $\mathbf{R}_{\text{MAC}}$, is the set of all rate vectors $(R_1, R_2, \ldots, R_L)$ such that for all $\epsilon > 0$ and n large enough there are L channel encoders and a channel decoder of the form:

$$\mathcal{E}_\ell^C : \{0,1\}^{nR_\ell} \rightarrow \mathcal{X}_\ell^n \tag{3.10}$$
$$\mathcal{D}^C : \mathcal{Y}^n \rightarrow \{0,1\}^{nR_1} \times \cdots \times \{0,1\}^{nR_L}, \tag{3.11}$$

for $\ell = 1, 2, \ldots, L$ such that all bits can be recovered with probability of error at most ϵ:

$$(\hat{w}_1, \ldots, \hat{w}_L) = \mathcal{D}^{\mathcal{C}}(Y^n)$$
$$\Pr\left((\hat{w}_1, \ldots, \hat{w}_L) \neq (w_1, \ldots, w_L)\right) < \epsilon. \tag{3.12}$$

Theorem 6 (Ahlswede-Liao)**.** *The multiple-access rate region, $\mathbf{R}_{MAC}$, is the closure of the convex hull of the set of all rate vectors, $(R_1, R_2, \ldots, R_L)$, satisfying:*

$$\sum_{\ell \in \mathcal{I}} R_\ell \leq I(X_{\mathcal{I}}; Y | X_{\mathcal{I}^C}) \quad \forall \mathcal{I} \subseteq \{1, 2, \ldots, L\}, \tag{3.13}$$

for some product distribution $p(x_1, x_2, \ldots, x_L) = \prod_{\ell=1}^{L} p(x_\ell)$ where $X_{\mathcal{I}} = \{X_\ell : \ell \in \mathcal{I}\}$.

See [29, Theorem 15.3.1] for a proof.

It is often easier to deal with the MAC rate region in terms of its maximum sum rate.

Definition 15. The *maximum sum rate* of a MAC is:

$$C_{\text{MAC}} = \max_{(R_1,R_2,\ldots,R_L)\in\mathbf{R}_{\text{MAC}}} \sum_{\ell=1}^{L} R_\ell. \tag{3.14}$$

Definition 16. We say that the maximum sum rate of a MAC is *symmetric* if $R_1^* = R_2^* = \cdots = R_L^*$ where

$$(R_1^*, R_2^*, \ldots, R_L^*) \in \operatorname*{arg\,max}_{(R_1,R_2,\ldots,R_L)\in\mathbf{R}_{\text{MAC}}} \sum_{\ell=1}^{L} R_\ell. \tag{3.15}$$

We can now formally define what we mean by separation-based computation.

Definition 17. A computation rate R_{COMP} is *achievable with separation* if:

$$\begin{aligned} &\mathcal{A} = \left\{ \left(\frac{R_1}{R_{\text{COMP}}}, \ldots, \frac{R_L}{R_{\text{COMP}}} \right) : (R_1, \ldots, R_L) \in \mathbf{R}_{\text{MAC}} \right\} \\ &\mathbf{R}_f \cap \mathcal{A} \neq \emptyset. \end{aligned} \tag{3.16}$$

As shown in [28], when we want to send dependent sources over a MAC, separation is not optimal. Clearly, if we allow our sources to be dependent but only require a function of these sources at the decoder, a separation-based scheme may not be optimal for the same reasons. However, even if we assume that the sources are independent, we still do not get a separation theorem as shown in the following example, taken from Problem 1.1 in [51].

Example 2. Let S_1 and S_2 be independent $\mathcal{B}(\frac{1}{2})$ sources. Each source is seen by a separate encoder with access to one terminal of a MAC. The MAC input alphabets are $\mathcal{X}_1 = \mathcal{X}_2 = \{0, 1\}$ and the output is $Y = X_1 \oplus X_2$. The maximum sum rate of this MAC is clearly $C_{\text{MAC}} = 1$. At the decoder, we would like to losslessly compute $U = S_1 \oplus S_2$. Using Lemma 3 and the data processing inequality, it can be shown that the best separation-based scheme achieves a computation rate of $R_{\text{COMP}} = \frac{1}{2}$. The separation-based scheme just amounts to using two channel uses, one to transmit each source. However, if we set $X_\ell = S_\ell$ then the channel computes the function directly and we can achieve the computation capacity $C_{\text{COMP}} = 1$ (see Lemma 6 for the converse).

As the example demonstrates, sometimes we can compute the desired function using the channel. In these cases, joint source-channel schemes can achieve a much higher computation rate than separation-based schemes, sometimes a factor of L higher. Of course, the above example is somewhat contrived, as the channel performs exactly the operation we desire and there is no noise. Our results show that using the channel's natural operation to compute a function can give us boosts over separation-based schemes, even when the channel is noisy.

3.3 Linear Computation Codes

In this section, we develop computation coding strategies that are well-suited for communicating linear functions. Each transmitter maps its observed source vector onto a codeword drawn from the same linear codebook. The receiver then attempts to directly decode the desired linear function from the channel output. In the special case where the channel itself is a noisy linear function, this attains the computation capacity.

Let $\mathbb{F}$ denote a finite field and let $\oplus$ and $\bigoplus$ denote addition and summation over the field, respectively. (We reserve $+$ and $\sum$ for addition and summation over the complex field $\mathbb{C}$.)

Definition 18 (Linear Functions)**.** Assume the sources take values on a finite field, $\mathcal{S}_\ell = \mathbb{F}$. We say the desired function U is *linear with respect to* $\mathbb{F}$ if it can be written as:

$$U[i] = \bigoplus_{\ell=1}^{L} \alpha_\ell S_\ell[i] \tag{3.17}$$

for some $\alpha_\ell \in \mathbb{F}$.

Theorem 7. *Assume that $\mathcal{S}_\ell = \mathbb{F}$, $|\mathcal{X}_\ell| \leq |\mathbb{F}|$, and the desired function U is linear with respect to $\mathbb{F}$. Choose any one-to-one (injective) functions from the finite field onto the input alphabets, $c_\ell : \mathbb{F} \rightarrow \mathcal{X}_\ell$. The following computation rate is achievable:*

$$R_{COMP} = \frac{I\left(\bigoplus_{\ell=1}^{L} c_\ell^{-1}(X_\ell); Y\right)}{H(U)} \tag{3.18}$$

for X_ℓ drawn independently and uniformly over the range of c_ℓ.

Proof. Let $\mathbf{s}_\ell = S_\ell^k$, $\mathbf{u} = U_\ell^k$, and choose $\epsilon > 0$. Using Theorem 4, choose a matrix $\mathbf{B}$ of size $k \times m$ over $\mathbb{F}$ for compressing U. If $m = k\left(\frac{H(U)+\epsilon}{\log_2|\mathbb{F}|}\right)$, then with probability greater than $1-\epsilon$, $\mathbf{u}$ can be recovered from $\mathbf{w}_U = \mathbf{u}\mathbf{B}$. Each encoder applies $\mathbf{B}$ to its source to get its message $\mathbf{w}_\ell = \alpha_\ell \mathbf{s}_\ell \mathbf{B}$.

Now, randomly generate a channel coding matrix $\mathbf{G}$ of size $m \times n$ with each element chosen independently and uniformly over $\mathbb{F}$. Each encoder multiplies its message by $\mathbf{G}$ to get $\tilde{\mathbf{x}}_\ell = \mathbf{w}_\ell \mathbf{G}$ and then applies c_ℓ symbolwise to get the sequence of channel inputs $\mathbf{x}_\ell$:

$$\mathbf{x}_\ell = c_\ell(\tilde{\mathbf{x}}_\ell) = \Big[c_\ell(\tilde{X}_\ell[1]) \; c_\ell(\tilde{X}_\ell[2]) \; \cdots \; c_\ell(\tilde{X}_\ell[n]) \Big] \tag{3.19}$$

Note that for all possible choices of source vectors $\mathbf{s}_1^*, \ldots, \mathbf{s}_L^* \in \mathbb{F}^k$ such that $\mathbf{u} = \bigoplus_{\ell=1}^L \alpha_\ell \mathbf{s}_\ell^*$, the corresponding codewords $\mathbf{x}_\ell^* = c_\ell(\alpha_\ell \mathbf{s}_\ell^* \mathbf{B}\mathbf{G})$ satisfy:

$$\bigoplus_{\ell=1}^L c_\ell^{-1}(\mathbf{x}_\ell^*) = \bigoplus_{\ell=1}^L c_\ell^{-1}(\mathbf{x}_\ell) = \bigoplus_{\ell=1}^L \alpha_\ell \mathbf{s}_\ell \mathbf{B}\mathbf{G} = \bigoplus_{\ell=1}^L \mathbf{w}_U \mathbf{G}. \tag{3.20}$$

Thus, we can treat the problem as if $\mathbf{w}_U$ was directly encoded into a "channel input" $\mathbf{x}_U = \bigoplus_{\ell=1}^L c_\ell^{-1}(\mathbf{x}_\ell)$ and observed at the receiver as $\mathbf{y}$. We can solve for $p_{Y|\bigoplus_{\ell=1}^L c_\ell^{-1}(X_\ell)}$ from $p_{Y|X_1 \cdots X_L}$ and then proceed as we would in a point-to-point channel coding problem. Note that there are The receiver looks for a sequence $\hat{\mathbf{x}}_U$ that is jointly typical with its received sequence $\mathbf{y}$. There is an error if $(\mathbf{x}_U, \mathbf{y})$ is not jointly typical or there is another typical $\mathbf{x}_U^* \neq \mathbf{x}_U$ such that $(\mathbf{x}_U^*, \mathbf{y})$ is jointly typical. Using Lemma 1, we have that for n large enough, the probability that $(\mathbf{x}_U, \mathbf{y})$ is not jointly typical is upper bounded by $\frac{\epsilon}{2}$. Since the codewords of $\mathbf{G}$ are pairwise independent, then it follows that:

$$Pr\left((\mathbf{x}_U^*, \mathbf{y}) \in A_\epsilon^{(n)} \right) \leq 2^{-n\left(I\left(\bigoplus_{\ell=1}^L c_\ell^{-1}(X_\ell); Y \right) - 3\epsilon \right)} \tag{3.21}$$

By the union bound, the probability of error, averaged over $\mathbf{G}$, is upper bounded by

$$\begin{aligned} P_e &< \frac{\epsilon}{2} + \sum_{\mathbf{x}_U^* \neq \mathbf{x}_U} 2^{-n\left(I\left(\bigoplus_{\ell=1}^L c_\ell^{-1}(X_\ell); Y \right) - 3\epsilon \right)} && (3.22) \\ &< \frac{\epsilon}{2} + |\mathbb{F}|^m 2^{-n\left(I\left(\bigoplus_{\ell=1}^L c_\ell^{-1}(X_\ell); Y \right) - 3\epsilon \right)} && (3.23) \\ &= \frac{\epsilon}{2} + 2^{m \log_2 |\mathbb{F}| - n\left(I\left(\bigoplus_{\ell=1}^L c_\ell^{-1}(X_\ell); Y \right) - 3\epsilon \right)} && (3.24) \end{aligned}$$

For n large enough and $m \log_2 |\mathbb{F}| < n\left(I\left(\bigoplus_{\ell=1}^L c_\ell^{-1}(X_\ell); Y \right) - 3\epsilon \right)$, we can drive the probability of error below ϵ. Substituting $m = k\left(\frac{H(U)+\epsilon}{\log_2 |\mathbb{F}|} \right)$, we get the following requirement

$$\frac{k}{n} < \frac{I\left(\bigoplus_{\ell=1}^L c_\ell^{-1}(X_\ell); Y \right) - 3\epsilon}{H(U) + \epsilon}. \tag{3.25}$$

By choosing ϵ small enough, we can achieve the computation rate in the theorem statement. Finally, we can argue that there must exist at least one good fixed $\mathbf{G}$ since the average performance was good. □

In general, the achievable rate in Theorem 7 is not optimal as shown in the following example.

Example 3. Let S_1 and S_2 be i.i.d. $\mathcal{B}(\frac{1}{2})^2$ Assume we want $U = S_1 \oplus S_2$ and the channel is the binary product of binary inputs, $Y = X_1 \odot X_2$. It can be shown that for all possible c_ℓ, $I(c_1^{-1}(X_1) \oplus c_2^{-1}(X_2); X_1 \odot X_2) = 1 - \frac{3}{4}h_B(\frac{1}{3}) \approx 0.311$. Since $H(U) = 1$, the computation rate is $R_{\text{COMP}} \approx 0.311$. By Lemma 3, the best separation-based scheme requires sending both sources and can achieve $R_{\text{COMP}} = \frac{1}{2}$.

3.3.1 Linear Multiple-Access Channels

We now describe a special class of multiple-access channels for which Theorem 7 yields the computation capacity for sending linear functions.

Definition 19. A multiple-access channel is *linear with respect to* $\mathbb{F}$ if its channel inputs take values on a Galois field $\mathbb{F}$ and we can represent the channel output $Y[i]$ as coming from a symmetric discrete memoryless channel (DMC), $p_{Y|V}$, where:

$$V[i] = \sum_{\ell=1}^{L} \beta_\ell X_\ell[i] \tag{3.26}$$

for some $\beta_\ell \in \mathbb{F} \setminus \{0\}$ where 0 is the zero symbol in $\mathbb{F}$. See Figure 3.3.

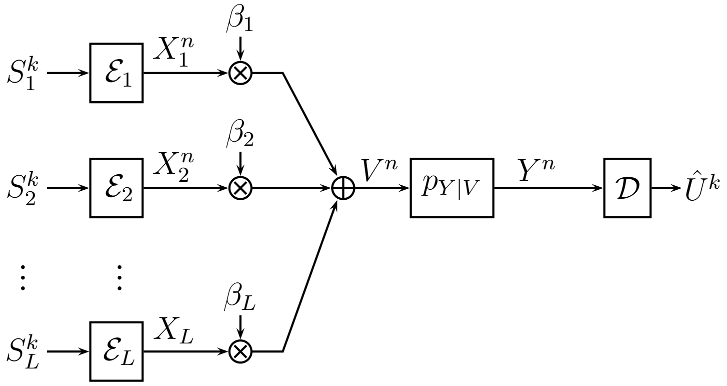

Figure 3.3: Linear Multiple-Access Channel

Note that we require the DMC to be symmetric so that the capacity-achieving input distribution is uniform.

[2] $\mathcal{B}(p)$ represents the Bernoulli distribution over $\{0, 1\}$ where p represents the probability of drawing 1.

Theorem 8. *If both the multiple-access channel and the desired function are linear with respect to $\mathbb{F}$, the computation capacity is:*

$$C_{COMP} = \frac{I(V;Y)}{H(U)} \tag{3.27}$$

where V is uniform over $\mathbb{F}$.

Proof. The achievability is given by Theorem 7 by choosing $c_\ell(X_\ell) = \beta_\ell^{-1} X_\ell$. It follows that

$$I\left(\sum_{\ell=1}^{L} c_\ell^{-1}(X_\ell); Y\right) = I\left(\sum_{\ell=1}^{L} \beta_\ell X_\ell; Y\right) = I(V;Y). \tag{3.28}$$

The converse is given by Lemma 6. □

This performance is not available to a separation-based scheme. If the sources are independent, we can invoke Lemma 3 to show that each encoder must communicate its source to the decoder. This requires a sum-rate of $\sum_{\ell=1}^{L} H(S_\ell)$ which means that the best separation-based computation rate is:

$$R_{\text{COMP}} = \frac{I(V;Y)}{\sum_{\ell=1}^{L} H(S_\ell)} \tag{3.29}$$

for V generated uniformly from $\mathbb{F}$. Clearly, $H(U) \leq \sum_{\ell=1}^{L} H(S_\ell)$ so this rate is lower than the computation capacity.

3.3.1.1 Extended Example: Mod-2 Adder MAC

We now explore an example that demonstrates the benefits of computation coding over separation-based schemes. Our example centers on the mod-2 adder MAC (M2MAC) (Figure 3.4). There are two sources, S_1 and S_2, generated from the following joint pdf:

$$\begin{aligned} \Pr(S_1 = 0, S_2 = 0) &= \Pr(S_1 = 1, S_2 = 1) = \frac{1-p}{2} \\ \Pr(S_1 = 0, S_2 = 1) &= \Pr(S_1 = 1, S_2 = 0) = \frac{p}{2}. \end{aligned} \tag{3.30}$$

A simple calculation will show that S_1 and S_2 have uniform marginal distributions. Our goal is to losslessly transmit $U = S_1 \oplus S_2$ across the channel at the highest computation rate $\kappa = \frac{k}{n}$. The entropy of U is given by the binary entropy function:

$$h_B(p) = -p \log p - (1-p) \log(1-p). \tag{3.31}$$

The channel input and output alphabets are given by $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{Y} = \{0, 1\}$. The channel inputs are added mod-2 to yield $V = X_1 \oplus X_2$ which is passed through a binary symmetric channel (BSC) with crossover probability q to give Y (see Figure 3.4).

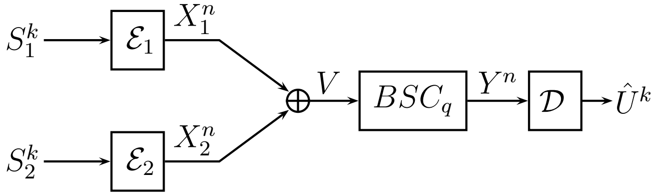

Figure 3.4: Mod-2 Adder Multiple-Access Channel (M2MAC)

Our sources and desired function are identical to those from the Körner-Marton problem [77] (see Section 2.3). By combining the Körner-Marton source coding scheme with an appropriate MAC code, we will get the optimal separation-based scheme.

For the M2MAC, the capacity region has only a single constraint:

$$R_1 + R_2 < 1 - h_B(q). \tag{3.32}$$

Note that this implies that time-sharing is optimal for the M2MAC.

We can now give the best possible computation rate available using separation. The sum source coding rate required is $2h_B(p)$ and the MAC sum capacity is $1 - h_B(q)$. Reliable communication requires that $k(2h_B(p)) < n(1 - h_B(q))$. This gives the optimal separation-based computation rate of:

$$R_{\text{COMP}} = \frac{1}{2}\left(\frac{1 - h_B(q)}{h_B(p)}\right). \tag{3.33}$$

Remark 2. The Körner-Marton scheme allows for a strictly lower sum source coding rate and thus, a higher computation rate than Slepian-Wolf coding of S_1 and S_2.

The best separation-based scheme for the M2MAC uses structured source coding to exploit the source correlations. The channel coding strategy focuses on avoiding the interference caused by the other user. Yet, the interference is due to the summation taken by the MAC. Computation coding exploits this summation by using both a structured source code and a structured channel code. In doing so, it can optimally exploit both the source correlations and the structure of the MAC. An application of Theorem 8 to this scenario yields the following corollary.

Corollary 1. *The computation capacity for sending $U = S_1 \oplus S_2$ over the M2MAC is*

$$C_{COMP} = \frac{1 - h_B(q)}{h_B(p)}. \tag{3.34}$$

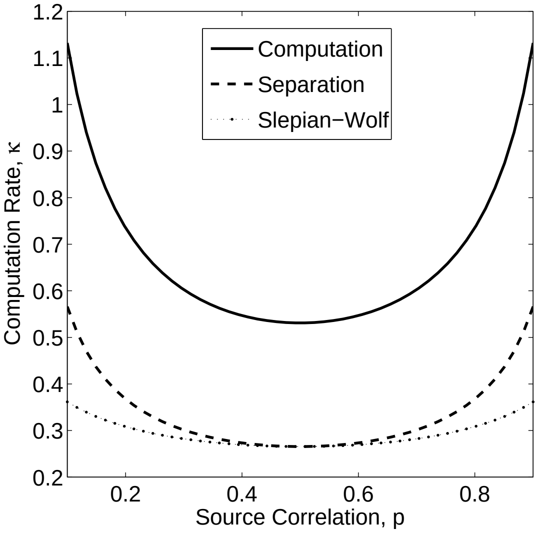

Figure 3.5: Comparison of schemes for computing parity over a noisy modulo-2 adder.

Somewhat surprisingly, this strategy allows for a computation rate twice that of the separation scheme, regardless of the source statistics. The computation rates for computation coding (Corollary 1), the best separation-based scheme (3.33), and a suboptimal separation-based scheme that uses Slepian-Wolf source coding over an M2MAC with crossover probability $q = 0.1$ are shown in Figure 3.5.

Both our computation coding scheme and the best separation-based scheme take advantage of the structure of the function for source coding. The computation coding scheme goes one step further and takes advantage of structure of channel. The computation rate is doubled by this structural gain. This shows that the MAC rate region is an insufficient characterization of the channel for distributed computation.

The symmetric source pdf (see (3.30)) used for the M2MAC example can be changed to any joint pdf and the computation capacity will still be achieved by the scheme put forth in Corollary 1. However, for an asymmetric pdf, the Körner-Marton scheme may not be the best separation-based strategy. It is only known to be optimal for the symmetric pdf in (3.30), as this is the most general pdf that results in uniform marginal pdfs. Ahlswede and Han showed that if the marginals are not uniform, there are achievable points outside the Körner-Marton region [5].

To be more specific, both the Körner-Marton scheme and computation coding calibrate their codes using the entropy of the desired function. Körner-Marton fails as a general solution as it then converts the linear representation into bits, which destroys the code's match with the function. The function-channel match in computation coding allows for a continuous abstraction of the problem in terms of the underlying finite field. This is why we are able to meet our upper bounds in matched cases.

The most interesting aspect of our strategy is that it depends entirely on codes that were originally intended to reduce system complexity. Elias' random linear coding proof was meant to show that the search for implementable codes is not futile; all of the benefits of Shannon's random codebooks can be transferred into random generator matrices [39]. The Körner-Marton result and our computation code show that structured codes can enable rate gains. In particular, structured codes allow redundancy to be added in a distributed, yet structured, fashion.

3.3.2 Extensions

We now describe a few simple extensions to Theorem 7. In some cases, we may be interested in a non-linear function f of the sources. One approach is to find a one-to-one (injective) map from f into some linear function g and then attempt to communicate g directly using Theorem 7. We may also want to communicate more than one (linear) function to the receivers. These linear functions may not be independent and we can take advantage of this to increase the computation rate.

Theorem 9. *Assume that $\mathcal{S}_\ell = \mathbb{F}$, $|\mathcal{X}_\ell| \leq |\mathbb{F}|$, and the desired functions $U_1, \dots, U_J$ are linear with respect to $\mathbb{F}$. Choose any one-to-one (injective) functions from the finite field onto the input alphabets, $c_\ell : \mathbb{F} \rightarrow \mathcal{X}_\ell$. The following computation rate is achievable:*

$$R_{COMP} = \frac{I\left(\bigoplus_{\ell=1}^{L} c_\ell^{-1}(X_\ell); Y\right)}{H(U_1, \dots, U_J)} \tag{3.35}$$

for X_ℓ drawn independently and uniformly over the range of c_ℓ.

Proof. Let $\mathbf{s}_\ell = S_\ell^k$, $\mathbf{u}_j = U_j^k$, and choose $\epsilon > 0$. Using Theorem 4, choose matrices $\mathbf{B}_1, \dots, \mathbf{B}_J$ of size $k \times m_j$ over $\mathbb{F}$ for Slepian-Wolf compression of $U_1, \dots, U_J$. If $m_j = k\left(\frac{H(U_j|U_{j-1},\dots,U_1)+\epsilon}{\log_2 |\mathbb{F}|}\right)$, then with probability than $1-\epsilon$, $\mathbf{u}_1, \dots, \mathbf{u}_J$ can be recovered from $\mathbf{w}_U = \mathbf{u}\mathbf{B}$ where $\mathbf{B} = \begin{bmatrix}\mathbf{B}_1 & \cdots & \mathbf{B}_J\end{bmatrix}$. Note that $\mathbf{B}$ is of size $k \times m$ with $m = \sum_j m_j = k\left(\frac{H(U_1,\dots,U_J)+J\epsilon}{\log_2 |\mathbb{F}|}\right)$. Each encoder applies $\mathbf{B}$ to its source to get its message $\mathbf{w}_\ell = \alpha_\ell \mathbf{s}_\ell \mathbf{B}$. The result follows by repeating the remaining steps from the proof of Theorem 7. □

This immediately yields the computation capacity for sending multiple linear functions over a linear MAC.

Theorem 10. *If both the multiple-access channel and the desired functions are linear with respect to $\mathbb{F}$, the computation capacity is:*

$$C_{COMP} = \frac{I(V;Y)}{H(U_1,\dots,U_J)} \tag{3.36}$$

where V is uniform over $\mathbb{F}$.

Consider the case where there is more than one receiver that wants to reconstruct the same function U. The channel between the L transmitters and M receivers is $p_{Y_1 \cdots Y_M | X_1 \cdots X_L}(y_1,\dots,y_M|x_1,\dots,x_L)$. Theorem 7 can easily be adapted by considering the worst channel over all receivers. This yields the following computation rate:

$$R_{\text{COMP}} = \min_{m=1,\dots,M} \frac{I\left(\bigoplus_{\ell=1}^{L} c_\ell^{-1}(X_\ell);Y_m\right)}{H(U)} \tag{3.37}$$

for some choice of injective functions $c_\ell : \mathbb{F} \rightarrow \mathcal{X}_\ell$ and with X_ℓ independently generated according to the uniform distribution over the range of c_ℓ. However, in many cases, this strategy has a far lower rate than possible with a single receiver. For instance, assume that each receiver sees the output of a linear MAC over $\mathbb{F}$:

$$Y_m[i] = \sum_{\ell=1}^{L} \beta_{m\ell} X_\ell[i] + Z_m[i] \tag{3.38}$$

where $\beta_{m\ell} \in \mathbb{F}$ and the $Z_m[i]$ are all i.i.d. according to the same distribution $p_Z(z)$ on $\mathbb{F}$. If $\beta_{m\ell} = \beta_{m^*\ell}$ for all $m \neq m^*$ then Theorem 8 can be applied directly to get a computation rate of

$$R_{\text{COMP}} = \frac{\log_2 |\mathbb{F}| - H(Z)}{H(U)}. \tag{3.39}$$

This works because the code first cancels out the $\beta_{m\ell}$ and then applies α_ℓ. However, if the channel coefficients differ from receiver to receiver then this cancellation is no longer possible in general. In a certain sense, this can be interpreted to mean that the channel is not well-matched to the desired computation. Therefore, by changing our objective to match the natural computation provided by the channel, much higher rates will be possible. In the following theorem, we take exactly this approach so that the receivers choose their desired function based on the channel coefficients.

Theorem 11. *There are M transmitters whose sources S_ℓ^k are i.i.d. according to the uniform distribution over $\mathbb{F}$. Each receiver observes a noisy linear combination of the transmitted signals X_ℓ^k according to (3.38). If each receivers want a linear function of the sources with* *coefficients chosen according to the channel coefficients,* $U_m[i] = \sum_{\ell=1}^{L} \beta_{m\ell} S_\ell[i]$*, the computation capacity is:*

$$C_{COMP} = \frac{\log_2 |\mathbb{F}| - H(Z)}{\log_2 |\mathbb{F}|}. \tag{3.40}$$

Proof. We essentially follow the proof of 7 except that we set $\mathbf{w}_\ell = \mathbf{s}_\ell$ and take all functions c_ℓ to be the identity function. The converse is given by Lemma 6. □

Note that there is no need to premultiply the sources by α_ℓ as the channel directly computes the desired functions.

Linear computation coding performs quite well when the channel is (approximately) a linear function of its inputs. As seen in Example 3, this strategy does not always work well for non-linear channels. In the next section, we develop another coding scheme that is useful in some scenarios where linear computation coding fails.

3.4 Systematic Computation Codes

In the point-to-point setting, systematic transmission refers to first sending a block of the source uncoded across the channel and then using a code to refine the noisy version of the source [139; 140]. The decoder uses the uncoded block as side information to infer the source from the received codeword. Systematic transmission is a good framework for the digital upgrade of analog systems. We propose a systematic computation coding scheme that first uses uncoded transmission to send a noisy function to the decoder and then refines this function with a separation-based scheme.

We briefly consider the code used in Section 3.3.1.1 for sending the parity of binary sources over the M2MAC. Assume the sources are independent and the channel code is written in systematic form. This computation coding scheme is also systematic in that the encoders first send a noisy version of the desired sum and then refine it with parity-check bits. In this setting, we allow the channel to merge both the information bits and the parity-check bits to give a codeword that describes the sum of the sources. However, for an arbitrary MAC, we may not able to use the channel to combine our codewords. Therefore, we only use a joint source-channel code to send a noisy version of the function. We will then switch over to a separation-based scheme that uses a linear source code and a capacity-achieving MAC code at each encoder to refine the noisy function.

Theorem 12. *Let* f *be an arbitrary function and let* $U = f_\ell(S_1, \ldots, S_L)$*. Choose a Galois field* $\mathbb{F}$*, a linear function* g *over* $\mathbb{F}$ *and functions* $c_\ell : \mathcal{S}_\ell \to \mathbb{F}$ *for* $\ell = 1, \ldots, L$ *and* $d : \mathbb{F} \to \mathcal{U}$ *such that:*

$$Pr(d(g(c_1(S_1), \ldots, c_L(S_L))) = f_\ell(S_1, \ldots, S_L)) = 1.$$

Let $V = g(c_1(S_1), \ldots, c_L(S_L))$. *If the maximum sum rate of the MAC is symmetric*[3], *then the computation rate*

$$R_{COMP} = \frac{C_{MAC}}{C_{MAC} + LH(V|T)} \tag{3.41}$$

is achievable for any joint pmf of the form:

$$p_{T|X_1 \cdots X_L}(t|x_1, \ldots, x_L) \left(\prod_{\ell=1}^{L} p_{X_\ell|S_\ell}(x_\ell|s_\ell) \right) (p_{S_1 \cdots S_L}(s_1, \ldots, s_L)) \tag{3.42}$$

where

$$p_{T|X_1 \cdots X_L}(t|x_1, \ldots, x_L) \;=\; p_{Y|X_1 \cdots X_L}(t|x_1, \ldots, x_L)$$

Proof. (*Uncoded Transmission.*) At time step i for $1 \leq i \leq k$, encoder ℓ maps $S_\ell[i]$ into a channel input, $X_\ell[i]$, according to $p_{X_j|S_j}(x_j|s_j)$. The decoder collects the channel outputs to use as side information in the next phase, $T^k = Y^k$.

(*Refinement.*) First, let $C_\ell[i] = c_\ell(S_\ell[i])$, $\mathbf{c}_\ell = [C_\ell[1] \ \cdots \ C_\ell[k]]$, $V[i] = g(C_1[i], \ldots, C_L[i])$, and $\mathbf{v} = [V[1] \ \cdots \ V[k]]$. Choose $\epsilon > 0$. Using Theorem 4, choose a matrix $\mathbf{B}$ of size $k \times m$ over $\mathbb{F}$ for compressing V given side information T. If $m = k\left(\frac{H(V|T)+\epsilon}{\log_2|\mathbb{F}|}\right)$, then with probability than $1 - \frac{\epsilon}{2}$, $\mathbf{v}$ can be recovered from $\mathbf{w} = \mathbf{v}\mathbf{B}$. Each encoder applies $\mathbf{B}$ to $\mathbf{c}_\ell$ to get its message $\mathbf{w}_\ell = \mathbf{c}_\ell\mathbf{B}$. It then transmits $\mathbf{w}_\ell$ to the decoder using a multiple-access channel code:

$$\mathcal{E}_\ell^C : \mathbb{F}^m \to \mathcal{X}_\ell^n,$$

targeted at the symmetric maximum sum rate, C_{MAC}. From Theorem 6, the decoder can recover $\mathbf{w}_1, \ldots, \mathbf{w}_L$ with probability of error less than $\frac{\epsilon}{2}$ for n large enough if $\log_2|\mathbb{F}|mL < (n-k)(C_{\text{MAC}} - \epsilon)$. The decoder then computes:

$$\begin{aligned} \mathbf{w} &= g(\mathbf{w}_1, \ldots, \mathbf{w}_L) && (3.43)\\ &= g(\mathbf{c}_1\mathbf{B}, \cdots, \mathbf{c}_L\mathbf{B}) && (3.44)\\ &= \mathbf{v}\mathbf{B} && (3.45) \end{aligned}$$

[3] We assume that maximum sum rate of the MAC is symmetric according to Definition 16 to simplify the statement of the theorem. This can be removed for a more general (but more cumbersome) theorem statement.

from which it can recover $\mathbf{v}$ and apply the function d to get the desired function with total probability of error no greater than ϵ. Solving for the computation rate $R_{\text{COMP}} = \frac{k}{n}$ we get

$$k\left(\frac{H(V|T)+\epsilon}{\log_2|\mathbb{F}|}\right)\log_2|\mathbb{F}|L < (n-k)(C_{\text{MAC}}-\epsilon) \tag{3.46}$$

$$k\left(C_{\text{MAC}} + LH(V|T) + L\epsilon\right) < n(C_{\text{MAC}}-\epsilon) \tag{3.47}$$

$$\frac{k}{n} = \frac{C_{\text{MAC}}-\epsilon}{C_{\text{MAC}} + LH(V|T) + L\epsilon}. \tag{3.48}$$

By choosing ϵ sufficiently small, we can approach the desired rate. □

Remark 3. In some cases, we will have $H(S_\ell) < H(V|T)$ at one or more encoders. In this case, these encodes can just send their source in its entirety to the decoder to lower the overall computation rate.

Remark 4. Theorem 12 can be further generalized by allowing for a different ratio of source symbols to channel symbols in the uncoded phase. As it is currently stated, Theorem 12 uses one channel symbol per source symbol in the uncoded phase. This causes the computation rate to be upper bounded by 1.

Remark 5. If the mapping d in Theorem 12 is invertible and entropy-preserving, then $H(V|T) = H(U|T)$.

We show that systematic computation coding can outperform separation-based coding with the following example.

Example 4. Our setting is basically the same as the M2MAC (see Section 3.3.1.1). For simplicity, we make S_1 and S_2 independent $\mathcal{B}(\frac{1}{2})$ processes. The only difference is the channel performs a real addition, $W = S_1 + S_2$, and then noise is added mod-3 to get the output: $Y = W \oplus_3 Z$. The additive noise Z is distributed according to $P(Z = 0) = .8$ and $P(Z = 1) = P(Z = 2) = .1$. Our desired function is the parity of the sources, $U = S_1 \oplus S_2$. We would like to apply Theorem 12 for computing U. In the first phase, we just transmit the sources uncoded ($p_{X_j|S_j}(x_j|s_j) = \delta(x_j - s_j)$) to get side information T at the decoder. The conditional entropy of our desired function, $U = S_1 \oplus S_2$, is $H(U|T) = 0.60$ and the maximum sum rate of the MAC is $C_{\text{MAC}} = 0.66$. With this method, we can achieve $R_{\text{COMP}} = 0.35$. From Lemma 3, we get that the best separation-based scheme gives a computation rate of $R_{\text{COMP}} = 0.33$. In fact, we can outperform our systematic scheme by using the quasi-linearity of the channel to merge codewords. We simply employ the computation code for the M2MAC from Corollary 1 directly and map the output symbol 2 to 0 at the decoder. This gives us an improved computation rate of $R_{\text{COMP}} = 0.40$. None of these schemes meet the upper bound given by Lemma 7: $R_{\text{COMP}} \leq 0.66$.

In the next example, we compute the parity of independent binary sources over a binary multiplying channel.

Example 5. S_1 and S_2 are $\mathcal{B}(\alpha)$ sources. We are interested in sending the mod-2 sum $U = S_1 \oplus S_2$ over the binary multiplying channel (BMC) $Y = X_1 \cdot X_2$ where $\mathcal{X}_1 = \mathcal{X}_2 = \{0, 1\}$. In the first phase, we send the sources uncoded across the channel and in the second phase, we use a MAC code to send our update bins. The computation rates for both separation-based coding (Lemma 3) and our scheme (Theorem 12) are plotted in Figure 3.6. The upper bound is significantly higher than both achievable rates and is not shown on the plot. Our scheme outperforms separation-based coding for α between approximately 0.65 and 0.85 (and between 0.15 and 0.35 by symmetry). The underlying reason is that these input distributions get close to the maximum mutual information for the MAC, resulting in good side information for the second phase. It is quite surprising that our scheme even moderately outperforms separation as there is almost no structural match between the channel and the desired function.

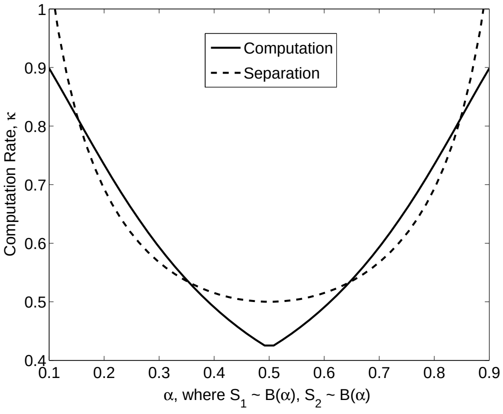

Figure 3.6: Computing parity over a binary multiplying channel

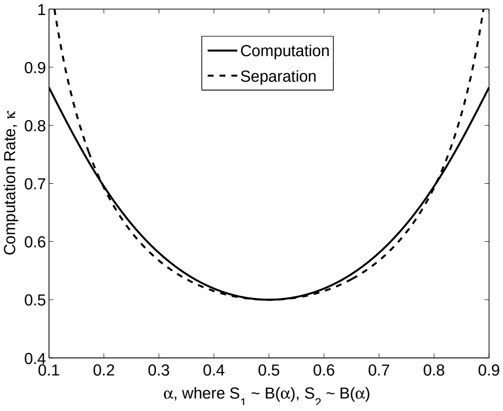

Figure 3.7: Computing a binary product over a mod-2 adder

The following example is the dual of the last one: we compute the product of binary independent sources over a mod-2 adder.

Example 6. S_1 and S_2 are independent $\mathcal{B}(\alpha)$ sources. We want to send $U = S_1 \cdot S_2$ over a mod-2 adder. The channel output is given by $Y = X_1 \oplus X_2$, $\mathcal{X}_1 = \mathcal{X}_2 = \{0, 1\}$. We can losslessly recover U from $V = S_1 \oplus_3 S_2$. Our scheme is to send S_1 and S_2 uncoded over the channel for phase one and then use this as side information to send V. The computation rates for both separation-based coding (Lemma 3) and our scheme (Theorem 12) are plotted in Figure 3.7. Again, the upper bound is significantly higher than both achievable rates and is not shown on the plot. Although the gains are marginal, that any gains are possible is surprising.

The next example demonstrates that there exists cases where computation coding is useful for sending a non-linear function over a noisy non-linear channel.

Example 7. S_1 and S_2 are independent sources drawn uniformly from $\{0, 1, 2\}$. We would like to know whether or not S_1 and S_2 are equal: $U = 1(S_1 \neq S_2)$. We can losslessly recover U from the linear function $V = S_1 \oplus_3 2S_2$ over GF(3). The channel is just $W = 1(S_1 \neq S_2)$ followed by a BSC with transition probability 0.1 to give Y. We employ Theorem 12. Our uncoded phase uses the sources directly; there is no remapping. In the update phase, we send V. With this strategy, we get a computation rate of $\kappa_{\text{COMP}} = 0.194$. Lemma 3 gives that the best separation-based computation rate is $\kappa_{\text{SEP}} = 0.168$. Finally, using Lemma 7 we get an upper bound of $\kappa_{\text{JOINT}} = 0.578$.

Systematic computation coding can also be extended to include multiple receivers and multiple functions along similar lines as in Section 3.3.2.

The two computation coding strategies developed in this chapter allow for higher rates when we want to communicate a function of the sources, instead of the sources in their entirety. In Chapters 5 and 7, we will use these strategies as building blocks for communicating messages over a network. In brief, intermediate nodes will collect functions of the messages and destinations, given sufficiently many functions, will be able to infer the original messages.

Appendix 3.A Upper Bounds

In this appendix, we develop upper bounds on the computation capacity. First, we give an upper bound on the distributed compression rate region. This can be combined with the multiple-access rate region to yield an upper bound on separation-based computation. Next, we give two upper bounds on the computation capacity.

3.A.1 Separation-Based Computation

For our upper bound, we will need a result of Orlitsky and Roche for computing with side information [117].

Let S_1 and S_2 be sources according to Definition 7 and let $f : \mathcal{S}_1 \times \mathcal{S}_2 \rightarrow \mathcal{U}$ be the desired function.

Definition 20. The elements of $\mathcal{S}_1$ are the vertices of the *characteristic graph*, G, of S_1, S_2, and f. Two distinct vertices, a and b, are connected if there is a $c \in \mathcal{S}_2$ such that $p_{S_1S_2}(a, c),\ p_{S_1S_2}(b, c) > 0$ and $f(a, c) \neq f(b, c)$. We say the graph is *complete* if each vertex is connected to every other vertex.

We say a set of vertices is independent if no two are connected. Let $\Gamma(G)$ be the collection of independent sets of the graph G.

Definition 21. The *conditional graph entropy* is given by:

$$H_G(S_1|S_2) \triangleq \min_{\substack{W-S_1-S_2\\ S_1\in W\in\Gamma(G)}} I(W;S_1|S_2), \tag{3.49}$$

where $W - S_1 - S_2$ signifies a Markov chain.

Lemma 4 (Orlitsky-Roche)**.** *Two sources, S_1 and S_2, are generated from the joint pmf $p_{S_1S_2}$. An encoder observes S_1 and must send enough bits to a decoder that sees S_2, such that the decoder can reconstruct $U = f(S_1, S_2)$ with a vanishing probability of error:*

$$\begin{aligned}
&\mathcal{E} : \mathcal{S}_1^k \to \{0,1\}^{kR} && (3.50)\\
&\mathcal{D} : \{0,1\}^{kR} \times \mathcal{S}_2^k \to \mathcal{U}^k && (3.51)\\
&\hat{U}^k = \mathcal{D}(\mathcal{E}(S_1^k), S_2^k) &&\\
&\lim_{k\to\infty} P(\hat{U}^k \neq U^k) = 0. && (3.52)
\end{aligned}$$

This is possible if and only if:

$$R > H_G(S_1|S_2). \tag{3.53}$$

We will use this side information result to generate individual rate constraints on separation-based schemes for distributed compression. There are M sources and a desired function $f(\cdot)$. Let $\mathbf{S}_\ell^C = (S_1, S_2, \ldots, S_{\ell-1}, S_{\ell+1}, \ldots, S_M)$.

Lemma 5. *The rate required for each encoder of a separation-based scheme for distributed compression of $U = f(S_1, S_2, \ldots, S_L)$ is lower bounded by*

$$R_\ell \geq H_G(S_\ell|\mathbf{S}_\ell^C) \quad \forall \ell \in \{1, 2, \ldots, L\}. \tag{3.54}$$

Proof. At encoder ℓ, assume that all other sources are available at the decoder. Clearly, this can only decrease the rate required of encoder ℓ. An application of Lemma 4 gives that a rate of $H_G(S_\ell|\mathbf{S}_\ell^C)$ is required from each encoder to reconstruct $f(S_1, S_2, \ldots, S_L)$ losslessly at the decoder. □

Proof of Lemma 3: We need the conditional graph entropy at each encoder used in the proof for Lemma 5 above. The characteristic graph for each encoder is complete. Therefore, the independent sets are the singletons and $W = S_\ell$. It follows that $H_G(S_\ell|\mathbf{S}_\ell^C) = I(W; S_\ell|\mathbf{S}_\ell^C)) = H(S_\ell|\mathbf{S}_\ell^C) = H(S_\ell)$.

3.A.2 Computation Capacity

We now give two upper bounds on the computation capacity. Our first bound comes from joining the encoders and reducing our problem to a point-to-point problem.

Definition 22. The *maximum joint sum rate* is the highest sum rate one can achieve on a MAC if the encoders are allowed to cooperate completely. It is given by:

$$C_{\text{JOINT}} = \max_{p(x_1,x_2,\ldots,x_L)} I(X_1, X_2, \ldots, X_L; Y). \tag{3.55}$$

Lemma 6. *The computation capacity is upper bounded as follows:*

$$C_{COMP} \geq \frac{C_{JOINT}}{H(U)}. \tag{3.56}$$

The proof follows immediately from joining the encoders and applying the point-to-point separation theorem. See [29, p. 216] for a full proof of the point-to-point separation theorem.

Our second bound is for the case when the sources are independent. We assume that the multiple-access channel has a symmetric maximum sum rate, C_{MAC}, according to Definition 16. This assumption can be removed for a more general statement of the lemma below.

Lemma 7. *If the sources are independent and the maximum sum rate of the MAC is symmetric then the reliable computation rate is upper bounded by*

$$\kappa_{IND} \leq \frac{C_{MAC}}{H(U)}. \tag{3.57}$$

Proof. Let $P_e = \Pr(\hat{U}^k \neq U^k)$. By Fano's inequality, we can show that $H(U^k|Y^n) \leq 1 + kP_e \log|\mathcal{U}|$. Now, set $\lambda_k = \frac{1}{k} + P_e \log|\mathcal{U}|$.

$$\begin{aligned}
H(U) &= \frac{1}{k}H(U^k) \\
&= \frac{1}{k}(H(U^k) - H(U^k|Y^n) + H(U^k|Y^n)) \\
&= \frac{1}{k}(I(U^k;Y^n) + H(U^k|Y^n)) \\
&\leq \frac{1}{k}I(U^k;Y^n) + \lambda_k \\
&\leq \frac{1}{k}I(X_1^n, X_2^n, \ldots, X_L^n; Y^n) + \lambda_k
\end{aligned}$$

where the last step is due to the data processing inequality. From here we are free to apply the standard MAC converse (see [29, pp.399-402]):

$$\frac{k}{n} \leq \frac{I(X_1, X_2, \ldots, X_L; Y)}{H(U)}.$$

for some pmf of the form $\prod_{j=1}^{M} p_{X_j}(x_j)$. The result follows immediately. □

It is also possible to give an upper bound that factors in the exact nature of the source correlations as in [153]. However, the focus of this thesis is on the gains that can be achieved by exploiting the structure rather than the correlations. All of our examples have independent sources so such a bound is unnecessary for the scope of this chapter.

Chapter 4

Compute-and-Forward: AWGN Networks

In this chapter, we develop a compute-and-forward scheme for wireless channel models. Specifically, we will show how to harness noisy linear combinations over the complex field for reliable computation over a finite field. The classical approach to wireless communication is to transform the physical layer into a set of *reliable bit pipes*, i.e. each link can accommodate a certain number of bits per time unit. These bit pipes can then be used seamlessly by higher layers in the protocol stack. Unfortunately, this approach means that wireless terminals must compete for the same fixed chunk of spectrum with diminishing rates as the network size increases. Recent work on cooperative communication has shown that this penalty can be overcome by adopting new strategies at the physical layer. The key idea is that users should help relay each other's messages by exploiting the broadcast and multiple-access properties of the wireless medium; properties that are usually viewed as a hindrance and are not captured by a bit pipe interface. To date, most proposed cooperative schemes have relied on one of the following three core relaying strategies:

- *Decode-and-Forward:* The relay decodes at least some part of the transmitted messages. The recovered bits are then re-encoded for collaborative transmission to the next relay. Although this strategy offers significant advantages, the relay is ultimately interference-limited as the number of transmitted messages increases [32; 83; 79; 38].

- *Compress-and-Forward:* The signal observed at the relay is vector quantized and this information is passed towards the destination. If the destination receives information from multiple relays, it can treat the network as a multiple-input multiple-output (MIMO) channel. Unfortunately, since no decoding is performed at intermediate nodes, noise builds up as messages traverse the network [32; 79; 72; 6; 131].

- *Amplify-and-Forward:* The relay simply acts as a repeater and transmits a scaled version of its observation. Like compress-and-forward, this strategy converts the network into a large MIMO channel with the added possibility of a beamforming gain. However, noise also builds up with each retransmission. [136; 83; 55; 16; 38].

Our compute-and-forward strategy simultaneously affords protection against noise and the opportunity to exploit interference for cooperative gains. Whereas compress-and-forward and amplify-and-forward convert a network into a set of noisy linear equations, compute-and-forward can convert it into a set of *reliable linear equations*. These equations can in turn be used for a digital implementation of cooperative schemes that could fit into a (slightly revised) network protocol stack.

We will develop a general framework for compute-and-forward that can be used in any relay network with linear channels and additive white Gaussian noise (AWGN).[1] Transmitters send out messages taking values in a prime-sized finite field and relays recover linear equations of the messages over the same field. To exploit the noisy linear equations provided by the channel, we use nested lattice codes as they have a linear structure and are well-suited for AWGN channels. As in the discrete case, the performance of this scheme is outside the reach of the usual random coding arguments. In Chapter 5, we will compare compute-and-forward to classical relaying strategies in two network scenarios, one based on distributed MIMO and the other wireless network coding. Classical relaying strategies perform well in either low or high signal-to-noise ratio (SNR) regimes. As we will see, compute-and-forward offers advantages in moderate SNR regimes where both interference and noise are significant factors.

Summary of Results

Our basic strategy is to take messages from a finite field, map them onto lattice points, and transmit (dithered versions of) these across the channel. Each relay observes a linear combination of these lattice points and attempts to decode an integer combination of them. This equation of lattice points is finally mapped back to a linear equation over a finite field. Our main theorems are summarized below:

- Theorems 13 and 14 give our achievable rates for sending equations over a finite field from transmitters to relays. The strategy relies on a nested lattice code which is developed in Theorem 15.
- Theorems 16, 17, and 18 give sufficient conditions on the equation coefficients so that a destination can recover one or more of the original messages.
- In Theorems 19 and 20 we generalize our compute-and-forward scheme to include successive cancellation and superposition coding.

[1] The material in this chapter is drawn from [108].

Related Work

There is a large body of work on lattice codes and their applications in communications. We cannot do justice to all this work here and point the interested reader to an excellent survey by Zamir [160]. The basic insight is that, for many AWGN networks of interest, nested lattice codes can approach the performance of standard random coding arguments. One key result by Erez and Zamir showed that nested lattice codes (combined with lattice decoding) can achieve the capacity of the point-to-point AWGN channel [42]. More generally, Zamir, Shamai, and Erez demonstrated how to use nested lattice codes for many classical AWGN multi-terminal problems in [162]. Subsequent work by El Gamal, Caire, and Damen showed that nested lattice codes achieve the diversity-multiplexing tradeoff of MIMO channels [50]. Recall that, in general, structured codes are not sufficient to prove capacity results. For instance, group codes cannot approach the capacity of asymmetric discrete memoryless channels [2].

It has now become clear that for certain network communication scenarios, structured codes can actually outperform standard random coding arguments. [2] For AWGN networks, nested lattice codes have been shown to outperform i.i.d. random codes in several scenarios apart from those considered in this thesis. For instance, Philosof et al. demonstrated that lattice codes enable distributed dirty paper coding for Gaussian multiple-access channels in [121]. Subsequent work by Sanderovich, Peleg, and Shamai used lattices to derive better scaling laws for decentralized processing in cellular networks [129].

There has also been a great deal of interest in using lattice codes for wireless network coding (see Chapter 5 for our contribution). Narayanan, Wilson, and Sprintson developed a nested lattice strategy for the two-way relay channel [101]. Nam, Chung, and Lee studied the two-way relay channel and Gaussian multiple-access networks with asymmetric power constraints in [99; 100].

Work on interference alignment by Maddah-Ali, Motahari, and Khandani [91] as well as Cadambe and Jafar [21] has shown that large gains are possible for interference channels at high SNR. The key is to have users transmit along subspaces chosen such that all interference stacks up in the same dimensions at the receivers. Lattice codes can be used to realize these gains at finite SNR. Bresler, Parekh, and Tse used lattice codes to approximate the capacity of the many-to-one and one-to-many interference channels to within a constant number of bits [19]. This scheme was employed for bursty interference channels in [71]. For symmetric interference channels, Sridharan et al. developed a layered lattice strategy in [149].

Krithivasan and Pradhan have employed nested lattice codes for distributed compression of linear functions of jointly Gaussian sources [80]. Wagner derived an outer bound for the Gaussian case in [155]. Finally, recent work by He and Yener has shown that lattices are useful for information theoretic secrecy [59]. See also [1].

[2] In a recent paper, we made a case for structured random codes in the proofs of network capacity theorems [106].

4.1 Problem Statement

Our relaying strategy is applicable to any configuration of sources, relays, and destinations that are linked through linear AWGN channels. We will refer to such configurations as AWGN networks. To simplify the description of the scheme, we will first focus on how to deliver equations to a single set of relays. We will then show how a destination, given sufficiently many equations, can recover the intended messages. These two components are sufficient to completely describe an achievable rate region for any AWGN network. We begin with the necessary definitions for L transmitters to send equations to M relays over a wireless channel. Then, we will provide a framework for recovering the original messages given a set of equations.

Remark 6. In fact, one can apply this strategy to non-linear and non-Gaussian channels as well by using the modulo-lattice transformation developed by Erez and Zamir in [43].

4.1.1 Decoding Equations

We are primarily interested in narrowband wireless channel models so we will specify our encoding and decoding schemes for complex baseband and assume that modulation is handled separately. In Section 4.1.3, we will restate a few definitions for real-valued channel models so that we can give corollaries that specialize our results to real values.

Let $\mathbb{C}$ denote the complex field and $\mathbb{F}_p$ denote the finite field of size p where p is always assumed to be prime. Let $+$ denote addition over the complex field and $\oplus$ addition over the finite field. Furthermore, let $\sum$ denote summation over the complex field and $\bigoplus$ denote summation over the finite field. We also set $j = \sqrt{-1}$ and assume that the log operation is with respect to base 2. It will be useful to map between the finite field and a corresponding subset of the integers. We let $g : \mathbb{F}_p \rightarrow \{0, 1, 2, \dots, p-1\}$ be this one-to-one map. This is essentially an identity map except for the change of alphabet. If g or its inverse g^{-1} are applied to a vector we assume they operate element-wise.

We will use boldface lowercase letters to denote column vectors and boldface uppercase letters to denote matrices. For example, $\mathbf{h} \in \mathbb{C}^L$ and $\mathbf{H} \in \mathbb{C}^{M \times L}$. Let $\|\mathbf{h}\| \triangleq \sqrt{\sum_{i=1}^{L} |h[i]|^2}$ denote the ℓ^2-norm of $\mathbf{h}$. Also, let $\mathbf{h}^*$ and $\mathbf{h}^T$ denote the Hermitian (or conjugate) transpose and the regular transpose of $\mathbf{h}$, respectively. Finally, let $\mathbf{0}$ denote the zero vector, δ_ℓ denote the unit vector with 1 in the ℓ^{th} entry and 0 elsewhere, and $\mathbf{I^{M \times M}}$ denote the identity matrix of size M.

Our main goal is to reliably transmit messages across the network at the highest possible rates. However, intermediate nodes in the network may only need to decode a function of the messages. These functions can be decoded at high rates if we choose their algebraic structure to match the structure of the channel. Thus, we will split our messages over the finite field into two parts: one for the real part of the channel and the other for the imaginary part.

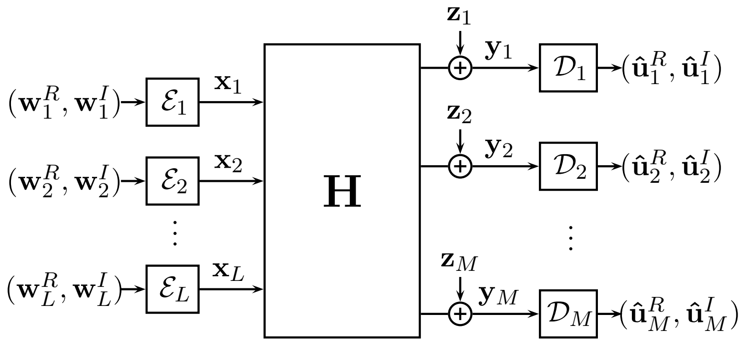
$(\mathbf{w}_1^R, \mathbf{w}_1^I)$
$\mathcal{E}_1$
$\mathbf{x}_1$
$(\mathbf{w}_2^R, \mathbf{w}_2^I)$
$\mathcal{E}_2$
$\mathbf{x}_2$
$(\mathbf{w}_L^R, \mathbf{w}_L^I)$
$\mathcal{E}_L$
$\mathbf{x}_L$
$\mathbf{H}$
$\mathbf{z}_1$
$\mathbf{y}_1$
$\mathcal{D}_1$
$(\hat{\mathbf{u}}_1^R, \hat{\mathbf{u}}_1^I)$
$\mathbf{z}_2$
$\mathbf{y}_2$
$\mathcal{D}_2$
$(\hat{\mathbf{u}}_2^R, \hat{\mathbf{u}}_2^I)$
$\mathbf{z}_M$
$\mathbf{y}_M$
$\mathcal{D}_M$
$(\hat{\mathbf{u}}_M^R, \hat{\mathbf{u}}_M^I)$

Figure 4.1: L transmitters reliably communicate linear functions to M relays.

Definition 23 (Messages)**.** Each transmitter (indexed by $\ell = 1, 2, \ldots, L$) has two length-k_ℓ vectors over a prime-size finite field, $\mathbf{w}_\ell^R, \mathbf{w}_\ell^I \in \mathbb{F}_p^{k_\ell}$. The superscript denotes whether the vector is intended for the real part or the imaginary part of the channel. Together these vectors are the *message* of transmitter ℓ, $\mathbf{w}_\ell = (\mathbf{w}_\ell^R, \mathbf{w}_\ell^I)$. Without loss of generality, we assume that the transmitters are numbered by increasing message length. Since we are interested in functions of these messages vectors, we zero-pad them to a common length $k \triangleq \max_\ell k_\ell$ prior to encoding.

Remark 7. We are interested in average probability of error results so we assume that all messages are drawn independently and uniformly from the set of all possible values.

Definition 24 (Encoders)**.** Each transmitter is equipped with an *encoder*, $\mathcal{E}_\ell$, that maps length-k messages over the finite field to length-n codewords over the complex field:

$$\mathcal{E}_\ell : \mathbb{F}_p^k \times \mathbb{F}_p^k \to \mathbb{C}^n \tag{4.1}$$

for $\ell = 1, 2, \ldots, L$.

Definition 25 (Message Rate)**.** The *message rate* r_ℓ of each transmitter is:

$$r_\ell = \frac{2k_\ell}{n} \log p \tag{4.2}$$

Definition 26 (Power Constraint)**.** Each transmitter's length-n channel input, $\mathbf{x}_\ell = \mathcal{E}_\ell(\mathbf{w}_\ell)$, is subject to the usual *power constraint*:

$$\frac{1}{n}\|\mathbf{x}_\ell\|^2 \leq \mathsf{SNR} \tag{4.3}$$

for $\mathsf{SNR} \geq 0$ and $\ell = 1, 2, \ldots, L$.

Remark 8. Note that asymmetric power constraints can be easily modeled by scaling the channel coefficients appropriately.

Definition 27 (Channel Model)**.** Each relay observes a noisy linear combination of the transmitted signals through the *channel*:

$$\mathbf{y}_m = \sum_{\ell=1}^{L} h_{m\ell}\mathbf{x}_\ell + \mathbf{z}_m \tag{4.4}$$

for $m = 1, 2, \ldots, M$ where $h_{m\ell} \in \mathbb{C}$ are the channel coefficients and $\mathbf{z}$ is i.i.d. circularly symmetric Gaussian noise, $\mathbf{z} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}^{n \times n})$. Let $\mathbf{h}_m = [h_{m1} \cdots h_{mL}]^T$ denote the vector of channel coefficients to relay m and let $\mathbf{H} = \{h_{m\ell}\}$ denote the entire channel matrix. Note that by this convention the m^{th} row of $\mathbf{H}$ is $\mathbf{h}_m^T$.

Remark 9. For our initial analysis, we will assume that the channel coefficients are fixed for all time. However, these results can easily be extended to the slow fading case under an outage formulation which we develop in Section 5.2.

Remark 10. Our coding scheme only requires that each relay knows the channel coefficients from each transmitter to itself. Specifically, relay m only needs to know $\mathbf{h}_m$. Each transmitter only needs to know its target message rate, not the realization of the channel.

Definition 28 (Decoders)**.** Each relay is equipped with a *decoder* that maps the observed channel output from the complex field back to two equations of messages over the finite field:

$$\mathcal{D}_m : \mathbb{C}^n \rightarrow \mathbb{F}_p^k \times \mathbb{F}_p^k \tag{4.5}$$

for $m = 1, 2, \ldots, M$.

Our choice of equation structure is inspired by the real-valued decomposition of a complex-valued channel. Recall that for any $\mathbf{H} \in \mathbb{C}^{M \times L}, \mathbf{x} \in \mathbb{C}^L,$ and $\mathbf{z} \in \mathbb{C}^M$, the channel output $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}$ can be written as:

$$\begin{bmatrix} \mathsf{Re}(\mathbf{y}) \\ \mathsf{Im}(\mathbf{y}) \end{bmatrix} = \begin{bmatrix} \mathsf{Re}(\mathbf{H}) & -\mathsf{Im}(\mathbf{H}) \\ \mathsf{Im}(\mathbf{H}) & \mathsf{Re}(\mathbf{H}) \end{bmatrix} \begin{bmatrix} \mathsf{Re}(\mathbf{x}) \\ \mathsf{Im}(\mathbf{x}) \end{bmatrix} + \begin{bmatrix} \mathsf{Re}(\mathbf{z}) \\ \mathsf{Im}(\mathbf{z}) \end{bmatrix}$$

where $\mathsf{Re}(\cdot)$ and $\mathsf{Im}(\cdot)$ represent the real and imaginary parts.

Note that the structure of the equations at an intermediate relay is not important, so long as they are useful for conveying messages to the destination at the highest possible rate.

Definition 29 (Desired Equations)**.** The goal of each relay is to reliably recover a *linear combination* of the transmitted messages. Relay m selects coefficients $q_{m\ell}^R, q_{m\ell}^I \in \mathbb{F}_p$ for

$\ell = 1, 2, \ldots, L$ and attempts to decode two equations:

$$\mathbf{u}_m^R = \bigoplus_{\ell=1}^{L} q_{m\ell}^R \mathbf{w}_\ell^R \oplus (-q_{m\ell}^I)\mathbf{w}_\ell^I \tag{4.6}$$

$$\mathbf{u}_m^I = \bigoplus_{\ell=1}^{L} q_{m\ell}^I \mathbf{w}_\ell^R \oplus q_{m\ell}^R \mathbf{w}_\ell^I \tag{4.7}$$

where $(-q_{m\ell})$ denotes the additive inverse of $q_{m\ell}$.

Although our desired equations are evaluated over the finite field $\mathbb{F}_p$, the channel operates over the complex field $\mathbb{C}$. Our coding scheme will allow us to efficiently exploit the channel for reliable computation if the desired equation coefficients are close to the channel coefficients in an appropriate sense. The definition below provides an embedding from the finite field to the complex field that will be useful in quantifying this closeness.

Definition 30 (Coefficient Vector)**.** The *equation with coefficient vector* $\mathbf{a}_m = [a_{m1}\ a_{m2}\ \cdots\ a_{mL}]^T \in \{\mathbb{Z} + j\mathbb{Z}\}^L$ is the linear combination of the transmitted messages with coefficients given by

$$q_{m\ell}^R = g^{-1}\left([\mathsf{Re}(a_{m\ell})] \mod p\right) \tag{4.8}$$

$$q_{m\ell}^I = g^{-1}\left([\mathsf{Im}(a_{m\ell})] \mod p\right). \tag{4.9}$$

Definition 31 (Probability of Error)**.** We say that the equations with coefficient vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$ are decoded with *average probability of error* ϵ if:

$$(\hat{\mathbf{u}}_m^R, \hat{\mathbf{u}}_m^I) \triangleq \mathcal{D}_m(\mathbf{y}_m) \tag{4.10}$$

$$P\left(\bigcup_{m=1}^{M} \{(\hat{\mathbf{u}}_m^R, \hat{\mathbf{u}}_m^I) \neq (\mathbf{u}_m^R, \mathbf{u}_m^I)\}\right) < \epsilon. \tag{4.11}$$

Definition 32 (Computation Rate)**.** We say that the *computation rate* $R(\mathbf{h}, \mathbf{a})$ is achievable if for any $\epsilon > 0$ and n large enough, there exist encoders and decoders, $\mathcal{E}_1, \ldots, \mathcal{E}_L, \mathcal{D}_1, \ldots, \mathcal{D}_M$, such that for any set of channels, $\mathbf{h}_1, \ldots, \mathbf{h}_M \in \mathbb{C}^L$, and coefficient vectors, $\mathbf{a}_1, \ldots, \mathbf{a}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$, all relays can recover their desired equations with average probability of error ϵ so long as the underlying message rates $r_1, \ldots, r_L$ satisfy:

$$r_\ell < \min_{m: a_{m\ell} \neq 0} R(\mathbf{h}_m, \mathbf{a}_m) \tag{4.12}$$

Remark 11. Note that the above definition means that each relay is free to choose which equation to recover. The only constraint is that the message rates be lower than the computation rate for that choice of coefficients. In many cases, it will be useful to have each relay recover the equation that is available at the highest computation rate.

Remark 12. The minimization in (4.12) is taken over non-zero equation coefficients since only the information in these messages is required for the desired equation.

4.1.2 Recovering Messages

After the relays decode their equations, they forward them towards the appropriate destinations which may only be interested in recovering the original messages. If the equation coefficients satisfy appropriate conditions, then the destinations can solve for the original messages. Although our scheme can be employed in any AWGN network, we will omit formal definitions for such networks and assume that equations of messages arrive at a destination. This may occur through a single layer of channels as described above or through multiple layers.

Definition 33 (Recovery)**.** We say that message $\mathbf{w}_\ell \in \mathbb{F}_p^{k_\ell}$ can be *recovered* at rate r_ℓ from the equations with coefficient vectors $\mathbf{a}_1, \ldots, \mathbf{a}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$ if for any $\epsilon > 0$ and n large enough, there exists a decoder $\mathcal{D} : \{\mathbb{F}_p^k \times \mathbb{F}_p^k\}^M \rightarrow \mathbb{F}_p^{k_\ell} \times \mathbb{F}_p^{k_\ell}$ such that:

$$\hat{\mathbf{w}}_\ell = \mathcal{D}\left((\mathbf{u}_1^R, \mathbf{u}_1^I), \ldots, (\mathbf{u}_M^R, \mathbf{u}_M^I)\right) \tag{4.13}$$

$$P\left(\hat{\mathbf{w}}_\ell \neq \mathbf{w}_\ell\right) < \epsilon \tag{4.14}$$

where $(\mathbf{u}_m^R, \mathbf{u}_m^I)$ represents the equations from relay m as in Definition 29.

4.1.3 Real-Valued Channels

We will now restate some of the above definitions for real-valued channel models. Each transmitter ℓ has a length-k_ℓ message that takes values over a prime-size finite field, $\mathbf{w}_\ell \in \mathbb{F}_p^{k_\ell}$. As before, we zero-pad all messages to a common length $k = \max_\ell k_\ell$ prior to encoding. The message rate is given by $r_\ell = \frac{k_\ell}{n} \log p$. The channel model is as in (4.4) except all inputs and outputs are over the reals instead of the complex field and the noise is i.i.d. Gaussian with mean zero and variance one.

For real-valued channels, the desired linear combination of messages at relay m is:

$$\mathbf{u}_m = \bigoplus_{\ell=1}^{L} q_{m\ell} \mathbf{w}_\ell \tag{4.15}$$

for coefficients $q_{m\ell} \in \mathbb{F}_p$. These finite field coefficients can be embedded in the reals as well. The equation with coefficient vector $\mathbf{a}_m \in \mathbb{Z}^L$ is the linear combination of the transmitted messages with coefficients given by $q_{m\ell} = [g^{-1}(a_{m\ell})] \mod p$. The rest of the definitions map in a straightforward fashion to the reals.

4.2 Nested Lattice Codes

In order to allow relays to decode integer combinations of codewords, we need codebooks with a linear structure. We will use lattice codes that have both good statistical and good algebraic properties. Erez and Zamir showed that nested lattice codes can approach the capacity of point-to-point AWGN channels in [42]. These codes operate under a modulo arithmetic that is well-suited for mapping operations over a finite field to the complex field.

First, we will provide some necessary definitions from [42] on nested lattice codes. Note that all of these definitions are given over $\mathbb{R}^n$. Our scheme will use the same lattice code over the real and imaginary parts of the channel input (albeit with different messages). Next, we will describe how to construct nested lattice codes which have certain desirable properties at sufficiently high dimension.

4.2.1 Lattice Definitions

Definition 34 (Lattice)**.** An n-dimensional *lattice*, Λ, is a set of points in $\mathbb{R}^n$ such that if $\mathbf{x}, \mathbf{y} \in \Lambda$, then $\mathbf{x} + \mathbf{y} \in \Lambda$, and if $\mathbf{x} \in \Lambda$, then $-\mathbf{x} \in \Lambda$. A lattice can always be written in terms of a lattice generator matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$:

$$\Lambda = \{\mathbf{x} = \mathbf{L}\mathbf{w} : \mathbf{w} \in \mathbb{Z}^n\}. \tag{4.16}$$

Note that the origin is always a point in the lattice.

Definition 35 (Nested Lattices)**.** A lattice Λ is said to be *nested* in a lattice Λ_1 if $\Lambda \subseteq \Lambda_1$. We will sometimes refer to Λ as the coarse lattice and Λ_1 as the fine lattice. More generally, a sequence of lattices $\Lambda, \Lambda_1, \ldots, \Lambda_L$ is nested if $\Lambda \subseteq \Lambda_1 \subseteq \cdots \subseteq \Lambda_L$.

Definition 36 (Quantizer)**.** A *lattice quantizer* is a map, $Q_\Lambda : \mathbb{R}^n \to \Lambda$, that sends a point, $\mathbf{x}$, to the nearest lattice point in Euclidean distance:

$$Q_\Lambda(\mathbf{x}) = \arg\min_{\lambda \in \Lambda} ||\mathbf{x} - \lambda||. \tag{4.17}$$

Definition 37 (Voronoi Region)**.** The *fundamental Voronoi region*, $\mathcal{V}$, of a lattice, is the set of all points in $\mathbb{R}^n$ that are closest to the zero vector: $\mathcal{V} = \{\mathbf{x} : Q_\Lambda(\mathbf{x}) = \mathbf{0}\}$. Let $\text{Vol}(\mathcal{V})$ denote the volume of $\mathcal{V}$.

Definition 38 (Modulus)**.** Let $[\mathbf{x}] \bmod \Lambda$ denote the quantization error of $\mathbf{x} \in \mathbb{R}^n$ with respect to the lattice Λ:

$$[\mathbf{x}] \bmod \Lambda = \mathbf{x} - Q_\Lambda(\mathbf{x}) \tag{4.18}$$

For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and $\Lambda \subseteq \Lambda_1$, the mod Λ operation satisfies:

$$[\mathbf{x} + \mathbf{y}] \bmod \Lambda = [[\mathbf{x}] \bmod \Lambda + \mathbf{y}] \bmod \Lambda \tag{4.19}$$

$$[Q_{\Lambda_1}(\mathbf{x})] \bmod \Lambda = [Q_{\Lambda_1}([\mathbf{x}] \bmod \Lambda)] \bmod \Lambda \tag{4.20}$$

$$[a\mathbf{x}] \bmod \Lambda = [a[\mathbf{x}] \bmod \Lambda] \bmod \Lambda \qquad \forall a \in \mathbb{Z} \tag{4.21}$$

$$\beta[\mathbf{x}] \bmod \Lambda = [\beta\mathbf{x}] \bmod \beta\Lambda \qquad \forall \beta \in \mathbb{R} \tag{4.22}$$

Definition 39 (Nested Lattice Codes)**.** A *nested lattice code* $\mathcal{L}$ is the set of all points of a fine lattice Λ_1 that are within the fundamental Voronoi region $\mathcal{V}$ of a coarse lattice Λ:

$$\mathcal{L} = \Lambda_1 \cap \mathcal{V} = \{\mathbf{x} : \mathbf{x} = \lambda \bmod \Lambda, \lambda \in \Lambda_1\}. \tag{4.23}$$

The rate of a nested lattice code is:

$$R = \frac{1}{n} \log |\mathcal{L}| = \frac{1}{n} \log \frac{\mathrm{Vol}(\mathcal{V})}{\mathrm{Vol}(\mathcal{V}_1)}. \tag{4.24}$$

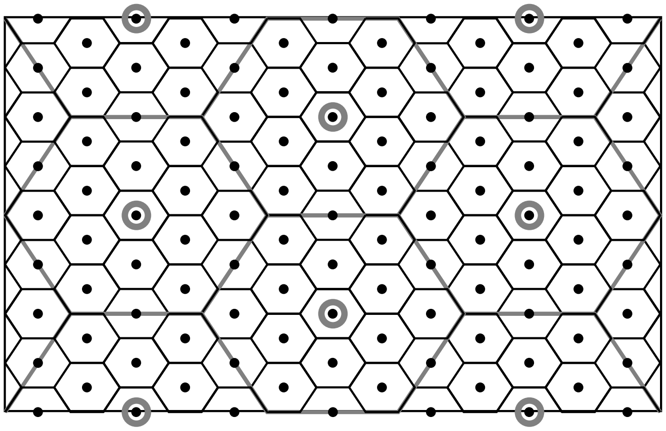

Figure 4.2: Part of a nested lattice $\Lambda \subset \Lambda_1 \subset \mathbb{R}^2$. Black points are elements of the fine lattice Λ_1 and gray circles are elements of the coarse lattice Λ. The Voronoi regions for the fine and coarse lattice are drawn in black and gray respectively. A nested lattice code is the set of all fine lattice points within the Voronoi region of the coarse lattice centered on zero.

Our scheme relies on mapping messages from a finite field to codewords from a nested lattice code. The relay will first decode an integer combination of lattice codewords and then convert this into an equation of the messages.

Definition 40 (Lattice Equation)**.** A *lattice equation* $\mathbf{v} \in \mathcal{L}$ is an integer combination of lattice codewords $\mathbf{t}_\ell \in \mathcal{L}$ modulo the coarse lattice:

$$\mathbf{v} = \left[\sum_{\ell=1}^{L} a_\ell t_\ell \right] \mod \Lambda \tag{4.25}$$

for some coefficients $a_\ell \in \mathbb{Z}$.

Let $\mathcal{B}(r)$ denote an n-dimensional ball of radius r:

$$\mathcal{B}(r) \triangleq \{\mathbf{x} : \|\mathbf{x}\| \leq r, \ \mathbf{x} \in \mathbb{R}^n\} \tag{4.26}$$

and let $\text{Vol}(\mathcal{B}(r))$ denote its volume.

Definition 41 (Covering Radius)**.** The *covering radius* of a lattice Λ is the smallest real number r_{COV} such that $\mathbb{R}^n \subseteq \Lambda + \mathcal{B}(r_{\text{COV}})$.

Definition 42 (Effective Radius)**.** The *effective radius* of a lattice with Voronoi region $\mathcal{V}$ is the real number r_{EFFEC} that satisfies $\text{Vol}(\mathcal{B}(r_{\text{EFFEC}})) = \text{Vol}(\mathcal{V})$.

Definition 43 (Moments)**.** The *second moment* of a lattice Λ is defined as the second moment per dimension of a uniform distribution over the fundamental Voronoi region $\mathcal{V}$:

$$\sigma_\Lambda^2 = \frac{1}{n\text{Vol}(\mathcal{V})} \int_{\mathcal{V}} \|\mathbf{x}\|^2 d\mathbf{x}. \tag{4.27}$$

The *normalized second moment* of a lattice is given by:

$$G(\Lambda) = \frac{\sigma_\Lambda^2}{(\text{Vol}(\mathcal{V}))^{2/n}} \tag{4.28}$$

The following three definitions are the basis for proving AWGN channel coding theorems using nested lattice codes.

Definition 44 (Covering Goodness)**.** A sequence of lattices $\Lambda^{(n)} \subset \mathbb{R}^n$ is *covering good* if:

$$\lim_{n\to\infty} \frac{r_{\text{COV}}^{(n)}}{r_{\text{EFFEC}}^{(n)}} = 1. \tag{4.29}$$

Such lattices were shown to exist by Rogers [127].

Definition 45 (Quantization Goodness)**.** A sequence of lattices $\Lambda^{(n)} \subset \mathbb{R}^n$ is *good for mean-squared error (MSE) quantization* if:

$$\lim_{n\to\infty} G(\Lambda^{(n)}) = \frac{1}{2\pi e}. \tag{4.30}$$

Zamir, Feder, and Poltyrev showed that sequences of such lattices exist in [161].

Definition 46 (AWGN Goodness)**.** Let $\mathbf{z}$ be a length-n random vector with distribution $\mathcal{N}(0, \sigma_Z^2 \mathbf{I}^{n\times n})$. The volume-to-noise ratio of a lattice is given by:

$$\mu(\Lambda, P_e) = \frac{(\text{Vol}(\mathcal{V}))^{2/n}}{\sigma_Z^2} \tag{4.31}$$

where σ_Z^2 is chosen such that $\Pr\{\mathbf{z} \notin \mathcal{V}\} = P_e$. A sequence of lattices $\Lambda^{(n)}$ is *AWGN good* if

$$\lim_{n\to\infty} \mu(\Lambda^{(n)}, P_e) = 2\pi e \quad \forall P_e \in (0,1) \tag{4.32}$$

and for fixed volume-to-noise ratio greater than $2\pi e$, the probability of error decays exponentially in n. In [122], Poltyrev demonstrated the existence of such lattices.

4.2.2 Lattice Constructions

Our nested lattice codes are a slight variant of those used by Erez and Zamir to approach the capacity of a point-to-point AWGN channel [42]. As in their considerations, we will have a coarse lattice that is covering, quantization, and AWGN good and a fine lattice that is AWGN good. We generalize this construction to include multiple nested fine lattices all of which are AWGN good. This will allow each transmitter to operate at a different rate.

Lemma 8 (Erez-Litsyn-Zamir)**.** *There exists a sequence of lattices $\Lambda^{(n)}$ that is simultaneously covering, quantization, and AWGN good.*

This is a corollary of their main result which develops lattices that are good in all the above senses as well as for packing [41, Theorem 5]. Note that these lattices are built using Construction A which is described below.

We will use a coarse lattice Λ of dimension n from Lemma 8 scaled such that its second moment is equal to $\frac{\mathsf{SNR}}{2}$. Let $\mathbf{L} \in \mathbb{R}^{n\times n}$ denote the generator matrix of this lattice. Our fine lattices are defined using the following procedure (the first three steps of which are often referred to as Construction A [88; 41]):

1. Draw a matrix $\mathbf{G}_L \in \mathbb{F}_p^{n\times k_L}$ with every element chosen i.i.d. according to the uniform distribution over $\{0, 1, 2, \ldots, p-1\}$. Recall that p is prime.

2. Define the codebook $\mathcal{C}_L$ as follows:

$$\mathcal{C}_L = \left\{ \mathbf{c} = \mathbf{G}_L \mathbf{w} : \mathbf{w} \in \mathbb{F}_p^{k_L} \right\}. \tag{4.33}$$

   All operations in this step are over $\mathbb{F}_p$.

3. Form the lattice $\tilde{\Lambda}_L$ by projecting the codebook into the reals by $g(\cdot)$, scaling down by a factor of p, and placing a copy at every integer vector. This tiles the codebook over $\mathbb{R}^n$:

$$\tilde{\Lambda}_L = p^{-1} g(\mathcal{C}_L) + \mathbb{Z}^n \tag{4.34}$$

4. Rotate $\tilde{\Lambda}_L$ by the generator matrix of the coarse nested lattice to get the fine lattice for transmitter L:

$$\Lambda_L = \mathbf{L}\tilde{\Lambda}_L \tag{4.35}$$

5. Repeat steps 1) - 4) for each transmitter $\ell = 1, 2, \ldots, L-1$ by replacing $\mathbf{G}_L$ with $\mathbf{G}_\ell$ which is defined to be the first k_ℓ columns of $\mathbf{G}_L$.

Clearly, any pair of fine lattices $\Lambda_{\ell_1}, \Lambda_{\ell_2}, 1 \leq \ell_1 < \ell_2 < L$ are nested since all elements of $\mathcal{C}_{\ell_1}$ can be found from $\mathbf{G}_{\ell_2}$ by multiplying by all $\mathbf{w} \in \mathbb{F}^{n \times k_{\ell_2}}$ with zeros in the last $\ell_2 - \ell_1$ elements. Also note that $\Lambda = \mathbf{L}\mathbb{Z}^n$ is nested within each fine lattice by construction. Finally, we get the desired set of nested lattices $\Lambda \subseteq \Lambda_1 \subseteq \cdots \subseteq \Lambda_L$.

By the union bound, we get that:

$$\begin{aligned} \Pr\left(\bigcup_{\ell=1}^{L} \{\text{rank}(\mathbf{G}_\ell) < k_\ell\} \right) &\leq \sum_{\ell=1}^{L} \sum_{\substack{\mathbf{w} \in \mathbb{F}_p^{k_\ell} \\ \mathbf{w} \neq \mathbf{0}}} \Pr\{\mathbf{G}_\ell \mathbf{w} = \mathbf{0}\} \\ &\leq p^{-n} \sum_{\ell=1}^{L} (p^{k_\ell} - 1) \end{aligned} \tag{4.36}$$

Thus, so long as p or $k_1, \ldots, k_L$ grow appropriately with n, all matrices $\mathbf{G}_1, \ldots, \mathbf{G}_L$ are full rank with probability that goes to 1. Note that if $\mathbf{G}_\ell$ has full rank, then the number of fine lattice points in the fundamental Voronoi region $\mathcal{V}$ of the coarse lattice is given by $|\Lambda_\ell \cap \mathcal{V}| = p^{k_\ell}$ so that the rate of Λ_ℓ is

$$R_\ell = \frac{1}{n} \log |\Lambda_\ell \cap \mathcal{V}| = \frac{k_\ell}{n} \log_2 p = \frac{1}{2} r_\ell \tag{4.37}$$

as desired. Furthermore, each message vector $\mathbf{w} \in \mathbb{F}_p^{k_\ell}$ can be put into one-to-one correspondence with a point in $\Lambda_\ell \cap \mathcal{V}$ as shown in Lemma 12. In Appendix 4.A.2, we show that the fine lattices are AWGN good so long as $\frac{n}{p} \rightarrow 0$ as n grows. It is clear that one can choose $p, k_1, \ldots, k_L$ so that the fine lattices have all the desired properties. One possibility is to let p grow like $n \log n$ and set $k_\ell = \lfloor nR_\ell (\log p)^{-1} \rfloor$.

Lemma 9. *Any lattice Λ that results from Construction A has a full-rank generator matrix $\mathbf{L}$.*

Proof. Note that $\mathbb{Z}^n \subset \Lambda$ so that Λ contains all of the unit vectors by default. Thus, $\mathbf{L}$ spans $\mathbb{R}^n$ and is full rank. □

Remark 13. We require that the fine lattices are generated from full-rank submatrices of the same finite field codebook so that it is possible to compute linear equations over messages with different rates. The full rank condition on the coarse lattice allows us to move between lattice equations and equations of finite field messages.

In [41; 81], some useful properties of nested lattices derived from Construction A are established. These apply to our construction as well and we repeat them below.

Lemma 10. *Choose any fine lattice Λ_ℓ from the construction above and let $\Lambda_\ell(i)$ denote the i^{th} point in $\Lambda_\ell \cap \mathcal{V}$ for $i = 0, 1, 2, \ldots, p^{k_\ell} - 1$. We have that:*

- *$\Lambda_\ell(i)$ is uniformly distributed over $p^{-1}\Lambda \cap \mathcal{V}$.*
- *For any $i_1 \neq i_2, [\Lambda_\ell(i_1) - \Lambda_\ell(i_2)] \bmod \Lambda$ is uniformly distributed over $\{p^{-1}\Lambda\} \cap \mathcal{V}$.*

Thus, each fine lattice can be interpreted as a diluted version of a scaled down coarse lattice $p^{-1}\Lambda$.

4.3 Lattice Computation Codes

We now develop our main result which provides achievable rates for computing linear functions of messages across AWGN networks. As we will see, these rates are often far higher than those available either by first recovering all messages then computing the function or by standard i.i.d. random coding arguments.

First, we will state the achievable computation rate and show how to optimally choose the free parameter in the rate expression. Next, we will provide some illustrative examples. In Section 4.3.1, we give a detailed proof of the main result.

Let $(x)^+ \triangleq \max(x, 0)$.

Theorem 13. *For channel vectors $\mathbf{h}_1, \ldots, \mathbf{h}_M \in \mathbb{C}^L$, the relays can recover any set of linear equations with coefficient vectors $\mathbf{a}_1, \ldots, \mathbf{a}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$ so long as the message rates are less than the computation rate*

$$r_\ell < \min_{m: a_{ml} \neq 0} \left(R(\mathbf{h}_m, \mathbf{a}_m)\right)^+ \tag{4.38}$$

$$R(\mathbf{h}_m, \mathbf{a}_m) = \log\left(\frac{\mathsf{SNR}}{|\alpha_m|^2 + \mathsf{SNR}\|\alpha_m \mathbf{h}_m - \mathbf{a}_m\|^2}\right) \tag{4.39}$$

for some choice of $\alpha_m \in \mathbb{C}$.

Theorem 14. *For a given $\mathbf{h}_m \in \mathbb{C}^L$, $\mathbf{a}_m \in \{\mathbb{Z}+j\mathbb{Z}\}^L$, the computation rate given in Theorem 13 is uniquely maximized by choosing α_m to be the MMSE coefficient*

$$\alpha_{\text{MMSE}} = \frac{\mathsf{SNR}\ \mathbf{h}_m^* \mathbf{a}_m}{1 + \mathsf{SNR}\|\mathbf{h}_m\|^2} \tag{4.40}$$

which results in a computation rate

$$R(\mathbf{h}_m, \mathbf{a}_m) = \log\left(\left(\|\mathbf{a}_m\|^2 - \frac{\mathsf{SNR}\ |\mathbf{h}_m^* \mathbf{a}_m|^2}{1 + \mathsf{SNR}\|\mathbf{h}_m\|^2}\right)^{-1}\right) \tag{4.41}$$

Proof. Note that the denominator of (4.39) is quadratic in α_m. Thus, it can be uniquely minimized by setting the first derivative to zero.

$$\begin{aligned} f(\alpha_m) &= \alpha_m^* \alpha_m + \mathsf{SNR}(\alpha_m \mathbf{h}_m - \mathbf{a}_m)^*(\alpha_m \mathbf{h}_m - \mathbf{a}_m) \\ \frac{df}{d\alpha_m} &= 2\alpha_m + \mathsf{SNR}(2\alpha_m \mathbf{h}_m^* \mathbf{h}_m - 2\mathbf{h}_m^* \mathbf{a}_m) = 0 && (4.42) \\ \alpha_m(2 + 2\mathsf{SNR}\|\mathbf{h}_m\|^2) &= 2\mathsf{SNR}\ \mathbf{h}_m^* \mathbf{a}_m && (4.43) \end{aligned}$$

We solve this to get α_{MMSE} and plug back into $f(\alpha_m)$.

$$\begin{aligned} f(\alpha_{\text{MMSE}}) =& \frac{\mathsf{SNR}^2 |\mathbf{h}_m^* \mathbf{a}_m|^2}{(1 + \mathsf{SNR}\|\mathbf{h}_m\|^2)^2} + \frac{\mathsf{SNR}^3 \|\mathbf{h}_m\|^2 |\mathbf{h}_m^* \mathbf{a}_m|^2}{(1 + \mathsf{SNR}\|\mathbf{h}_m\|^2)^2} \ \cdots \\ & \cdots \ - 2\frac{\mathsf{SNR}^2 |\mathbf{h}_m^* \mathbf{a}_m|^2}{1 + \mathsf{SNR}\|\mathbf{h}_m\|^2} + \mathsf{SNR}\|\mathbf{a}_m\|^2 && (4.44) \\ =& -\frac{\mathsf{SNR}^2 |\mathbf{h}_m^* \mathbf{a}_m|^2}{1 + \mathsf{SNR}\|\mathbf{h}_m\|^2} + \mathsf{SNR}\|\mathbf{a}_m\|^2 && (4.45) \end{aligned}$$

Substituting this into $\log\left(\frac{\mathsf{SNR}}{f(\alpha_{\text{MMSE}})}\right)$ yields the desired computation rate. □

The above two theorems can be interpreted in the following sense. Each transmitter uses a fixed (lattice) codebook with a specified rate. By setting $\alpha_m = 1$ and decoding to the closest lattice point (in Euclidean distance), the relays could recover any linear equation so long as the message rates are less than

$$\log\left(\frac{\mathsf{SNR}}{1+\mathsf{SNR}\|\mathbf{h}_m - \mathbf{a}_m\|^2}\right) \tag{4.46}$$

However, if each relay scales its channel output by the appropriate α prior to decoding, then higher rates are possible. This is because the channel output $\alpha_m \mathbf{y}_m = \sum \alpha_m h_{m\ell}\mathbf{x}_\ell + \alpha_m \mathbf{z}_m$ can be equivalently written as a channel output $\tilde{\mathbf{y}}_m = \sum \tilde{h}_{m\ell}\mathbf{x}_\ell + \tilde{\mathbf{z}}_m$ where $\tilde{h}_{m\ell} = \alpha_m h_{m\ell}$ and $\tilde{\mathbf{z}}_m$ is i.i.d. circularly symmetric Gaussian noise with variance $|\alpha_m|^2$. Since there is a rate penalty both for noise and for non-integer channel coefficients, then α should be used to optimally balance between the two as in Theorem 14. This is quite similar to the role of the MMSE scaling coefficient used by Erez and Zamir to achieve the capacity of the point-to-point AWGN channel in [42].

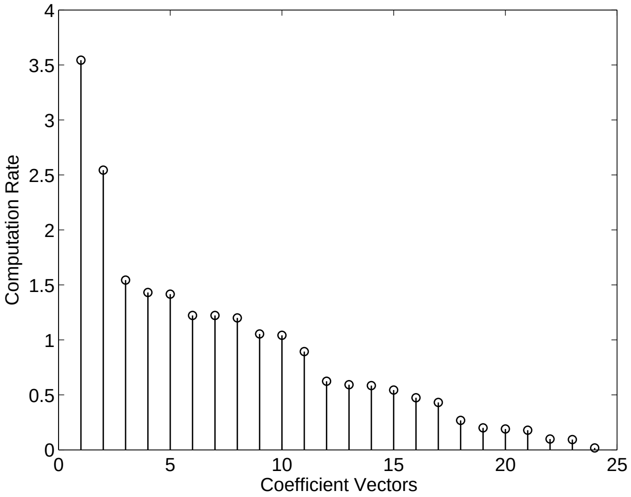

Figure 4.3: Computation rate at $\mathsf{SNR} = 5$dB for different coefficient vectors $\mathbf{a} \in \{\mathbb{Z} + j\mathbb{Z}\}^2$ for channel $\mathbf{h} = [-1.1744 + j2.1496 \;\; 1.2512 - j1.6335]^T$. The coefficient vector with the highest rate is $\mathbf{a} = [1 \;\; -1]^T$ with $R = 3.5436$. Using standard multiple-access codes, this equation can only be computed at rate $R_{\mathsf{MAC}} = 2.5301$.

We will often be interested in finding the coefficient vectors with the highest computation rates. This does not require a search over all integer vectors as most vectors trivially give zero rate.

Lemma 11. *For a given channel vector* $\mathbf{h} \in \mathbb{C}^L$*, the computation rate from Theorem 14 is zero if the coefficient vector* $\mathbf{a} \in \{\mathbb{Z} + j\mathbb{Z}\}^L$ *satisfies:*

$$\|\mathbf{a}\|^2 \geq 1 + \|\mathbf{h}\|^2\mathsf{SNR}. \tag{4.47}$$

Proof. Note that $|\mathbf{h}_m^*\mathbf{a}_m|^2 \leq \|\mathbf{h}_m\|^2\|\mathbf{a}_m\|^2$ by the Cauchy-Schwarz inequality. Using this, we can upper bound the achievable rate from Theorem 14:

$$\begin{aligned}
&\log\left(\left(\|\mathbf{a}_m\|^2 - \frac{\mathsf{SNR}\ |\mathbf{h}_m^*\mathbf{a}_m|^2}{1+\mathsf{SNR}\|\mathbf{h}_m\|^2}\right)^{-1}\right) &(4.48)\\
&= \log\left(\frac{1+\mathsf{SNR}\|\mathbf{h}_m\|^2}{\|\mathbf{a}_m\|^2 + \mathsf{SNR}\|\mathbf{h}_m\|^2\|\mathbf{a}_m\|^2 - \mathsf{SNR}\ |\mathbf{h}_m^*\mathbf{a}_m|^2}\right)\\
&\leq \log\left(\frac{1+\mathsf{SNR}\|\mathbf{h}_m\|^2}{\|\mathbf{a}_m\|^2}\right). &(4.49)
\end{aligned}$$

The result follows immediately. □

In Figure 4.3, we have plotted the computation rates for coefficient vectors with non-zero entries for an example channel of length 2. The vectors are sorted by descending rate and we only include one of the four possible rotations of each vector, $\mathbf{a}, -\mathbf{a}, j\mathbf{a}$, and $-j\mathbf{a}$, as these yield identical rates.

Remark 14. Note that each relay is free to decode more than one equation, so long as all the appropriate computation rates are satisfied. In some cases, it may be beneficial to recover a desired equation by first decoding equations of subsets of messages and then combining them.

Example 8. Let the channel matrix take values on the complex integers, $\mathbf{H} \in \{\mathbb{Z}+j\mathbb{Z}\}^{M\times L}$, and assume that each relay wants a linear equation with a coefficient vector that corresponds exactly to the channel coefficients, $\mathbf{a}_m = \mathbf{h}_m$. Using Theorem 14 the achievable computation rate is:

$$\begin{aligned}
R &= \log\left(\left(\|\mathbf{h}_m\|^2 - \frac{\mathsf{SNR}\|\mathbf{h}_m\|^4}{1+\mathsf{SNR}\|\mathbf{h}_m\|^2}\right)^{-1}\right) &(4.50)\\
&= \log\left(\frac{1+\mathsf{SNR}\|\mathbf{h}_m\|^2}{\|\mathbf{h}_m\|^2 + \mathsf{SNR}\|\mathbf{h}_m\|^4 - \mathsf{SNR}\|\mathbf{h}_m\|^4}\right) &(4.51)\\
&= \log\left(\frac{1}{\|\mathbf{h}_m\|^2} + \mathsf{SNR}\right) &(4.52)
\end{aligned}$$

Remark 15. One interesting special case of Example 8 is computing the sum of codewords $\mathbf{w}_1 \oplus \mathbf{w}_2$ over a two-user Gaussian multiple-access channel $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{z}$. To date, the best known achievable rate for this scenario is $\log\left(\frac{1}{2} + \mathsf{SNR}\right)$. Several papers (including our own) have studied this special case and it is an open problem as to whether the best known outer bound $\log(1 + \mathsf{SNR})$ is achievable [105; 101; 99]. Clearly, one can do better in the low SNR regime using standard multiple-access codes to recover all the messages then compute the sum to get $\frac{1}{2}\log(1 + 2\mathsf{SNR})$.

Example 9. Assume there are L transmitters and L relays. Receiver m wants to recover the message from transmitter m. This corresponds to setting the desired coefficient vector to be a unit vector $\mathbf{a}_m = [0 \cdots 0 \; 1 \; 0 \cdots 0]^T$ where the m^{th} element is 1 and the rest are 0. Substituting this choice of $\mathbf{a}_m$ into Theorem 14, we get that the messages can be decoded if their rates satisfy:

$$r_m < \log\left(\left(1 - \frac{\mathsf{SNR}|h_{mm}|^2}{1 + \mathsf{SNR}\|\mathbf{h}_m\|^2}\right)^{-1}\right) \tag{4.53}$$

$$= \log\left(\left(\frac{1 + \mathsf{SNR}\sum_{\ell \neq m}|h_{m\ell}|^2}{1 + \mathsf{SNR}\|\mathbf{h}_m\|^2}\right)^{-1}\right) \tag{4.54}$$

$$= \log\left(1 + \frac{\mathsf{SNR}|h_{mm}|^2}{1 + \mathsf{SNR}\|\mathbf{h}_m\|^2}\right) \tag{4.55}$$

This is exactly the rate achievable with standard multiple-access techniques if the relays ignore all other messages as noise. In Section 4.5, we will use successive cancellation of lattice equations to show that if a relay wants all of the messages, any point in the Gaussian multiple-access rate region is achievable with compute-and-forward.

The following example shows that it is useful to allow for a different rate at each transmitter.

Example 10. We have $L = 4$ transmitters and $M = 2$ relays. The channel vectors are $\mathbf{h}_1 = [4 \;\; -4 \;\; 1 \;\; -1]^T$ and $\mathbf{h}_2 = [1 \;\; 1 \;\; 2 \;\; 2]^T$. The desired coefficient vectors are $\mathbf{a}_1 = \mathbf{h}_1$ and $\mathbf{a}_2 = [0 \;\; 0 \;\; 1 \;\; 1]$. These equations can be reliably recovered so long as the message rates satisfy:

$$r_\ell < \begin{cases} \log\left(\frac{1}{34} + \mathsf{SNR}\right) & \ell = 1, 2 \\ \log\left(\frac{1}{2} + \frac{4\mathsf{SNR}}{1+2\mathsf{SNR}}\right) & \ell = 3, 4 \end{cases} \tag{4.56}$$

4.3.1 Proof of Theorem 13

We now provide a detailed description of our encoding and decoding scheme. The following four steps are a basic outline:

1. Each transmitter maps its message from the finite field onto an element of a nested lattice code.

2. Lattice codewords are dithered and transmitted over the channel.

3. Receivers reliably decode a linear equation of the lattice codewords.

4. These lattice equations are mapped back to the finite field to get the desired linear combination of messages.

We will first show how to go from messages to lattice codewords and from lattice equations back to linear combinations of messages. Then, we prove that linear equations of lattice codewords can be recovered at the desired computation rates. Finally, we will combine all of these steps to get our compute-and-forward scheme.

The following lemma provides a mapping from messages over the finite field to points in a nested lattice code.

Lemma 12. *The function $\phi_\ell : \mathbb{F}_p^{k_\ell} \rightarrow \Lambda_\ell \cap \mathcal{V}$ is one-to-one where:*

$$\phi_\ell(\mathbf{w}) = \big[\mathbf{L}p^{-1}g(\mathbf{G}_\ell \mathbf{w})\big] \mod \Lambda. \tag{4.57}$$

Proof. Since $\mathbf{G}_\ell$ is assumed to be full rank, it takes $\mathbf{w}$ to a unique point in $\mathcal{C}_\ell$. Thus, $p^{-1}g(\mathbf{G}_\ell\mathbf{w})$ maps $\mathbf{w}$ to a unique point in $[0,1)^n$. Lemma 9 shows that $\mathbf{L}$ is full rank so we just need show that the mod Λ operation is a bijection between $\mathbf{L}[0,1)^n$ and $\mathcal{V}$. Assume, for the sake of a contradiction, $\exists x, y \in \mathbf{L}[0,1)^n, x \neq y$ such that $[x] \mod \Lambda = [y] \mod \Lambda$. This implies that:

$$\begin{aligned} x - Q_\Lambda(x) &= y - Q_\Lambda(y) \\ [\mathbf{L}^{-1}(x - Q_\Lambda(x))] \mod \mathbb{Z}^n &= [\mathbf{L}^{-1}(y - Q_\Lambda(y))] \mod \mathbb{Z}^n \\ [\mathbf{L}^{-1}x] \mod \mathbb{Z}^n &= [\mathbf{L}^{-1}y] \mod \mathbb{Z}^n \\ x &= y \end{aligned}$$

where the third step follows since for any $\lambda \in \Lambda$, $\mathbf{L}^{-1}\lambda \in \mathbb{Z}^n$. A contradiction has been reached which, combined with the fact that $|\mathbb{F}_p^{k_\ell}| = |\Lambda_\ell \cap \mathcal{V}| = p^{k_\ell}$, shows that ϕ_ℓ is a one-to-one map. □

Assume the messages are encoded onto nested lattice codes using ϕ_ℓ from Lemma 12 and that the relays successfully recover a linear combination of lattice points. The following lemma shows how to convert these lattice equations back to the desired messages equations.

Lemma 13. *For messages $\mathbf{w}_\ell \in \mathbb{F}_p^{k_\ell}$ zero-padded to length k_m, let $\mathbf{u} = \sum q_\ell \mathbf{w}_\ell$ for some coefficients $q_\ell \in \mathbb{F}_p$. Also, set $\mathbf{t}_\ell = \phi_\ell(\mathbf{w}_\ell)$ and $\mathbf{v} = [\sum a_\ell \mathbf{t}_\ell] \mod \Lambda$ for some $a_\ell \in \mathbb{Z}$ such* *that* $q_\ell = g^{-1}([a_\ell] \mod p)$*. We have that* $\mathbf{u} = \phi_m^{-1}(\mathbf{v})$ *where*

$$\phi_m^{-1}(\mathbf{v}) = (\mathbf{G}_m^T\mathbf{G}_m)^{-1}\mathbf{G}_m^T g^{-1}\left(p[\mathbf{L}^{-1}\mathbf{v}] \mod \mathbb{Z}^n\right). \tag{4.58}$$

Proof. Recall that since $\mathbf{L}$ is the generator matrix of Λ, $\mathbf{L}^{-1}\Lambda = \mathbb{Z}^n$. Also note that since $\mathbf{w}_\ell$ is zero-padded to length k_m, then multiplying by $\mathbf{G}_m$ has the same effect as multiplying the original message by $\mathbf{G}_\ell$. We have that:

$$\begin{aligned}
&[\mathbf{L}^{-1}\mathbf{v}] \mod \mathbb{Z}^n && (4.59)\\
&= \left[\mathbf{L}^{-1}\sum_{\ell=1}^{L} a_\ell \mathbf{t}_\ell + \mathbf{L}^{-1}Q_\Lambda\left(\sum_{\ell=1}^{L} a_\ell \mathbf{t}_\ell\right)\right] \mod \mathbb{Z}^n && (4.60)\\
&= \left[\mathbf{L}^{-1}\sum_{\ell=1}^{L} a_\ell \mathbf{t}_\ell\right] \mod \mathbb{Z}^n && (4.61)\\
&= \left[\sum_{\ell=1}^{L} a_\ell\left(p^{-1}g(\mathbf{G}_m\mathbf{w}_\ell) - \mathbf{L}^{-1}Q_\Lambda(\mathbf{L}p^{-1}g(\mathbf{G}_m\mathbf{w}_\ell))\right)\right] \mod \mathbb{Z}^n && (4.62)\\
&= \left[\sum_{\ell=1}^{L} a_\ell p^{-1} g(\mathbf{G}_m\mathbf{w}_\ell)\right] \mod \mathbb{Z}^n && (4.63)
\end{aligned}$$

Multiplying by p we get:

$$\begin{aligned}
p[\mathbf{L}^{-1}\mathbf{v}] \mod \mathbb{Z}^n &= \left[\sum_{\ell=1}^{L} a_\ell g(\mathbf{G}_m\mathbf{w}_\ell)\right] \mod p\mathbb{Z}^n && (4.64)\\
&= \left[g\left(\bigoplus_{\ell=1}^{L} q_\ell \mathbf{G}_m\mathbf{w}_\ell\right)\right] \mod p\mathbb{Z}^n && (4.65)\\
&= g\left(\bigoplus_{\ell=1}^{L} q_\ell \mathbf{G}_m\mathbf{w}_\ell\right) && (4.66)
\end{aligned}$$

Applying g^{-1} to move back to the finite field we get:

$$g^{-1}\left(p[\mathbf{L}^{-1}\mathbf{v}] \mod \mathbb{Z}^n\right) = \mathbf{G}_m\bigoplus_{\ell=1}^{L} q_\ell \mathbf{w}_\ell \tag{4.67}$$

Finally, note that $\left(\mathbf{G}_m^T\mathbf{G}_m\right)^{-1}\mathbf{G}_m^T$ is the left-inverse of $\mathbf{G}_m$ and we get that $\phi_m^{-1}(\mathbf{v}) = \mathbf{u}$. □

The proofs for our lattice coding scheme will make use of random dither vectors which are available at the transmitters and relays as common randomness. Since our scheme works with respect to expectation over these vectors, then it is clear that (at least) one set of good fixed dither vectors exists. See Appendix 4.A.3 for more details. The following lemma from [42] captures a key property of dithered nested lattice codes.

Lemma 14 (Erez-Zamir)**.** *Let* $\mathbf{t}$ *be a random vector taking values on* $\mathbb{R}^n$*. If* $\mathbf{d}$ *is independent of* $\mathbf{t}$ *and uniformly distributed over* $\mathcal{V}$*, then* $[\mathbf{t}-\mathbf{d}] \bmod \Lambda$ *is also independent of* $\mathbf{t}$ *and uniformly distributed over* $\mathcal{V}$*.*

Theorem 15. *For any* $\epsilon > 0$ *and* n *large enough, there exist nested lattice codes* $\Lambda \subseteq \Lambda_1 \subseteq \cdots \subseteq \Lambda_L$ *with rates* $R_1, \dots, R_L$*, such that for all channel vectors* $\mathbf{h}_1, \dots, \mathbf{h}_M \in \mathbb{C}^L$ *and coefficient vectors* $\mathbf{a}_1, \dots, \mathbf{a}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$*, each relay can decode lattice equations* $\mathbf{v}_m^R, \mathbf{v}_m^I$ *where*

$$\mathbf{v}_m^R = \left[\sum_{\ell=1}^{L} \mathsf{Re}(a_{m\ell})\mathbf{t}_\ell^R - \mathsf{Im}(a_{m\ell})\mathbf{t}_\ell^I\right] \bmod \Lambda \tag{4.68}$$

$$\mathbf{v}_m^I = \left[\sum_{\ell=1}^{L} \mathsf{Im}(a_{m\ell})\mathbf{t}_\ell^R + \mathsf{Re}(a_{m\ell})\mathbf{t}_\ell^I\right] \bmod \Lambda \tag{4.69}$$

of transmitted lattice points $\mathbf{t}_\ell^R, \mathbf{t}_\ell^I \in \Lambda_\ell \cap \mathcal{V}$ *with total probability of error* ϵ *so long as:*

$$R_\ell < \min_{m: a_{m\ell} \neq 0} \left(R(\mathbf{h}_m, \mathbf{a}_m)\right)^+ \tag{4.70}$$

$$R(\mathbf{h}_m, \mathbf{a}_m) = \frac{1}{2}\log\left(\frac{\mathsf{SNR}}{|\alpha_m|^2 + \mathsf{SNR}\|\alpha_m \mathbf{h}_m - \mathbf{a}_m\|^2}\right) \tag{4.71}$$

for some choice of $\alpha_m \in \mathbb{C}$*.*

Proof. Each encoder is given two dither vectors, $\mathbf{d}_\ell^R$ and $\mathbf{d}_\ell^I$, which are independently drawn according to a uniform distribution over $\mathcal{V}$. All dither vectors are made available to each relay. Encoder ℓ generates a channel input:

$$\mathbf{x}_\ell = [\mathbf{t}_\ell^R - \mathbf{d}_\ell^R] \bmod \Lambda + j[\mathbf{t}_\ell^I - \mathbf{d}_\ell^I] \bmod \Lambda. \tag{4.72}$$

By Lemma 14, the real and imaginary parts of $\mathbf{x}_\ell$ are independent and uniform over $\mathcal{V}$ so $\frac{1}{n}E[\|\mathbf{x}_\ell\|^2] = \mathsf{SNR}$, with expectation taken over the dithers. In Appendix 4.A.3, we argue that there exist fixed dithers that meet the power constraint set forth in (4.3).

The channel output at relay m is:

$$\mathbf{y}_m = \sum_{\ell=1}^{L} h_{m\ell}\mathbf{x}_\ell + \mathbf{z}_m. \tag{4.73}$$

Let $\ell(m) = \max\{\ell : a_{m\ell} \neq 0\}$ and let Q_m denote the lattice quantizer for $\Lambda_{\ell(m)}$. Note that $\ell(m)$ is the highest rate message in the equation and thus the rate of the equation that relay m wants to decode. Each relay computes:

$$\mathbf{s}_m^R = \mathsf{Re}(\alpha_m \mathbf{y}_m) + \sum_{\ell=1}^{L} \mathsf{Re}(a_{m\ell})\mathbf{d}_\ell^R - \mathsf{Im}(a_{m\ell})\mathbf{d}_\ell^I \tag{4.74}$$

$$\mathbf{s}_m^I = \mathsf{Im}(\alpha_m \mathbf{y}_m) + \sum_{\ell=1}^{L} \mathsf{Im}(a_{m\ell})\mathbf{d}_\ell^R + \mathsf{Re}(a_{m\ell})\mathbf{d}_\ell^I. \tag{4.75}$$

To get the estimates of the lattice equations, these vectors are quantized onto $\Lambda_{\ell(m)}$ modulo the coarse lattice Λ:

$$\hat{\mathbf{v}}_m^R = \left[Q_m(\mathbf{s}_m^R)\right] \mod \Lambda \tag{4.76}$$

$$\hat{\mathbf{v}}_m^I = \left[Q_m(\mathbf{s}_m^I)\right] \mod \Lambda. \tag{4.77}$$

Note that by (4.20) we have:

$$\left[Q_m(\mathbf{s}_m^R)\right] \mod \Lambda = \left[Q_m([\mathbf{s}_m^R] \mod \Lambda)\right] \mod \Lambda. \tag{4.78}$$

We now show that $[\mathbf{s}_m^R] \mod \Lambda$ is equivalent to $\mathbf{v}_m^R$ plus some noise terms.

$$
\begin{aligned}
&[\mathbf{s}_m^R] \bmod \Lambda \\
&= \left[\sum_{\ell=1}^{L} \mathsf{Re}(\alpha_m h_{m\ell})\mathsf{Re}(\mathbf{x}_\ell) - \mathsf{Im}(\alpha_m h_{m\ell})\mathsf{Im}(\mathbf{x}_\ell) + \mathsf{Re}(a_{m\ell})\mathbf{d}_\ell^R - \mathsf{Im}(a_{m\ell})\mathbf{d}_\ell^I + \mathsf{Re}(\alpha_m \mathbf{z}_m)\right] \bmod \Lambda \\
&= \left[\sum_{\ell=1}^{L} \mathsf{Re}(a_{m\ell})(\mathsf{Re}(\mathbf{x}_\ell) + \mathbf{d}_\ell^R) - \mathsf{Im}(a_{m\ell})(\mathsf{Im}(\mathbf{x}_\ell) + \mathbf{d}_\ell^I) + \mathsf{Re}\left((\alpha_m h_{m\ell} - a_{m\ell})\mathbf{x}_\ell + \alpha_m \mathbf{z}_m\right)\right] \bmod \Lambda
\end{aligned}
$$

$$
\begin{aligned}
&= \left[\sum_{\ell=1}^{L} \mathsf{Re}(a_{m\ell})([\mathbf{t}_\ell^R - \mathbf{d}_\ell^R] \bmod \Lambda + \mathbf{d}_\ell^R) - \mathsf{Im}(a_{m\ell})([\mathbf{t}_\ell^I - \mathbf{d}_\ell^I] \bmod \Lambda + \mathbf{d}_\ell^I) \cdots \right. \\
&\qquad \left. \cdots + \mathsf{Re}\Big((\alpha_m h_{m\ell} - a_{m\ell})\mathbf{x}_\ell + \alpha_m \mathbf{z}_m\Big)\right] \bmod \Lambda
\end{aligned} \tag{4.79}
$$

$$
= \left[\sum_{\ell=1}^{L} \mathsf{Re}(a_{m\ell})\mathbf{t}_\ell^R - \mathsf{Im}(a_{m\ell})\mathbf{t}_\ell^I + \mathsf{Re}\Big((\alpha_m h_{m\ell} - a_{m\ell})\mathbf{x}_\ell + \alpha_m \mathbf{z}_m\Big)\right] \bmod \Lambda \tag{4.80}
$$

$$
\begin{aligned}
&= \left[\mathbf{v}_m^R + \mathsf{Re}(\alpha_m \mathbf{z}_m) + \sum_{\ell=1}^{L} \mathsf{Re}(\alpha_m h_{m\ell} - a_{m\ell})[\mathbf{t}_\ell^R - \mathbf{d}_\ell^R] \bmod \Lambda \cdots \right. \\
&\qquad \left. \cdots - \mathsf{Im}(\alpha_m h_{m\ell} - a_{m\ell})[\mathbf{t}_\ell^I - \mathbf{d}_\ell^I] \bmod \Lambda\right] \bmod \Lambda
\end{aligned} \tag{4.81}
$$

Using similar manipulations, it can be shown that $[\mathbf{s}_m^I] \bmod \Lambda$ is equivalent to $\mathbf{v}_m^I$ plus some noise terms as well. Using Lemma 14, we can show that the channels from $\mathbf{v}_m^R$ to $\hat{\mathbf{v}}_m^R$ and from $\mathbf{v}_m^I$ to $\hat{\mathbf{v}}_m^I$ are equivalent in distribution to:

$$\hat{\mathbf{v}}_m^R = [Q_m(\mathbf{v}_m^R + \mathbf{z}_{eq,m}^R)] \bmod \Lambda \tag{4.82}$$

$$\hat{\mathbf{v}}_m^I = [Q_m(\mathbf{v}_m^I + \mathbf{z}_{eq,m}^I)] \bmod \Lambda \tag{4.83}$$

$$\mathbf{z}_{eq,m}^R = \mathsf{Re}(\alpha_m \mathbf{z}_m) + \sum_{\ell=1}^{L} \theta_{m\ell}^R \tilde{\mathbf{d}}_\ell^R - \theta_{m\ell}^I \tilde{\mathbf{d}}_\ell^I \tag{4.84}$$

$$\mathbf{z}_{eq,m}^I = \mathsf{Im}(\alpha_m \mathbf{z}_m) + \sum_{\ell=1}^{L} \theta_{m\ell}^I \tilde{\mathbf{d}}_\ell^R + \theta_{m\ell}^R \tilde{\mathbf{d}}_\ell^I \tag{4.85}$$

$$\theta_{m\ell}^R = \mathsf{Re}(\alpha_m h_{m\ell} - a_{m\ell}) \tag{4.86}$$

$$\theta_{m\ell}^I = \mathsf{Im}(\alpha_m h_{m\ell} - a_{m\ell}) \tag{4.87}$$

where each $\tilde{\mathbf{d}}_\ell^R$ and $\tilde{\mathbf{d}}_\ell^I$ is drawn independently and uniformly from $\mathcal{V}$.

Using Lemma 15 from Appendix 4.A.1, we have that the densities of both $\mathbf{z}_{eq,m}^R$ and $\mathbf{z}_{eq,m}^I$ are upper bounded (times a constant) by the density of an i.i.d. zero-mean Gaussian vector

$\mathbf{z}_m^*$ whose variance σ_m^2 approaches

$$N_{eq,m} = \frac{|\alpha_m|^2}{2} + \frac{\mathsf{SNR}}{2}\Big((\theta_{m\ell}^R)^2 + (\theta_{m\ell}^I)^2\Big) \tag{4.88}$$

$$= \frac{|\alpha_m|^2}{2} + \frac{\mathsf{SNR}}{2}\|\alpha_m \mathbf{h}_m - \mathbf{a}_m\|^2 \tag{4.89}$$

as $n \to \infty$. We also show in Appendix 4.A.2 that $\Lambda_1, \Lambda_2, \ldots, \Lambda_L$ are AWGN good. From Definition 46, this means that $\Pr(\mathbf{z}_m^* \notin \mathcal{V}_{\ell(m)})$ goes to zero exponentially in n so long as the volume-to-noise ratio satisfies $\mu(\Lambda_{\ell(m)}, P_e) > 2\pi e$. Since decoding is only in error if $\mathbf{z}_{eq,m}^R$ or $\mathbf{z}_{eq,m}^I$ leave the fundamental Voronoi region of the fine lattice $\Lambda_{\ell(m)}$ and the probability of this event is upper bounded by $\Pr(\mathbf{z}_m^* \notin \mathcal{V}_{\ell(m)})$ times a constant, we get that $\Pr(\hat{\mathbf{v}}_m^R \neq \mathbf{v}_m^R)$ and $\Pr(\hat{\mathbf{v}}_m^I \neq \mathbf{v}_m^I)$ go to zero exponentially in n.

To ensure that the probability of error goes to zero for all desired equations, we get that the volume of $\mathcal{V}_{\ell(m)}$ must satisfy

$$2\pi e < \mu(\Lambda_{\ell(m)}, P_e) = \frac{(\mathrm{Vol}(\mathcal{V}_{\ell(m)}))^{2/n}}{\sigma_m^2} \tag{4.90}$$

for all relays with $a_{m\ell} \neq 0$. If we set the volume of $\mathcal{V}_\ell$ as follows, the constraints are always met:

$$\mathrm{Vol}(\mathcal{V}_\ell) > \left(2\pi e \max_{m: a_{m\ell} \neq 0} \sigma_m^2\right)^{n/2} \tag{4.91}$$

Recall that the rate of a nested lattice code is

$$R_\ell = \frac{1}{n}\log\left(\frac{\mathrm{Vol}(\mathcal{V})}{\mathrm{Vol}(\mathcal{V}_\ell)}\right). \tag{4.92}$$

Using (4.28), we can solve for the volume of the fundamental Voronoi region of the coarse lattice:

$$\mathrm{Vol}(\mathcal{V}) = \left(\frac{\mathsf{SNR}/2}{G(\Lambda)}\right)^{n/2} \tag{4.93}$$

It follows that we can achieve any rate less than:

$$R_\ell < \min_{m: a_{m\ell} \neq 0} \frac{1}{2}\log\left(\frac{\mathsf{SNR}/2}{G(\Lambda) 2\pi e \sigma_m^2}\right) \tag{4.94}$$

while driving the probability of error to zero. Choose $\delta > 0$. Since Λ is good for quantization, for n large enough, we have that $G(\Lambda)2\pi e < (1+\delta)$. We also know that σ_m^2 converges to

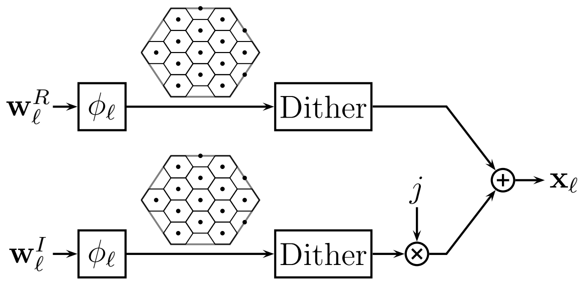

Figure 4.4: Block diagram of the compute-and-forward encoder for transmitter ℓ, $\mathcal{E}_\ell$. Messages from a finite field are mapped onto a nested lattice code, dithered, and transmitted across the channel.

$N_{eq,m}$ so for n large enough we have $\sigma_m^2 < (1+\delta)N_{eq,m}$. Finally, we get that the rate of each nested lattice code is at least:

$$\min_{m:a_{m\ell}\neq 0} \frac{1}{2}\log\left(\frac{\mathsf{SNR}}{|\alpha_m|^2 + \mathsf{SNR}\|\alpha_m\mathbf{h}_m - \mathbf{a}_m\|^2}\right) - \log(1+\delta)$$

Thus, by choosing δ small enough, we can approach the computation rates as closely as desired. □

Remark 16. Note that (4.71) differs from (4.39) by a factor of $\frac{1}{2}$. This is because (4.71) is the rate of the nested lattice code which operates in the reals and (4.39) is the rate of the message which has a real and an imaginary component.

Remark 17. The proof of Theorem 15 only requires channel state information at the relays. In particular, each relay only needs to know the channel coefficients from the transmitters to itself $\mathbf{h}_m$.

We now put all of these ingredients together to prove Theorem 13. See Figures 4.4 and 4.5 for block diagrams of the encoding and decoding process.

Encoder ℓ maps its finite field message vectors, $\mathbf{w}_\ell^R, \mathbf{w}_\ell^I \in \mathbb{F}_p^{k_\ell}$, to lattice points, $\mathbf{t}_\ell^R, \mathbf{t}_\ell^I \in \Lambda_\ell \cap \mathcal{V}$, using $\phi_\ell(\cdot)$ from Lemma 12.

$$\mathbf{t}_\ell^R = \phi_\ell(\mathbf{w}_\ell^R) \tag{4.95}$$
$$\mathbf{t}_\ell^I = \phi_\ell(\mathbf{w}_\ell^I) \tag{4.96}$$

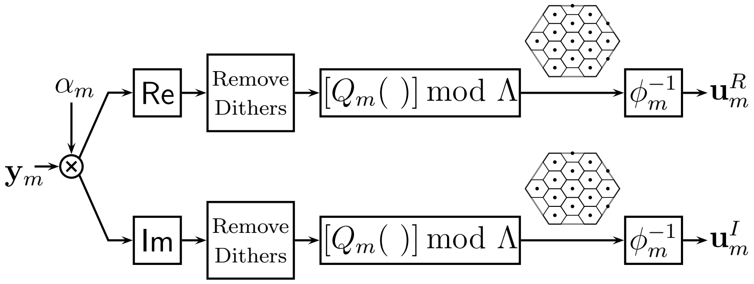

Figure 4.5: Block diagram of the compute-and-forward decoder for relay m, $\mathcal{D}_m$. The channel observation is scaled and decomposed into its real and imaginary components. The decoder then removes the dithers, quantizes onto the appropriate fine lattice, and takes the modulus over the coarse lattice. This results in an equation of lattice codewords which is then mapped into an equation of messages over the finite field.

Using Theorem 15, these lattice points are transmitted across the channel. Since the message rates are twice the lattice code rates, $r_\ell = 2R_\ell$, the relays can recover lattice equations

$$\mathbf{v}_m^R = \left[\sum_{\ell=1}^{L} \mathsf{Re}(a_{m\ell})\mathbf{t}_\ell^R - \mathsf{Im}(a_{m\ell})\mathbf{t}_\ell^I\right] \bmod \Lambda \tag{4.97}$$

$$\mathbf{v}_m^I = \left[\sum_{\ell=1}^{L} \mathsf{Im}(a_{m\ell})\mathbf{t}_\ell^R + \mathsf{Re}(a_{m\ell})\mathbf{t}_\ell^I\right] \bmod \Lambda \tag{4.98}$$

for any $\mathbf{a}_1, \dots, \mathbf{a}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$ so long as

$$r_\ell < \min_{m : a_{ml} \neq 0} \left(\log \left(\frac{\mathsf{SNR}}{|\alpha_m|^2 + \mathsf{SNR}\|\alpha_m \mathbf{h}_m - \mathbf{a}_m\|^2} \right) \right)^+$$

for some $\alpha_1, \dots, \alpha_M \in \mathbb{C}$. Finally, using ϕ_m^{-1} from Lemma 13, each relay can decode the desired linear combination of messages:

$$\mathbf{u}_m^R = \phi_m^{-1}(\mathbf{v}_m^R) = \bigoplus_{\ell=1}^{L} q_{m\ell}^R \mathbf{w}_\ell^R \oplus (-q_{m\ell}^I)\mathbf{w}_\ell^I \tag{4.99}$$

$$\mathbf{u}_m^I = \phi_m^{-1}(\mathbf{v}_m^I) = \bigoplus_{\ell=1}^{L} q_{m\ell}^I \mathbf{w}_\ell^R \oplus q_{m\ell}^R \mathbf{w}_\ell^I \tag{4.100}$$

4.3.2 Real-Valued Channels

We now restate our main results for real-valued channels.

Corollary 2. *For real-valued channel vectors $\mathbf{h}_1, \dots, \mathbf{h}_M \in \mathbb{R}^L$, the relays can recover any set of linear equations with coefficient vectors $\mathbf{a}_1, \dots, \mathbf{a}_M \in \mathbb{Z}^L$ so long as the message rates are less than the computation rate*

$$r_\ell < \min_{m:a_{ml} \neq 0} \left(R(\mathbf{h}_m, \mathbf{a}_m)\right)^+ \tag{4.101}$$

$$R(\mathbf{h}_m, \mathbf{a}_m) = \frac{1}{2} \log \left(\frac{\mathsf{SNR}}{\alpha_m^2 + \mathsf{SNR} \|\alpha_m \mathbf{h}_m - \mathbf{a}_m\|^2} \right) \tag{4.102}$$

for some choice of $\alpha_m \in \mathbb{R}$.

The coding scheme is quite similar to that for complex-valued channels: encoder ℓ maps its finite field message vector, $\mathbf{w}_\ell \in \mathbb{F}_p^{k_\ell}$, to a lattice point, $\mathbf{t}_\ell \in \Lambda_\ell \cap \mathcal{V}$, using $\phi_\ell(\cdot)$ from Lemma 12.

$$\mathbf{t}_\ell = \phi_\ell(\mathbf{w}_\ell) \tag{4.103}$$

It follows from the proof of Theorem 15 that if these lattice points are transmitted across the channel then each relay can recover a lattice equation

$$\mathbf{v}_m = \left[\sum_{\ell=1}^{L} a_{m\ell} \mathbf{t}_\ell \right] \mod \Lambda \tag{4.104}$$

for any $\mathbf{a}_1, \dots, \mathbf{a}_M \in \mathbb{Z}^L$ so long as the message rates satisfy

$$r_\ell < \min_{m:a_{ml} \neq 0} \frac{1}{2} \left(\log \left(\frac{\mathsf{SNR}}{\alpha_m^2 + \mathsf{SNR} \|\alpha_m \mathbf{h}_m - \mathbf{a}_m\|^2} \right) \right)^+$$

for some $\alpha_1, \dots, \alpha_M \in \mathbb{R}$. Finally, using ϕ_m^{-1} from Lemma 13, each relay can decode the desired linear combination of messages:

$$\mathbf{u}_m = \phi_m^{-1}(\mathbf{v}_m^I) = \bigoplus_{\ell=1}^{L} q_{m\ell} \mathbf{w}_\ell \tag{4.105}$$

Corollary 3. *For a given $\mathbf{h}_m \in \mathbb{R}^L$, $\mathbf{a}_m \in \mathbb{Z}^L$, the computation rate given in Corollary 2 is uniquely maximized by choosing α_m to be the MMSE coefficient*

$$\alpha_{_{MMSE}} = \frac{\mathsf{SNR}\ \mathbf{h}_m^T \mathbf{a}_m}{1 + \mathsf{SNR} \|\mathbf{h}_m\|^2} \tag{4.106}$$

which results in a computation rate

$$R(\mathbf{h}_m, \mathbf{a}_m) = \frac{1}{2}\log\left(\left(\|\mathbf{a}_m\|^2 - \frac{\mathsf{SNR}\ (\mathbf{h}_m^T\mathbf{a}_m)^2}{1+\mathsf{SNR}\|\mathbf{h}_m\|^2}\right)^{-1}\right)$$

4.3.3 Multi-Stage Networks

The framework developed in this section can easily be applied to AWGN networks with more than one layer of relays. Once the first layer has recovered its equations, it can just treat them as a set of messages for the second layer. The second layer simply decodes equations with coefficients that are close to the channel coefficients. This process repeats until the equations reach a destination. Since these layered equations are all linear, they can be expressed as linear equations over the original messages.

4.4 Recovering Messages

The primary goal of compute-and-forward is to enable higher achievable rates across an AWGN network. Relays decode linear equations of transmitted messages and pass them towards the destination nodes which, upon receiving enough equations, attempt to solve for their desired messages. In this section, we give sufficient conditions for recovering messages from a given set of equations.

It will be useful to represent the equations in matrix form. Let $\mathbf{Q}^R = \{q_{m\ell}^R\}$ and $\mathbf{Q}^I = \{q_{m\ell}^I\}$. Now let $\mathbf{Q}$ denote the coefficient matrix where

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}^R & -\mathbf{Q}^I \\ \mathbf{Q}^I & \mathbf{Q}^R \end{bmatrix}. \tag{4.107}$$

It is clear that we can write the linear combinations of messages $\mathbf{u}_m^R$ and $\mathbf{u}_m^I$ as

$$\begin{bmatrix} \mathbf{u}_1^R \\ \vdots \\ \mathbf{u}_M^R \\ \mathbf{u}_1^I \\ \vdots \\ \mathbf{u}_M^I \end{bmatrix} = \mathbf{Q} \begin{bmatrix} \mathbf{w}_1^R \\ \vdots \\ \mathbf{w}_M^R \\ \mathbf{w}_1^I \\ \vdots \\ \mathbf{w}_M^I \end{bmatrix}. \tag{4.108}$$

Using this representation, we can easily obtain the following two theorems for recovering messages.

Theorem 16. *Given L linear combinations of messages with coefficient matrix $\mathbf{Q} \in \mathbb{F}_p^{2L \times 2L}$, a destination can recover all messages if $\mathbf{Q}$ is full rank over $\mathbb{F}_p$.*

The proof follows by noting that if $\mathbf{Q}$ is full rank, the destination can simply apply the inverse matrix to the vector of equations in (4.108) to recover the original messages.

Recall that δ_ℓ is the unit vector with 1 in the ℓ^{th} entry and 0 elsewhere.

Theorem 17. *Given K linear combinations of messages with coefficient matrix $\mathbf{Q} \in \mathbb{F}_p^{2K \times 2L}$, a destination can recover the message from encoder ℓ if there exists a $2 \times 2K$ matrix $\mathbf{\Phi}$ such that*

$$\mathbf{\Phi}\mathbf{Q} = \left[\begin{array}{c} \delta_\ell^T \\ \delta_{\ell+M}^T \end{array} \right] \tag{4.109}$$

Again, it is clear from (4.108) that applying $\mathbf{\Phi}$ to the vector of equations allows the destination to recover $\mathbf{w}_\ell^R$ and $\mathbf{w}_\ell^I$ since they occupy positions ℓ and $\ell + M$ in the vector of messages.

Remark 18. These conditions can also be easily stated in terms of the coefficient vectors $\mathbf{a}_1, \dots, \mathbf{a}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$. First set $\mathbf{A} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_M]^T$ and let $\tilde{\mathbf{A}}$ denote its real-valued decomposition:

$$\tilde{\mathbf{A}} = \left[\begin{array}{cc} \mathsf{Re}(\mathbf{A}) & -\mathsf{Im}(\mathbf{A}) \\ \mathsf{Im}(\mathbf{A}) & \mathsf{Re}(\mathbf{A}) \end{array} \right]. \tag{4.110}$$

Now one can replace $\mathbf{Q}$ with $\tilde{\mathbf{A}}$ in Theorems 16 and 17 so long as all operations are taken modulo p.

It may be more convenient to evaluate the rank of the coefficients directly on the complex field. This is possible, given some assumptions on the equation coefficients.

Theorem 18. *The destination is given L linear equations with coefficient vectors $\mathbf{a}_1, \dots, \mathbf{a}_L \in \mathbb{F}_p^{2L \times 2L}$. Assume that magnitude of each equation coefficient is upper bounded by a constant, $|a_{m\ell}| < a_{MAX}$. Then, there exists an n_0 such that for all blocklengths $n \geq n_0$, the destination can recover all messages if $\mathbf{A} = [\mathbf{a}_1 \ \cdots \ \mathbf{a}_L]^T$ is full rank over the complex field.*

Proof. $\mathbf{A}$ is full rank over the complex field if and only if its real-valued decomposition $\tilde{\mathbf{A}}$ from (4.110) is full rank over the reals. Recall that a matrix is full rank only if its determinant is non-zero. We will now show that for sufficiently large p, if the determinant of $\tilde{\mathbf{A}}$ is non-zero over the reals it is non-zero modulo p. The determinant over $\mathbb{R}$ can be written as:

$$\det(\tilde{\mathbf{A}}) = \sum_{\sigma \in \mathcal{S}} \operatorname{sgn}(\sigma) \prod_{m=1}^{2L} \tilde{a}_{m\sigma(m)} \tag{4.111}$$

where $\mathcal{S}$ is the set of all permutations of $\{1, 2, \ldots, 2L\}$, $\text{sgn}(\sigma)$ is the signature of the permutation which is equal to 1 if it is an even permutation and -1 if it is an odd permutation, and $\tilde{a}_{m\ell}$ are the entries of $\tilde{\mathbf{A}}$. Using the bound on $a_{m\ell}$ and the fact that $|\mathcal{S}| = (2L)!$, the determinant is lower and upper bounded as follows:

$$-(2L)!(a_{\text{MAX}})^{2L} \leq \det(\tilde{\mathbf{A}}) \leq (2L)!(a_{\text{MAX}})^{2L} \tag{4.112}$$

The determinant under modulo p arithmetic can be written as:

$$\left[\sum_{\sigma \in \mathcal{S}} \text{sgn}(\sigma) \prod_{m=1}^{2L} \tilde{a}_{m\sigma(m)}\right] \mod p \tag{4.113}$$

Since the underlying field size $p \to \infty$ as $n \to \infty$, for large enough blocklength n, we can use the bounds on $\det(\tilde{\mathbf{A}})$ to show that the determinant modulo p does not wrap around zero. This immediately implies that it is zero if and only the determinant is zero over the reals. □

Remark 19. Theorem 18 can also be stated in terms of bounds on the channel coefficients. For instance, if $|h_{m\ell}| < h_{\text{MAX}}$, then we can use the bound in Lemma 11, to show that $|a_{m\ell}|$ is bounded as well. More generally, the result holds if the channel coefficients are drawn from a distribution such that $Pr\left(\cup_{m\ell}\{|h_{m\ell}| > h_{\text{MAX}}\}\right) \to 0$ as $h_{\text{MAX}} \to \infty$. In this case, we choose h_{MAX} such that this probability is very small and can be absorbed into the total probability of error for our scheme. The result follows by taking an appropriate increasing sequence of h_{MAX}.

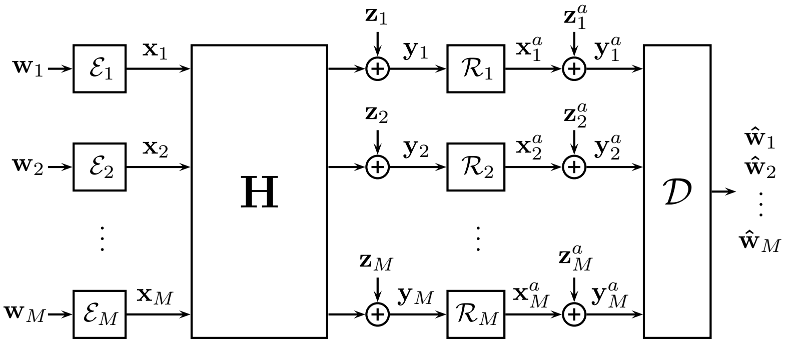

Figure 4.6: A linear relay network where compute-and-forward is beneficial.

Example 11. Consider the AWGN network in Figure 4.6. Encoder $\mathcal{E}_1, \ldots, \mathcal{E}_M$ send messages through a channel $\mathbf{H}$ to relays $\mathcal{R}_1, \ldots, \mathcal{R}_M$. Each relay has a point-to-point AWGN channel to the final decoder $\mathcal{D}$ which wants to recover all of the messages at the highest possible symmetric rate. Each channel input has power at most SNR and all noise terms are i.i.d. circularly symmetric Gaussian with variance 1. Let $\mathbf{H}$ be an $M \times M$ Hadamard matrix. (We assume that M is chosen such that a Hadamard matrix of that size exists.) Recall that a Hadamard matrix has ±1 entires such that $\mathbf{H}\mathbf{H}^T = M\mathbf{I}$.

Using Theorems 14 and 18 and setting the coefficient vectors equal to the channel vectors, $\mathbf{a}_m = \mathbf{h}_m$, compute-and-forward can achieve

$$R_{\text{COMP}} = \log\left(\frac{1}{M} + \mathsf{SNR}\right) \tag{4.114}$$

bits per channel use per user since $\mathbf{H}$ is full rank. It can be shown that decode-and-forward, amplify-and-forward, and compress-and-forward (with i.i.d. Gaussian codebooks) can achieve

$$R_{\text{DF}} = \frac{1}{M}\log\left(1 + M\mathsf{SNR}\right) \tag{4.115}$$

$$R_{\text{AF}} = R_{\text{CF}} = \log\left(1 + \mathsf{SNR}\left(\frac{\mathsf{SNR}}{M\mathsf{SNR}+1}\right)\right) \tag{4.116}$$

bits per channel use per user. Compute-and-forward is the dominant strategy except at very low SNR and it approaches the upper bound $R_{\text{UPPER}} = \log\left(1 + \mathsf{SNR}\right)$ as $\mathsf{SNR} \to \infty$. As M increases the rates of decode-and-forward, amplify-and-forward, and compress-and-forward go to 0.

Example 12. There are 3 destinations which each want to decode the message associated with their index. Assume that the underlying field size is $p = 3$. The first destination receives equations with coefficient vectors $\mathbf{a}_1 = [4\ 2\ 2]^T$ and $\mathbf{a}_2 = [5j\ \ -1\ \ -1]^T$, the second receives $\mathbf{b}_1 = [1\ 2\ 1]^T$ and $\mathbf{b}_2 = [-1\ 0\ -1]^T$, and the third receives $\mathbf{c} = [0\ 0\ 1+j]^T$. The finite field matrix representation of these equations is:

$$\mathbf{Q}_a^R = \begin{bmatrix} 1 & 2 & 2 \\ 0 & 2 & 2 \end{bmatrix} \qquad \mathbf{Q}_a^I = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \end{bmatrix} \tag{4.117}$$

$$\mathbf{Q}_b^R = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 2 \end{bmatrix} \qquad \mathbf{Q}_b^I = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.118}$$

$$\mathbf{Q}_c^R = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{Q}_c^I = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{4.119}$$

Using Theorem 17 with the following matrices, each destination can recover its desired message:

$$\mathbf{\Phi}_a = \begin{bmatrix} 2 & 1 & 2 & 1 \\ 1 & 2 & 2 & 1 \end{bmatrix} \qquad \mathbf{\Phi}_b = \begin{bmatrix} 2 & 2 & 0 & 0 \\ 0 & 0 & 2 & 2 \end{bmatrix} \tag{4.120}$$

$$\mathbf{\Phi}_c = \begin{bmatrix} 2 & 2 \\ 1 & 2 \end{bmatrix} \tag{4.121}$$

4.5 Successive Cancellation

Once a relay has recovered an equation of messages, it can subtract it from its channel observation. This results in a residual channel output from which it can extract a different equation, potentially with a higher rate than possible over the original channel. One key difference from standard applications of successive cancellation is that the relay cannot completely cancel out all channel inputs associated with the decoded equation. This is because in the first step, it only decodes an integer combination of the messages, which is often not the same as the linear combination taken by the channel.

We demonstrate an achievable region for decoding two different equations using successive cancellation at each relay. This can be easily generalized to more than two equations.

Theorem 19. *For channel vectors $\mathbf{h}_1, \ldots, \mathbf{h}_M \in \mathbb{C}^L$, the relays can first decode any set of linear equations with coefficient vectors $\mathbf{a}_1, \ldots, \mathbf{a}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$ and then any set with coefficient vectors $\mathbf{b}_1, \ldots, \mathbf{b}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$ so long as the message rates are less than the computation rates:*

$$\begin{aligned}
r_\ell &< \min\left(\min_{m:a_{m\ell}\neq 0} R_1(\mathbf{h}_m, \mathbf{a}_m), \min_{m:b_{m\ell}\neq 0} R_2(\mathbf{h}_m, \mathbf{a}_m, \mathbf{b}_m)\right) \\
R_1(\mathbf{h}_m, \mathbf{a}_m) &= \log\left(\frac{\mathsf{SNR}}{|\alpha_m|^2 + \mathsf{SNR}\|\alpha_m \mathbf{h}_m - \mathbf{a}_m\|^2}\right) \\
R_2(\mathbf{h}_m, \mathbf{a}_m, \mathbf{b}_m) &= \begin{cases} R_{B1}, & \mathbf{a}_m = \delta_i \text{ for some } i, \\ R_{B2}, & \text{otherwise.} \end{cases} \\
R_{B1} &= \log\left(\frac{\mathsf{SNR}}{|\beta_m|^2 + \mathsf{SNR}\sum_{\ell\neq i}|\beta_m h_{m\ell} - b_{m\ell}|^2}\right) \\
R_{B2} &= \log\left(\frac{\mathsf{SNR}}{|\beta_m|^2 + \mathsf{SNR}\|\beta_m \mathbf{h}_m - \tau_m \mathbf{a}_m - \mathbf{b}_m\|^2}\right)
\end{aligned}$$

for some choice of $\alpha_m, \beta_m \in \mathbb{C}$ and $\tau_m \in \mathbb{Z} + j\mathbb{Z}$.

Proof. All messages are mapped on to lattice points, dithered, and transmitted across the channel as in the proof of Theorem 13. The first set of equations can be reliably decoded using the procedure from Theorem 13 as well. Now, we condition on the event that each relay has successfully recovered the equation with coefficient vectors $\mathbf{a}_m$.

Consider the case where the first coefficient vector at relay m is a unit vector $\mathbf{a}_m = \delta_i$. This means that relay m can successfully decode the message $(\mathbf{w}_i^R, \mathbf{w}_i^I)$ from encoder i. It can then replicate the encoding process to get $\mathbf{x}_i$. Now, the relay computes

$$\mathbf{y}_m - h_{mi}\mathbf{x}_i = \sum_{\ell \neq i} h_{m\ell}\mathbf{x}_\ell + \mathbf{z}_m \tag{4.122}$$

and uses this as a channel output for Theorem 13 to get the equation with coefficient vector $\tilde{\mathbf{b}}_m$ which is equal to $\mathbf{b}_m$ except that it has 0 in the i^{th} position. It then adds $\mathbf{w}_i^R$ and $\mathbf{w}_i^I$ to the recovered equation to get $\mathbf{b}_m$.

If $\mathbf{a}_m$ is not a unit vector, the decoder has access to the lattice equations:

$$\mathbf{v}_m^R = \left[\sum_{\ell=1}^{L} \mathsf{Re}(a_{m\ell})\mathbf{t}_\ell^R - \mathsf{Im}(a_{m\ell})\mathbf{t}_\ell^I\right] \bmod \Lambda \tag{4.123}$$

$$\mathbf{v}_m^I = \left[\sum_{\ell=1}^{L} \mathsf{Im}(a_{m\ell})\mathbf{t}_\ell^R + \mathsf{Re}(a_{m\ell})\mathbf{t}_\ell^I\right] \bmod \Lambda \tag{4.124}$$

From which it computes

$$\begin{aligned}
\tilde{\mathbf{v}}_m^R &= \left[\mathbf{v}_m^R - \left(\sum_{\ell=1}^{L} \mathsf{Re}(a_{m\ell})\mathbf{d}_\ell^R - \mathsf{Im}(a_{m\ell})\mathbf{d}_\ell^I\right)\right] \bmod \Lambda \\
&= \left[\sum_{\ell=1}^{L} \mathsf{Re}(a_{m\ell}\mathbf{x}_\ell)\right] \bmod \Lambda && (4.125) \\
\tilde{\mathbf{v}}_m^I &= \left[\mathbf{v}_m^R - \left(\sum_{\ell=1}^{L} \mathsf{Im}(a_{m\ell})\mathbf{d}_\ell^R + \mathsf{Re}(a_{m\ell})\mathbf{d}_\ell^I\right)\right] \bmod \Lambda \\
&= \left[\sum_{\ell=1}^{L} \mathsf{Im}(a_{m\ell}\mathbf{x}_\ell)\right] \bmod \Lambda && (4.126)
\end{aligned}$$

$$\begin{aligned}
\tilde{\mathbf{y}}_m^R &= [\mathsf{Re}(\beta_m \mathbf{y}_m) - \mathsf{Re}(\tau_m)\tilde{\mathbf{v}}_m^R + \mathsf{Im}(\tau_m)\tilde{\mathbf{v}}_m^I] \bmod \Lambda \\
&= \left[\mathsf{Re}\left(\sum_{\ell=1}^{L} (\beta_m h_{m\ell} - \tau_m a_{m\ell})\mathbf{x}_\ell + \mathbf{z}_m\right)\right] \bmod \Lambda \\
\tilde{\mathbf{y}}_m^I &= [\mathsf{Im}(\beta_m \mathbf{y}_m) - \mathsf{Im}(\tau_m)\tilde{\mathbf{v}}_m^R - \mathsf{Re}(\tau_m)\tilde{\mathbf{v}}_m^I] \bmod \Lambda \\
&= \left[\mathsf{Im}\left(\sum_{\ell=1}^{L} (\beta_m h_{m\ell} - \tau_m a_{m\ell})\mathbf{x}_\ell + \mathbf{z}_m\right)\right] \bmod \Lambda
\end{aligned}$$

Now we can follow the steps in the proof of Theorem 15. In (4.74), replace $\mathsf{Re}(\alpha_m \mathbf{y}_m)$ with $\tilde{\mathbf{y}}_m^R$ and in (4.75), replace $\mathsf{Im}(\alpha_m \mathbf{y}_m)$ with $\tilde{\mathbf{y}}_m^I$. In all steps of the proof, we substitute $a_{m\ell}$ with $b_{m\ell}$, $\alpha_m h_{m\ell}$ with $\beta_m h_{m\ell} - \tau_m a_{m\ell}$, and, if has not already been replaced, α_m with β_m. □

Remark 20. Given, $\mathbf{a}_m$, $\mathbf{b}_m$, and τ_m, we can solve for the optimal β_m following the steps of the proof of Theorem 14.

Example 13. There are $L = 4$ transmitters and $M = 1$ relay and the channel vector is $\mathbf{h}_1 = [10 \ 10 \ 8j \ 8j]^T$. The relay wants to first decode the equation with coefficient vector $\mathbf{a}_1 = [1 \ 1 \ j \ j]^T$ and then with coefficient vector $\mathbf{b}_1 = [1 \ \ 1 \ -j \ -j]^T$. Using Theorem 19, this is possible if the message rates satisfy:

$$r_\ell < \min\left(\log\left(\frac{1}{4} + \frac{81\mathsf{SNR}}{1+4\mathsf{SNR}}\right), \log\left(\frac{1}{328} + \mathsf{SNR}\right)\right)^+$$

by using $\tau_1 = 9$ so that $\mathbf{h}_1 - \tau_1 \mathbf{a}_1 = \mathbf{b}_1$. Note that if we applied Theorem 13 to decode $\mathbf{b}_1$, we would not be able to get a positive rate.

Remark 21. As noted in Remark 14, it may be more efficient to recover an equation piecewise by recovering equations of subsets of messages and taking an appropriate linear combination of these equations. Theorem 19 is strictly better for this process than Theorem 13.

Assume there is only one relay and that it wants to recover all transmitted messages. This is the standard Gaussian multiple-access problem whose capacity region is well-known to be the set of all rate tuples $(r_1, \ldots, r_L)$ satisfying:

$$\sum_{\ell \in S} r_\ell \quad < \quad \log\left(1 + \mathsf{SNR}\sum_{\ell \in S} |h_{1\ell}|^2\right) \tag{4.127}$$

for all subsets $S \subseteq \{1, 2, \ldots, L\}$ [29, Theorem 14.3.5]. We now show that compute-and-forward includes the multiple-access capacity region as a special case. First, we consider the corner point of the capacity region associated with decoding the messages in ascending order. From Example 9, this is possible if:

$$r_1 < \log\left(1 + \frac{|h_{11}|^2\mathsf{SNR}}{1 + \mathsf{SNR}\sum_{i=2}^{L}|h_{1i}|^2}\right). \tag{4.128}$$

Using Theorem 19, the relay removes $\mathbf{x}_1$ from the channel observation to get $\sum_{\ell=2}^{L} h_{1\ell}\mathbf{x}_\ell + \mathbf{z}_1$. It then repeats the above procedure for each message in ascending order to get:

$$r_\ell < \log\left(1 + \frac{|h_{1\ell}|^2\mathsf{SNR}}{1 + \mathsf{SNR}\sum_{i=\ell+1}^{L}|h_{1i}|^2}\right) \tag{4.129}$$

This is clearly a corner point of the multiple-access capacity region. By changing the decoding order, any corner point is achievable. Note that any point on the boundary of the capacity region is achievable by time-sharing between two corner points.

4.6 Superposition

In the previous section, we considered the scenario where each relay decodes several equations, but the transmitters each use a single codebook (as in Theorem 13). However, when decoding multiple equations, it is sometimes useful to superimpose multiple codebooks. We investigate this possibility in this section.

We will assume that there are two levels A and B and that each relay wants to a recover an equation from both levels. (If it is not interested in a level, it can just set its desired coefficients to zero.)

Each encoder has two messages $\mathbf{w}_{\ell A} = (\mathbf{w}_{\ell A}^R, \mathbf{w}_{\ell A}^I)$ and $\mathbf{w}_{\ell B} = (\mathbf{w}_{\ell B}^R, \mathbf{w}_{\ell B}^I)$ with rates $r_{\ell A}$ and $r_{\ell B}$ respectively. Each relay wants to decode equations $\mathbf{u}_{mA}^R, \mathbf{u}_{mA}^I$ and $\mathbf{u}_{mB}^R, \mathbf{u}_{mB}^I$ with coefficient vectors $\mathbf{a}_m$ and $\mathbf{b}_m$ respectively. In the theorem below, we give achievable rates for this scenario by combining superposition and successive cancellation. The basic idea is to superimpose two lattice codes at each receiver (in both the real and imaginary dimensions) scaled by $\gamma_{\ell A}$ and $\gamma_{\ell B}$ to ensure that the power constraint is met.

Theorem 20. *Choose $\gamma_{\ell A}, \gamma_{\ell B}$ such that $|\gamma_{\ell A}|^2 + |\gamma_{\ell B}|^2 = 1$. For channel vectors $\mathbf{h}_1, \ldots, \mathbf{h}_M \in \mathbb{C}^L$, the relays can first decode any set of linear equations over $\mathbf{w}_{\ell A}$ with coefficient vectors $\mathbf{a}_1, \ldots, \mathbf{a}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$ and then any set of linear equations over $\mathbf{w}_{\ell B}$ with coefficient vectors*

$\mathbf{b}_1, \ldots, \mathbf{b}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$ *so long as the message rates are less than the computation rates:*

$$\begin{aligned}
&\mathbf{h}_{mA} = [\gamma_{1A} h_{m1} \ \cdots \ \gamma_{LA} h_{mL}]^T \\
&\mathbf{h}_{mB} = [\gamma_{1B} h_{m1} \ \cdots \ \gamma_{LB} h_{mL}]^T \\
&r_{\ell A} < \min_{m: a_{m\ell} \neq 0} \left(R_1(\mathbf{h}_{mA}, \mathbf{h}_{mB}, \mathbf{a}_m)\right)^+ \\
&R_1(\mathbf{h}_{mA}, \mathbf{h}_{mB}, \mathbf{a}_m) = \log\left(\frac{\mathsf{SNR}}{|\alpha_m|^2(1 + \mathsf{SNR}\|\mathbf{h}_{mB}\|^2) + \mathsf{SNR}\|\alpha_m \mathbf{h}_{mA} - \mathbf{a}_m\|^2}\right) \\
&r_{\ell B} < \min_{m: b_{m\ell} \neq 0} \left(R_2(\mathbf{h}_{mA}, \mathbf{h}_{mB}, \mathbf{a}_m, \mathbf{b}_m)\right)^+ \\
&R_2(\mathbf{h}_{mA}, \mathbf{h}_{mB}, \mathbf{a}_m, \mathbf{b}_m) = \begin{cases} R_{B1}, & \mathbf{a}_m = \delta_i \text{ for some } i, \\ R_{B2}, & \text{otherwise.} \end{cases} \\
&R_{B1} = \log\left(\frac{\mathsf{SNR}}{|\beta_m|^2(1 + \mathsf{SNR}\sum_{\ell \neq i} |\gamma_{\ell A} h_{m\ell}|^2) + \mathsf{SNR}\|\beta_m \mathbf{h}_{mB} - \mathbf{b}_m\|^2}\right) \\
&R_{B2} = \log\left(\frac{\mathsf{SNR}}{|\beta_m|^2 + \mathsf{SNR}\|\beta_m \mathbf{h}_{mA} - \tau_m \mathbf{a}_m\|^2 + \mathsf{SNR}\|\beta_m \mathbf{h}_{mB} - \mathbf{b}_m\|^2}\right)
\end{aligned}$$

for some choice of $\alpha_m, \beta_m \in \mathbb{C}$ *and* $\tau_m \in \mathbb{Z} + j\mathbb{Z}$.

Proof. Choose a set of nested lattices $\Lambda, \Lambda_{1A}, \ldots, \Lambda_{LA}, \Lambda_{1B}, \ldots, \Lambda_{LB}$ with appropriate rates. Each encoder maps its messages onto lattices and dithers them with $\mathbf{d}_{\ell A}^R, \mathbf{d}_{\ell A}^I, \mathbf{d}_{\ell B}^R, \mathbf{d}_{\ell B}^I$ generated independently and uniformly from $\mathcal{V}$:

$$\mathbf{t}_{\ell A}^R = \phi_{\ell A}(\mathbf{w}_{\ell A}^R) \qquad \mathbf{t}_{\ell A}^I = \phi_{\ell A}(\mathbf{w}_{\ell A}^I) \tag{4.130}$$

$$\mathbf{t}_{\ell B}^R = \phi_{\ell B}(\mathbf{w}_{\ell B}^R) \qquad \mathbf{t}_{\ell B}^I = \phi_{\ell B}(\mathbf{w}_{\ell B}^I) \tag{4.131}$$

$$\mathbf{x}_{\ell A} = [\mathbf{t}_{\ell A}^R - \mathbf{d}_{\ell A}^R] \bmod \Lambda + j[\mathbf{t}_{\ell A}^I - \mathbf{d}_{\ell A}^I] \bmod \Lambda \tag{4.132}$$

$$\mathbf{x}_{\ell B} = [\mathbf{t}_{\ell B}^R - \mathbf{d}_{\ell B}^R] \bmod \Lambda + j[\mathbf{t}_{\ell B}^I - \mathbf{d}_{\ell B}^I] \bmod \Lambda \tag{4.133}$$

It then combines $\mathbf{x}_{\ell A}$ and $\mathbf{x}_{\ell B}$ according to $\gamma_{\ell A}$ and $\gamma_{\ell B}$ which guarantees the power constraint is met:

$$\mathbf{x}_\ell = \gamma_{\ell A} \mathbf{x}_{\ell A} + \gamma_{\ell B} \mathbf{x}_{\ell B} \tag{4.134}$$

$$\frac{1}{n} E[\|\mathbf{x}_\ell\|^2] = (|\gamma_{\ell A}|^2 + |\gamma_{\ell B}|^2)\mathsf{SNR} = \mathsf{SNR} \tag{4.135}$$

At each receiver, we can just treat the channel output as if it came from $2L$ transmitters labelled $1A, \ldots, LA, 1B, \ldots, LB$. We can write the channel to receiver m and the desired coefficient vectors as:

$$\tilde{\mathbf{h}}_m = \begin{bmatrix} \mathbf{h}_{mA} \\ \mathbf{h}_{mB} \end{bmatrix} \qquad \tilde{\mathbf{a}}_m = \begin{bmatrix} \mathbf{a}_m \\ \mathbf{0} \end{bmatrix} \qquad \tilde{\mathbf{b}}_m = \begin{bmatrix} \mathbf{0} \\ \mathbf{b}_m \end{bmatrix}. \tag{4.136}$$

We can now directly apply Theorem 19 with $\tilde{\mathbf{h}}_m$, $\tilde{\mathbf{a}}_m$, and $\tilde{\mathbf{b}}_m$ to get the desired result. □

Remark 22. In order to focus on concepts and keep notation manageable, we have chosen to present our result as above in Theorem 20. There are several immediate extensions, including:

- More than two levels.
- Allowing a different decoding order at each relay.
- Equations spanning different levels.

Example 14. There are $L = 3$ transmitters and $M = 1$ relay and the channel vector is $\mathbf{h}_1 = [1\ 1\ \sqrt{2}]^T$. Levels A and B that use scaling coefficients $\gamma_{1A} = \gamma_{2A} = 0$, $\gamma_{1B} = \gamma_{2B} = 1$, and $\gamma_{3A} = \gamma_{3B} = 1/\sqrt{2}$. The relay wants to first decode the equation with coefficient vector $\mathbf{a}_1 = [0\ 0\ 1]^T$ from level A and then the equation with coefficient vector $\mathbf{b}_1 = [1\ 1\ 1]^T$ from level B. Using Theorem 20, this is possible if the message rates satisfy:

$$r_{3A} < \log\left(1 + \frac{\mathsf{SNR}}{1 + 3\mathsf{SNR}}\right) \tag{4.137}$$

$$r_{\ell B} < \left(\log\left(\frac{1}{3} + \mathsf{SNR}\right)\right)^+ \qquad \ell = 1, 2, 3. \tag{4.138}$$

Remark 23. It can be shown that nested lattice codes can approach the capacity region of the standard Gaussian broadcast problem. See [162] for more details.

Remark 24. For an application of this superposition scheme to a backhaul-limited cellular uplink network, see [111].

4.7 Upper Bound

In this section, we give a simple upper bound on the computation rate through a genie-aided argument. This bound does not match our achievable strategy in general and it may be possible to construct tighter outer bounds by taking into account the mismatch between the desired function and the function naturally provided by the channel.

Theorem 21. *Assume the channel between the transmitters and the relays is $p(y_1, \ldots, y_M | x_1, \ldots, x_L)$. If the relays, want equations with coefficient vectors $\mathbf{a}_1, \ldots, \mathbf{a}_M \in \{\mathbb{Z} + j\mathbb{Z}\}^L$, the message rates are upper bounded as follows:*

$$r_\ell \leq \min_{m: a_{m\ell} \neq 0} I(X_\ell; Y_m | X_1, \ldots, X_{\ell-1}, X_{\ell+1}, \ldots, X_L)$$

For the Gaussian channel model considered in this chapter, with channel vectors $\mathbf{h}_1, \ldots \mathbf{h}_M \in \mathbb{C}^L$, this specializes to:

$$r_\ell \leq \min_{m: a_{m\ell} \neq 0} \log\left(1 + |h_{m\ell}|^2 \mathsf{SNR}\right) \tag{4.139}$$

Proof. To each relay m for which $a_{m\ell} \neq 0$, we provide all messages except that from encoder ℓ as genie-aided side-information. Now, we are left with a multicasting problem from encoder ℓ to all relays with $a_{m\ell} \neq 0$. Clearly, the multicast rate is upper bounded by the lowest rate link. For the Gaussian case, it is easy to show that the mutual information expressions are maximized by the Gaussian distribution. □

Appendix 4.A Proofs

4.A.1 Upper Bound on Noise Densities

In this appendix, we demonstrate that the densities of the noise terms in Theorem 15 are upper bounded by the density of an i.i.d. Gaussian vector. The proof follows that of Lemmas 6 and 11 from [42].

Lemma 15. *Let $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \frac{1}{2}\mathbf{I}^{n \times n})$ and let $\mathbf{d}_\ell$ be independently generated according to a uniform distribution over $\mathcal{V}$. Also, let $\sigma_{\mathcal{B}}^2$ denote the second moment of an n-dimensional ball whose radius is equal to the covering radius r_{COV} of Λ and let $\mathbf{z}_\ell^*$ be independently generated according to $\mathcal{N}(\mathbf{0}, \sigma_{\mathcal{B}}^2 \mathbf{I}^{n \times n})$. Now, let*

$$\mathbf{z}_{eq} = \alpha \mathbf{z} + \sum_{\ell=1}^{L} \theta_\ell \mathbf{d}_\ell \tag{4.140}$$

where $\alpha, \theta_\ell \in \mathbb{R}$. There exists an i.i.d. Gaussian vector

$$\mathbf{z}^* = \alpha \mathbf{z} + \sum_{\ell=1}^{L} \theta_\ell \mathbf{z}_\ell^* \tag{4.141}$$

with variance σ^2 satisfying

$$\sigma^2 \leq \frac{\alpha^2}{2} + \left(\frac{r_{\scriptscriptstyle COV}}{r_{\scriptscriptstyle EFFEC}}\right)^2 \frac{\mathsf{SNR}}{2} \sum_{\ell=1}^{L} \theta_\ell^2 \tag{4.142}$$

such that the density of $\mathbf{z}_{eq}$ is upper bounded as follows:

$$f_{\mathbf{z}_{eq}}(\mathbf{z}) \leq e^{Lc(n)n} f_{\mathbf{z}^*}(\mathbf{z}) \tag{4.143}$$

$$c(n) = \ln\left(\frac{r_{\scriptscriptstyle COV}}{r_{\scriptscriptstyle EFFEC}}\right) + \frac{1}{2}\ln 2\pi e G_{\mathcal{B}}^{(n)} + \frac{1}{n} \tag{4.144}$$

where $G_{\mathcal{B}}^{(n)}$ is the normalized second moment of an n-dimensional ball and $r_{\scriptscriptstyle EFFEC}$ is the effective radius of Λ.

Proof. First, we will show that the density of $\mathbf{z}_{eq}$ is upper bounded as desired. From Lemma 11 in [42], we have that:

$$f_{\mathbf{d}_\ell}(\mathbf{z}) \leq e^{c(n)n} f_{\mathbf{z}_\ell^*}(\mathbf{z}) \tag{4.145}$$

Since $\mathbf{z}, \mathbf{d}_1, \ldots, \mathbf{d}_L$ are independent, we can write the density of $\mathbf{z}_{eq}$ as an n-dimensional convolution of the densities of its components:

$$f_{\mathbf{z}_{eq}}(\mathbf{z}) = f_{\alpha\mathbf{z}}(\mathbf{z}) * f_{\theta_1\mathbf{d}_1}(\mathbf{z}) * \cdots * f_{\theta_L\mathbf{d}_L}(\mathbf{z}) \tag{4.146}$$

Similarly, we can write the density of $\mathbf{z}^*$ as:

$$f_{\mathbf{z}^*}(\mathbf{z}) = f_{\alpha\mathbf{z}}(\mathbf{z}) * f_{\theta_1\mathbf{z}_1^*}(\mathbf{z}) * \cdots * f_{\theta_L\mathbf{z}_L^*}(\mathbf{z}) \tag{4.147}$$

Since probability densities are non-negative, we can use the upper bound in (4.145) to get:

$$f_{\alpha\mathbf{z}}(\mathbf{z}) * f_{\theta_\ell\mathbf{d}_\ell}(\mathbf{z}) \leq f_{\alpha\mathbf{z}}(\mathbf{z}) * e^{c(n)n} f_{\theta_\ell\mathbf{z}_\ell^*}(\mathbf{z}) \tag{4.148}$$

Applying this idea L times to $f_{\mathbf{z}_{eq}}(\mathbf{z})$ yields:

$$f_{\mathbf{z}_{eq}}(\mathbf{z}) \leq e^{Lc(n)n} f_{\mathbf{z}^*}(\mathbf{z}) \tag{4.149}$$

We must now upper bound the variance of $\mathbf{z}^*$. By Definition 42, $\mathrm{Vol}(\mathcal{B}(r_{\text{EFFEC}})) = \mathrm{Vol}(\mathcal{V})$. Recall that a ball has the smallest second moment for a given volume. Let $\mathbf{b}$ be generated according to the uniform distribution over $\mathcal{B}(r_{\text{COV}})$. It follows that

$$\frac{\mathsf{SNR}}{2} = \frac{1}{n} E\left[\|\mathbf{d}_\ell\|^2\right] \tag{4.150}$$

$$\geq \frac{1}{n} E\left[\left\|\frac{r_{\text{EFFEC}}}{r_{\text{COV}}}\mathbf{b}\right\|^2\right] = \left(\frac{r_{\text{EFFEC}}}{r_{\text{COV}}}\right)^2 \sigma_{\mathcal{B}}^2 \tag{4.151}$$

Finally, we get:

$$\sigma^2 = \frac{1}{n} E\left[\|\alpha \mathbf{z}\|^2\right] + \frac{1}{n}\sum_{\ell=1}^{L} E\left[\|\theta_\ell \mathbf{z}_\ell^*\|^2\right] \tag{4.152}$$

$$= \frac{\alpha^2}{2} + \sigma_{\mathcal{B}}^2 \sum_{\ell=1}^{L} \theta_\ell^2 \tag{4.153}$$

$$\leq \frac{\alpha^2}{2} + \left(\frac{r_{\text{COV}}}{r_{\text{EFFEC}}}\right)^2 \frac{\mathsf{SNR}}{2}\sum_{\ell=1}^{L} \theta_\ell^2 \tag{4.154}$$

□

Since and $\frac{r_{\text{COV}}}{r_{\text{EFFEC}}} \to 1$ and $G_{\mathcal{B}}^{(n)} \to \frac{1}{2\pi e}$ as $n \to \infty$, $c(n) \to 0$ as $n \to \infty$. AWGN good lattices have a positive error exponent for i.i.d. Gaussian noise with variance smaller than the second moment of the lattice so this means that the probability of error can be driven to zero as the blocklength increases.

4.A.2 Fine Lattices are AWGN Good

We now show that the fine lattices can recover from i.i.d. Gaussian noise.

Lemma 16. *$\Lambda_1, \Lambda_2, \ldots, \Lambda_L$ are AWGN good with probability that goes to 1 as $n \to \infty$ so long as $\frac{n}{p} \to 0$.*

Proof. Let $\mathbf{z}_\ell$ denote an i.i.d. Gaussian vector with zero-mean and any variance σ_ℓ^2 such that the volume-to-noise ratio for Λ_ℓ is greater than $2\pi e$. Consider the following channel from $\mathbf{x}_\ell \in \mathcal{V}$ to $\hat{\mathbf{x}}_\ell \in \mathcal{V}$:

$$\hat{\mathbf{x}}_\ell = [Q_{\Lambda_\ell}(\mathbf{x}_\ell + \mathbf{z}_\ell)] \mod \Lambda \tag{4.155}$$

and let $P_{e,\ell} = Pr(\hat{\mathbf{x}}_\ell \neq \mathbf{x}_\ell)$. In Appendix B of [42], it is shown that the random coding error exponent for this channel (with $\mathbf{x}_\ell$ generated uniformly over $\mathcal{V}$) is equal to the Poltyrev exponent. This means that $P_{e,\ell}$ decreases exponentially with n for volume-to-noise ratio greater than $2\pi e$. Appendix C of [42] shows that the same performance is possible if $\mathbf{x}_\ell$ is drawn according to a uniform distribution over $\{p^{-1}\Lambda\} \cap \mathcal{V}$ and $\frac{n}{p} \to 0$.

From Lemma 10, we know that the marginal distribution of each element of $\Lambda_\ell \cap \mathcal{V}$ is uniform over $\{p^{-1}\Lambda\} \cap \mathcal{V}$. Furthermore, all points in the set $\Lambda_\ell \cap \mathcal{V}$ are pairwise independent. This is all that is required to apply the union bound and obtain the same performance as i.i.d. inputs over $\{p^{-1}\Lambda\} \cap \mathcal{V}$ in terms of the error exponent.

Thus, the probability that Λ_ℓ is AWGN good (with the Poltyrev error exponent) goes to 1 as $n \to \infty$. It follows from a simple union bound that $\Lambda_1, \ldots, \Lambda_L$ are simultaneously AWGN good with high probability as $n \to \infty$. □

4.A.3 Fixed Dithers

We now show that there exist fixed dithers that are appropriate for our coding scheme. We begin by showing that with high probability over the dither vectors, the power constraint can be met (since in the proof of Theorem 15 this is only done in expectation).

Choose $\delta > 0$. Scale the second moment of the coarse lattice to be $\sigma_\Lambda^2 = \frac{\mathsf{SNR}}{2} - \delta \left(\frac{r_{\text{EFFEC}}}{r_{\text{COV}}}\right)^2$ and consider the real part of the channel input from encoder ℓ, $[\mathbf{t}_\ell^R - \mathbf{d}_\ell^R] \bmod \Lambda$. By Lemma 14, this is uniformly distributed over $\mathcal{V}$. Let $\mathbf{d}$ be a vector drawn from the uniform distribution over $\mathcal{V}$ and $\mathbf{z}^*$ be an i.i.d. Gaussian vector with mean 0 and variance $\sigma^2 = \left(\frac{r_{\text{COV}}}{r_{\text{EFFEC}}}\right)^2 \sigma_\Lambda^2$. Since $\mathbf{z}^*$ is i.i.d. Gaussian, by the weak law of large numbers

$$P\left(\left|\frac{1}{n}\|\mathbf{z}^*\|^2 - \sigma^2\right| > \delta\right) \leq \frac{2\sigma^4}{\delta n}. \tag{4.156}$$

From Lemma 11 in [42], we have that:

$$f_{\mathbf{d}}(\mathbf{z}) \leq e^{c(n)n} f_{\mathbf{z}^*}(\mathbf{z}). \tag{4.157}$$

with $c(n)$ as in (4.144). Note that $\sigma^2 = \left(\frac{r_{\text{COV}}}{r_{\text{EFFEC}}}\right)^2 \frac{\mathsf{SNR}}{2} - \delta$ so the probability that $\mathbf{d}$ violates the power constraint can be upper bounded in terms of σ^2:

$$P\left(\frac{1}{n}\|\mathbf{d}\|^2 > \frac{\mathsf{SNR}}{2}\right) \leq P\left(\left|\frac{1}{n}\|\mathbf{d}\|^2 - \sigma^2\right| > \delta\right) \tag{4.158}$$

$$\leq e^{c(n)n} \frac{2\sigma^4}{\delta n} \tag{4.159}$$

This goes to 0 as n goes to infinity so, by the union bound, all channel inputs obey the power constraint with high probability. Since Λ is covering good and the rates are continuous in the second moment of the coarse lattice, we can choose a decreasing sequence of δ that approaches the rates achieved when $\sigma_\Lambda^2 = \frac{\mathsf{SNR}}{2}$.

We note that the probability of decoding error in Theorem 15 goes to zero as n increases when using random dithers. Taking a union bound over this error event and the event that any dither exceeds a power constraint, we get that for n large enough there is (at least) one good fixed set of dither vectors.

Chapter 5

Communication Network Applications

The two previous chapters showed how to send functions of messages efficiently through the use of computation codes. In a communication network, we are often only interested in relaying messages to their destinations, not in computing functions. However, only the destinations need to learn their messages, all other relay nodes can follow any strategy that helps achieve this objective. In a wireless network, the usual approach is to use channel codes so that relays can send messages to one another. This forms a bit pipe network on top of which the messages can be routed from source to destination. Unfortunately, establishing these bit pipes can be quite costly as the channel codes must treat interference as noise. Our approach is to apply compute-and-forward and and have relays decode functions of the messages with coefficients chosen to match the fading realization. These equations are sent towards the destinations which can solve for the original messages after collecting enough equations.

One key advantage of compute-and-forward is that it works with the noisy linear combinations provided by the wireless medium. Another is that it is a digital strategy: relays send out message vectors over a finite field and recover equations over the same field. The same cannot be said for other cooperative strategies such as amplify-and-forward and compress-and-forward. These analog strategies work directly with the received waveforms to extract gains. While they perform quite well in the high SNR regime, analog strategies suffer as noise builds up with each retransmission. For larger networks at moderate SNR values, it is better to remove the noise at each stage using compute-and-forward. Moreover, from an engineering standpoint, we prefer modular solutions that fit into the network protocol stack. The current network architecture relies on a digital representation in bits at all layers above the physical layer. With compute-and-forward, we can nearly retain this representation by working with equations of bits at the physical layer. Analog strategies do not seem to admit such a representation and might require a cross-layer architecture. Thus, we would potentially have to sacrifices the advantages of modularity to improve the end-to-end throughput [70].

We will explore several different network scenarios and compare the performance of compute-and-forward to other cooperative strategies. First, we consider a cellular uplink system with cooperating cell-sites. Next, we look at a two-user distributed MIMO system. Afterwards, we show how to apply compute-and-forward for network coding over interfering links.

5.1 Backhaul Constrained Cellular Uplink

A major challenge in cellular communication is efficient interference management for the uplink channel. Each basestation in a cellular deployment receives signals from users within its cell as well as interference in the form of signals from users in adjacent cell sites. Classical communication schemes attempt to mitigate the effects of this interference either by orthogonalization or interference averaging. Significant gains are possible over these strategies by allowing some degree of cooperation between the basestations, which is known as *joint multiple cell-site processing.*

Wyner's original paper on cellular systems considers full cooperation between the basestations and finds the capacity in the nonfading, symmetric case [157]. Somekh et al. extended this result to the flat fading case and demonstrated that out-of-cell interference can be completely eliminated by joint processing [146]. Both of these works assume that the "backhaul capacity" from each cell-site to the remote central processor (RCP) is infinite. For most networks, this assumption may not hold; however, it does establish a promising target for the limited backhaul case. Recent work by Sanderovich et al. gave an achievable strategy based on compress-and-forward (also known as estimate-and-forward [33]) and decode-and-forward for the limited backhaul case [131]. They also showed that compress-and-forward carries an additional benefit: basestations can be oblivious to the codebooks of neighboring cell-sites. An overview of this work can be found in [138]. Marsch and Fettweis have also recently extended this strategy to include superposition coding [92].

We will use lattice computation codes to allow the cell-sites to decode equations of codewords and forward these to the RCP, which, given a full rank system of such equations, can recover the original messages.[1] This strategy strikes a balance between noise removal at the basestations and joint decoding at the RCP. Recall that in compute-and-forward, each receiver decodes an equation with integer coefficients that approximate the channel coefficients. The remainder from this integer approximation acts like additional noise at the receiver. Here, we will use superposition to reduce this approximation penalty for Wyner's cellular model. In our scheme, every other user employs some of its power towards a private message. By changing the power allocation, we can effectively steer the interference parameter to increase the compute-and-forward rate.

[1] The material in this section originally appeared in [111].

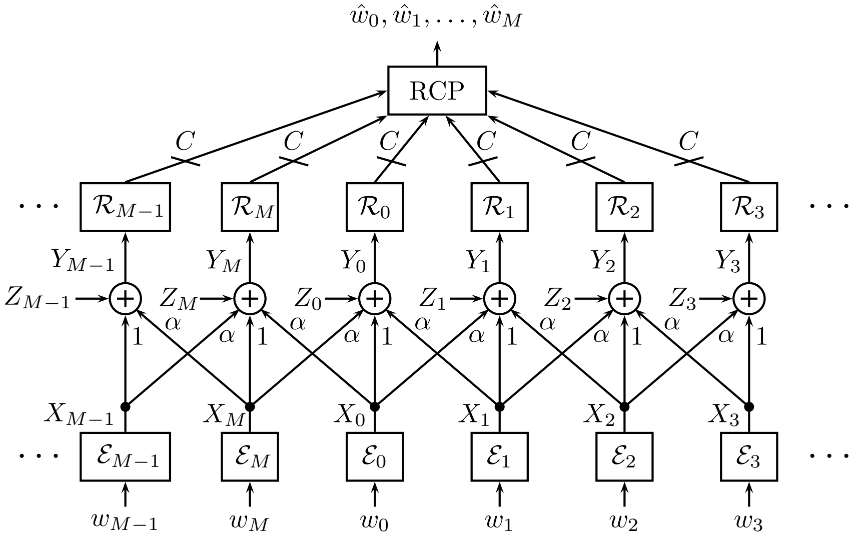

Figure 5.1: Idealized cellular uplink network from users to cell-sites to the remote central processor (RCP).

5.1.1 Problem Statement: Wyner's Cellular Model

We will focus on the model introduced by Sanderovich, Somekh and Shamai in [131] (see Figure 5.1) which is a variation on the classical Wyner model [157]. Consider an idealized cellular uplink network where M cell-sites are equally spaced on a circle. We assume that only one user is active in a cell per time slot and that cell-sites only see interference from a neighboring cells' users. The users are given n channel uses to reliably convey their messages. The complex signal $X_m[i]$ is transmitted by the m^{th} user at time i and is power-limited in the usual way, $\frac{1}{n}\sum_{i=1}^{n}|X_m[i]|^2 \leq P$. The signal seen by the m^{th} cell-site at time i is:

$$Y_m[i] = X_m[i] + \alpha(X_{[m-1]_M}[i] + X_{[m+1]_M}[i]) + Z_m[i]$$

where $[m]_M \triangleq m \mod M$, $\alpha \in [0,1]$ is the inter-cell interference level, and $Z_m[i]$ is i.i.d. $\mathcal{CN}(0,N)$ noise. For notational convenience, we set $\mathsf{SNR} \triangleq \frac{P}{N}$ to be the signal-to-noise ratio per user. Unlike the Wyner model, where unlimited backhaul is available, in the Sanderovich et al. model we use here, the cell-sites have a lossless backhaul link to the RCP with rate B bits per channel use [131]. We focus on the symmetric case where each user targets the same rate.

Each user is equipped with an encoding function, $\mathcal{E}_m : \{1, 2, \ldots, 2^{nR}\}$, that maps its messages w_m into channel input X_m^n. Each cell-site has a relaying function, $\mathcal{R}_m : \mathbb{C}^n \rightarrow \{1, 2, \ldots, 2^{nB}\}$, that it uses to communicate its observation Y_m^n to the RCP. The RCP has a decoding function, $\mathcal{D} : \{1, 2, \ldots, 2^{nB}\}^M \rightarrow \{1, 2, \ldots, 2^{nR}\}^M$, to make estimates $\hat{w}_m$ of the original messages. We say that a *per user rate* R is achievable if each user's message can be decoded by the RCP with vanishing probability of error (in the blocklength n).

In matrix form, the channel from users to cell-sites is:

$$\mathbf{H} = \{h_{m\ell}\}, \quad h_{m\ell} = \begin{cases} 1 & \text{if } \ell = m, \\ \alpha & \text{if } \ell = [m+1]_M \text{ or } [m-1]_M, \\ 0 & \text{otherwise.} \end{cases}$$

5.1.2 Upper Bound: Cut-Set

We will use a simple cut-set bound to upper bound the capacity. Since we are only interested in symmetric rate tuples and the channel is symmetric, the sum rate is upper bounded by the dominant cut set bound. Recall that for a MIMO channel with independent (and equal power) channel inputs the capacity is given by:

$$C_{\text{MIMO}} = \log\det\left(\mathbf{I} + \mathsf{SNR}\ \mathbf{H}\mathbf{H}^*\right) \tag{5.1}$$

where $\mathbf{H}^*$ denotes the Hermetian transpose of $\mathbf{H}$ [151]. Clearly, the rate of information flow between the cell-sites and the RCP is upper bounded by MB. Also, if the cell-sites could cooperate freely, then the channel from the users becomes a MIMO channel. Thus, our cut-set bound is the minimum between C_{MIMO} and MB, normalized by the number of users:

$$C_{\text{UPPER}} = \min\left(B, \frac{1}{M}\log\det\left(\mathbf{I} + \mathsf{SNR}\ \mathbf{H}\mathbf{H}^*\right)\right) \tag{5.2}$$

This upper bound reveals two potential information bottlenecks. One is the MIMO behavior of the channel and the other is the finite backhaul. An ideal scheme for this problem will cope with both bottlenecks simultaneously.

For the special circularly symmetric matrix at hand, Wyner showed in [157] that as M tends to infinity we get:

$$C_{\text{MIMO}} = \int_0^1 \log\left(1 + \mathsf{SNR}(1 + 2\alpha\cos 2\pi\theta)^2\right) d\theta. \tag{5.3}$$

5.1.3 Classical Random Coding Strategies

We now review the performance of two known strategies, compress-and-forward (CF) and decode-and-forward (DF).

5.1.3.1 Compress-and-Forward

Sanderovich et al. showed in [131] that compress-and-forward achieves the following rate per user:

$$R_{\text{CF}} = \frac{1}{M} \max_{0 \leq r} \min_{\mathcal{I} \subseteq \{1,\ldots,M\}} \Big(|\mathcal{I}|(B-r) + \log\det\big(\mathbf{I} + \mathsf{SNR}(1-2^{-r})\mathbf{H}_{\mathcal{I}^C}\mathbf{H}^*_{\mathcal{I}^C}\big)\Big) \tag{5.4}$$

where $\mathbf{H}_{\mathcal{I}^C}$ is the submatrix of $\mathbf{H}$ that includes only rows in the subset $\mathcal{I}^C$ (which is the complement of the subset $\mathcal{I}$). This rate in the asymptotic case, where $M \to \infty$ is [131]:

$$R_{\text{CF}} = F(r^*) \text{ where } r^* \text{ is the solution of } F(r^*) = B - r^*,$$
$$F(r) = \int_0^1 \log\big(1 + \mathsf{SNR}(1-2^{-r})(1+2\alpha\cos 2\pi\theta)^2\big)\, d\theta.$$

The approach of [131] is oblivious in terms of assuming no decoding at cell-sites and is optimal if B or SNR tends to infinity. As B tends to infinity, the rate converges to the infinite backhaul capacity case given by Equation (5.3). As SNR tends to infinity, this rate converges to B. For moderate SNR and B, compress-and-forward roughly follows the outer cut-set bound (with α) but never touches it. This gap from the outer bound occurs since the scheme forwards the entire channel output, including the noise.

5.1.3.2 Decode-and-Forward

Sanderovich et al. also derive the performance of decode-and-forward in [131]. For the no fading case, the channel to each cell-site is equivalent to a three input multiple-access channel, so that the decode-and-forward rate is:

$$R_1 = \log\left(1 + \frac{\mathsf{SNR}}{1+2\alpha^2\mathsf{SNR}}\right) \tag{5.5}$$

$$R_2 = \min\left(\frac{1}{2}\log\big(1+2\alpha^2\mathsf{SNR}\big), \frac{1}{3}\log\big(1+(1+2\alpha^2)\mathsf{SNR}\big)\right) \tag{5.6}$$

$$R_{\text{DF}} = \min\Big(\max(R_1, R_2), B\Big) \tag{5.7}$$

R_1 models decoding when the other signals are treated as noise, while R_2 assumes full reliable decoding of all three data streams received at the cell-site. Note that there is no joint processing gain. When the backhaul capacity B is small compared to the rates between the users and the cell-sites, or when the interference level is low, decode-and-forward is optimal.

5.1.4 Compute-and-Forward

Each cell-site sees a linear combination of its user's transmission as well as the transmissions of two neighboring users plus noise. Using the lattice compute-and-forward strategy from Chapter 4, we can remove the noise and send the error-free linear combination of codewords to the RCP. If the RCP can collect enough linear equations to form a full rank system, it can recover the original messages.

Due to the symmetry in the inter-cell interference, we can have each cell-site use the same recovery coefficients for its linear function and guarantee that the resulting matrix is full rank. We choose our integer decoding coefficients $a_{m\ell}$ to match the structure of the channel coefficients:

$$a_{m\ell} = \begin{cases} b_1 & \text{if } \ell = m, \\ b_2 & \text{if } \ell = [m+1]_M \text{ or } [m-1]_M, \\ 0 & \text{otherwise.} \end{cases}$$

where $b_1, b_2 \in \mathbb{Z},\ b_1 \neq 0$. The diagonal structure of the channel matrix guarantees the full rank condition.

Theorem 22. *The following rate per user is achievable with backhaul rate B using the compute-and-forward strategy:*

$$R = \min(R_{COMP}, B) \tag{5.8}$$

$$R_{COMP} = \max_{b_1, b_2 \in \mathcal{A}} -\log\left(b_1^2 + 2b_2^2 - \frac{\mathsf{SNR}(b_1 + 2\alpha b_2)^2}{1 + \mathsf{SNR}(1 + 2\alpha^2)}\right) \tag{5.9}$$

$$\mathcal{A} = \{(b_1, b_2) : b_1, b_2 \in \mathbb{Z},\ b_1 \neq 0, b_1^2 + 2b_2^2 \leq 1 + \mathsf{SNR}(1 + 2\alpha^2)\}. \tag{5.10}$$

The restriction of b_1 and b_2 to $\mathcal{A}$ makes it possible to evaluate the rate expression exactly since $\mathcal{A}$ is a finite set. Note that all (b_1, b_2) pairs outside $\mathcal{A}$ trivially give zero rate by Lemma 11.

In Figure 5.2, we plot the performance of Theorem 22 against the interference strength $\alpha \in [0, 1]$ in the infinite backhaul case, $B = \infty$. We also plot the upper bound from (5.3) and the decode-and-forward achievable rate from (5.7). (Note that since the backhaul rate is infinite, the compress-and-forward rate matches the upper bound.) Each plot is for a fixed value of SNR from 5dB at the top to 10, 20 and finally 40dB at the bottom. As we increase the SNR, the size of $\mathcal{A}$ increases and we can do a better job of approximating α with $\frac{b_2}{b_1}$. When α is exactly captured by the rational approximation $\frac{b_2}{b_1}$, the compute-and-forward rate is at a local maximum. Conversely, when α is poorly captured, the compute-and-forward is at a local minimum. This means that at higher values of SNR, the performance fluctuates more with α. This behavior is somewhat surprising as decode-and-forward, compress-and-

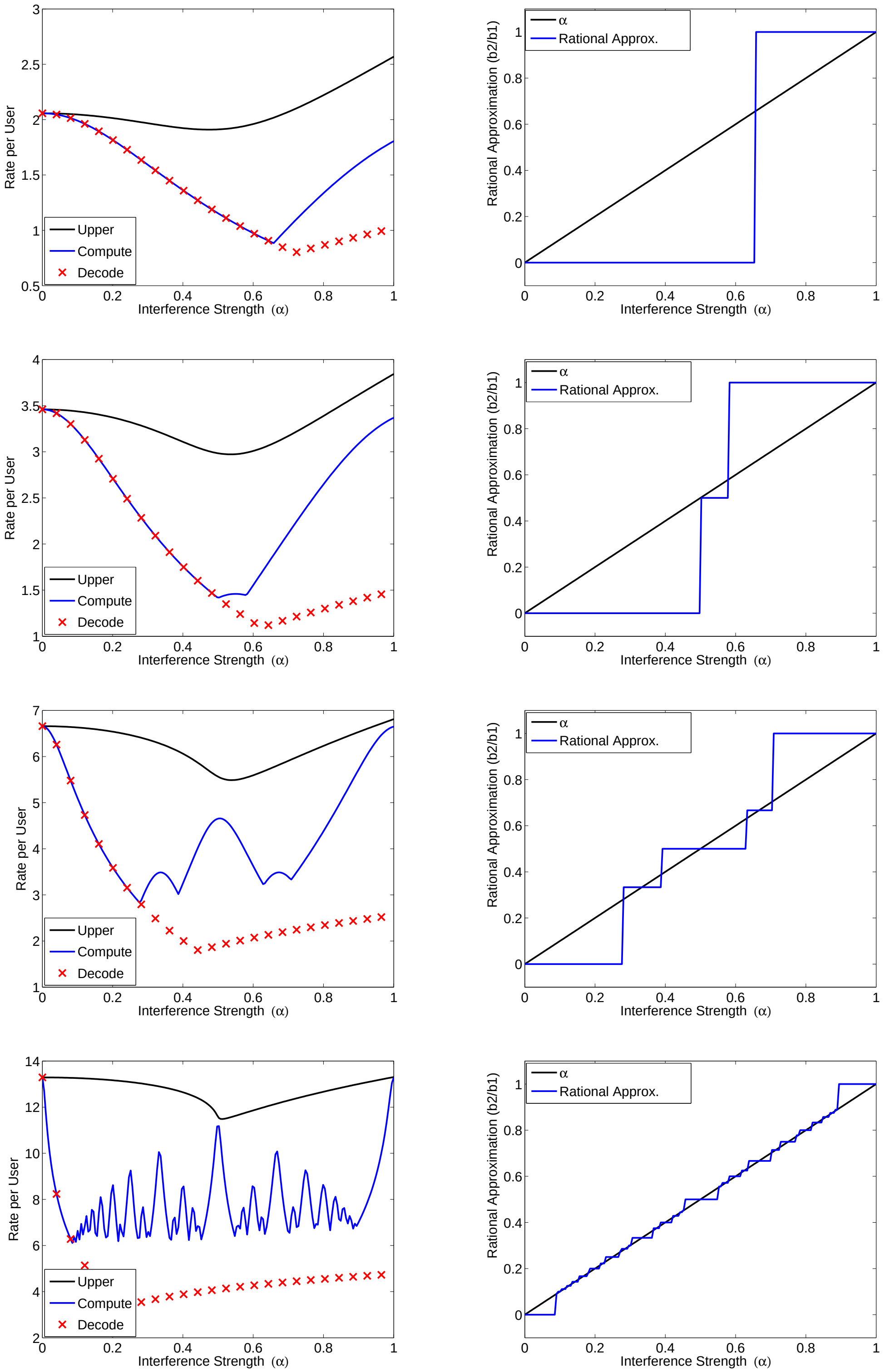

Figure 5.2: Compute-and-forward performance at SNR $= 5, 10, 20, 40$dB (left) along with the approximation of α by integer decoding coefficients b_1 and b_2 (right).

forward, and the upper bound are much smoother with α. Next, we develop a superposition strategy that can smooth out these fluctuations to a degree and achieve higher rates.

Remark 25. From Figure 5.2, it seems as if compute-and-forward approaches the upper bound as the SNR increases for $\alpha = \frac{1}{2}$. It would be interesting to prove that it does indeed meet the inner bound exactly as $\mathsf{SNR} \to \infty$.

5.1.5 Superposition Strategy

While the compute-and-forward performance in Figure 5.2 is promising, the significant variations due to integer effects do not seem fundamental to the problem. That is, the rate should not sharply degrade with only a small change in α. We now develop a superposition strategy that decreases the effective interference seen by the even users and increases the effective interference seen by the odd users. With this strategy, we can avoid the local minima in α. Usually, superposition coding is used to provide different messages (or rates) to users of varying quality. Here, we are using superposition to adjust the interference structure of the problem itself. The resulting performance boost to compute-and-forward is a new phenomenon: traditional strategies would not benefit at all from such a scheme.

Our strategy is a special case of Theorem 20. Each odd-numbered user is allocated power νP for its two lattice codewords (one for the real part of the channel and the other for the imaginary part) for some $\nu \in [0, 1]$. Each even-numbered user will use power $(1 + \delta)P$ for its lattice codewords where $\delta \in [0, 1 - \nu]$ and power $(1 - \nu - \delta)P$ for the superimposed codewords. In the language of Theorem 20, this translates to:

$$\gamma_{\ell A} = \begin{cases} 0 & \text{if } \ell \text{ is odd,} \\ \sqrt{1 - \nu - \delta} & \text{if } \ell \text{ is even.} \end{cases} \tag{5.11}$$

$$\gamma_{\ell B} = \begin{cases} \sqrt{\nu} & \text{if } \ell \text{ is odd,} \\ \sqrt{1 + \delta} & \text{if } \ell \text{ is even.} \end{cases} \tag{5.12}$$

Note that the even users utilize more power than the odd users. To satisfy the power constraint, we run the scheme in this fashion for half the channel uses and then switch the role of the odd and even users.

The even-numbered cell-sites always decode the superimposed codeword $\mathbf{a}_m = \delta_m$ (at rate R_S^E). Upon successful decoding, they cancel out $\mathbf{x}_{mA}$ and use the standard compute-and-forward strategy on the resulting channel output (at rate R_C^E). The effective channel gains for compute-and-forward after this successive cancellation step are:

$$h_{m\ell}^{\text{EVEN}} = \begin{cases} \sqrt{1+\delta} & \text{if } \ell = m, \\ \alpha\sqrt{\nu} & \text{if } \ell = [m+1]_M \text{ or } [m-1]_M, \\ 0 & \text{otherwise.} \end{cases}$$

The odd-numbered cell-sites either decode and remove both superimposed messages from their two neighbors (at rate R_S^{DO}) or treat them as additional noise. They then use compute-and-forward on the output (at rate R_C^{DO} or rate R_C^{IO}) which has the following effective channel coefficients:

$$h_{m\ell}^{\text{ODD}} = \begin{cases} \sqrt{\nu} & \text{if } \ell = m, \\ \alpha\sqrt{1+\delta} & \text{if } \ell = [m+1]_M \text{ or } [m-1]_M, \\ 0 & \text{otherwise.} \end{cases}$$

The odd and even cell-sites need to maximize over integer coefficients to best fit the channel as before.

Theorem 23. *The following rate is achievable using the superposition strategy as $M \to \infty$:*

$$
\begin{aligned}
R &= \min\left(\max_{\nu\in[0,1]}\max_{\delta\in[0,1-\nu]}\max\left(R^D, R^I\right),\ C\right)\\
\mathcal{A} &= \left\{(b_1,b_2) : b_1, b_2 \in \mathbb{Z},\ b_1 \neq 0, b_1^2 + 2b_2^2 \leq 1 + \mathsf{SNR}(1+2\alpha^2)(1+\delta)\right\}\\
R^D &= \min\left(\frac{R_S^E}{2}, \frac{R_S^{DO}}{2}\right) + \min\left(\max_{b_1,b_2\in\mathcal{A}} R_C^E,\ \max_{b_1,b_2\in\mathcal{A}} R_C^{DO}\right)\\
R_S^E &= \log\left(1 + \frac{\mathsf{SNR}(1-\nu-\delta)}{1+\mathsf{SNR}(1+\delta+2\alpha^2\nu)}\right)\\
R_S^{DO} &= \frac{1}{2}\log\left(1 + \frac{2\alpha^2\mathsf{SNR}(1-\nu-\delta)}{1+\mathsf{SNR}(\nu+2\alpha^2(1+\delta))}\right)\\
R_C^E &= \log\left(\left(b_1^2 + 2b_2^2 - \frac{\mathsf{SNR}(\sqrt{1+\delta}b_1 + 2\alpha\sqrt{\nu}b_2)^2}{1+\mathsf{SNR}(1+\delta+2\alpha^2\nu)}\right)^{-1}\right)\\
R_C^{DO} &= \log\left(\left(b_1^2 + 2b_2^2 - \frac{\mathsf{SNR}(\sqrt{\nu}b_1 + 2\alpha\sqrt{1+\delta}b_2)^2}{1+\mathsf{SNR}(\nu+2\alpha^2(1-\nu))}\right)^{-1}\right)\\
R^I &= \frac{R_S^E}{2} + \min\left\{\max_{b_1,b_2\in\mathcal{A}} R_C^E,\ \max_{b_1,b_2\in\mathcal{A}} R_C^{IO}\right\}\\
R_C^{IO} &= \log\left(\left(b_1^2 + 2b_2^2 - \frac{\mathsf{SNR}(\sqrt{\nu}b_1 + 2\alpha\sqrt{1+\delta}b_2)^2}{1+\mathsf{SNR}(\nu+2\alpha^2(1-\nu))}\right)^{-1}\right)
\end{aligned}
$$

where R_S^E is the rate at which the even cell-sites can recover the superimposed message, R_S^{DO} is the rate at which the odd cell-sites can recover the superimposed message, R_C^E is the compute-and-forward rate for the even cell-sites, R_C^{DO} is the compute-and-forward rate for the odd cell-sites when they first decode the superimposed message, and R_C^{IO} is the compute-and-forward rate for the odd cell-sites when they ignore the superimposed message as noise.

Remark 26. Assume that the first and last transmitter are inactive. It can then be shown that through a filterbank argument we can solve for the original messages from the odd and even equations so long as $b_1 \neq 0$ for both equations [102]. By a timesharing argument, we can thus achieve $\frac{M-2}{M}$ of the desired rate and, as $M \to \infty$, we can achieve the rate in Theorem 23. If we add a full rank requirement to the theorem statement, then we can give achievable rates at finite M as well.

In Figure 5.3, we plot the performance of the superposition compute-and-forward strategy at 25dB with infinite backhaul capacity, $B = \infty$, versus the regular compute-and-forward strategy, decode-and-forward, and the upper bound in (5.3). As discussed earlier, the basic compute-and-forward strategy varies significantly with α with local minima at values of α

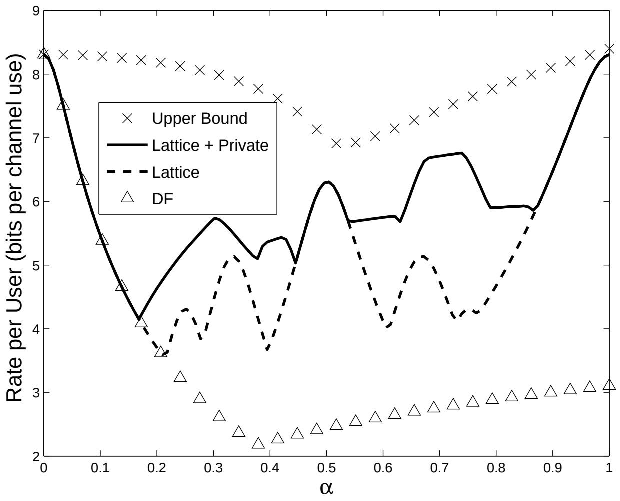

Figure 5.3: $\mathsf{SNR} = 25$dB. Achievable rates per user as a function of the inter-cell interference levels α, for infinite backhaul capacity. In this case, compress-and-forward matches the upper bound.

that are hard to approximate with small integers. The superposition strategy does a good job of filling in these "valleys" and has higher rates for the middle range of α.

5.1.6 Performance Comparisons

We now compare the rates achieved by the superposition strategy, compute-and-forward, decode-and-forward, and compress-and-forward in the asymptotic $M \to \infty$ regime with finite backhaul rate B. In Figure 5.4, we have plotted these schemes for three different values of SNR and B. For most of the range of α, superposition compute-and-forward achieves the highest rate. Note that if we were to rely on regular compute-and-forward, the rate "valleys" would often pull the performance below that achieved by compress-and-forward. Also, observe that the performance of compute-and-forward matches that of decode-and-forward for low values of α. In this regime, the best equation to decode is the unit vector which amounts to treating interference as noise.

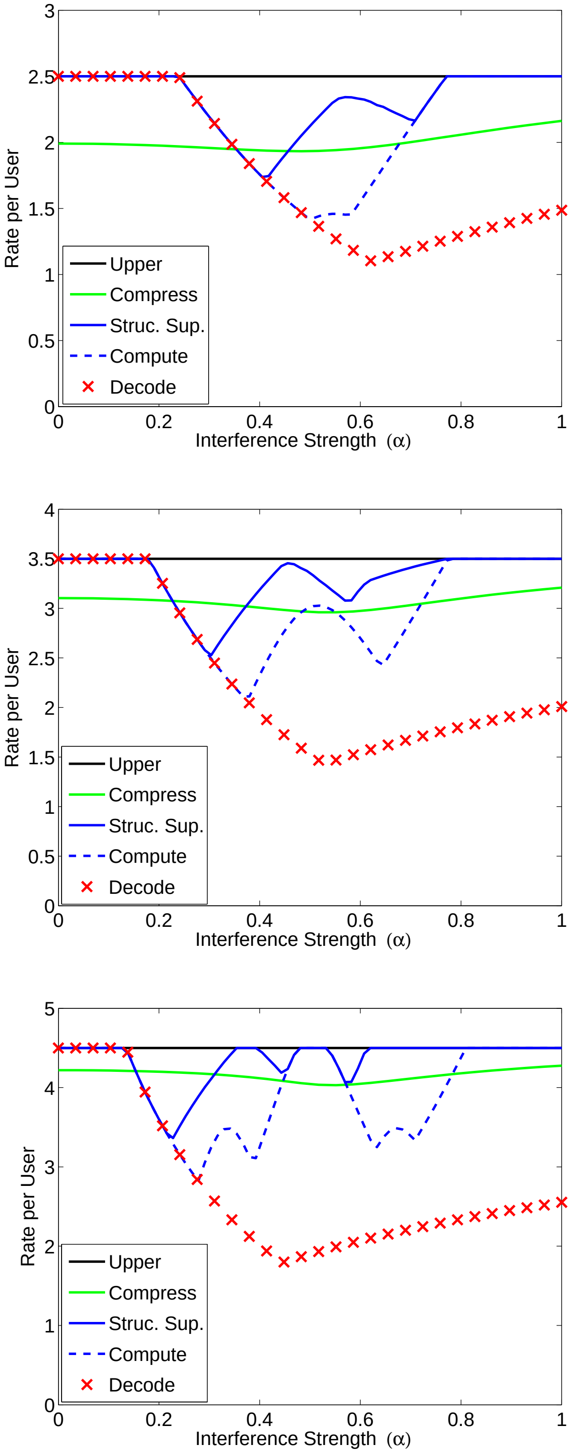

Figure 5.4: Rate per user for the upper bound, compress-and-forward, superposition compute-and-forward, regular compute-and-forward, and decode-and-forward at SNR $= 10$dB, $B = 2.5$ (top), SNR $= 15$dB, $B = 3.5$ (middle), and SNR $= 20$dB, $B = 4.5$ (bottom).

5.2 Slow Fading and Distributed MIMO

So far, we have considered fixed channel coefficients. Now, we demonstrate that our scheme can be applied to the slow fading scenario, even if the transmitters do not know the channel realization.[2] Under a slow fading model, the channel matrix $\mathbf{H}$ is chosen according to some probability distribution and then remains fixed for all time. As a result, we must accept some probability that the rate used by the transmitters is above the maximum rate permitted for those channel coefficients. For an achievable strategy with rate $R_{\text{SCHEME}}(\mathbf{H})$ for fixed $\mathbf{H}$, this *outage probability* is given by:

$$\rho_{\text{OUT}}(R) = P_{\mathbf{H}}\left(R_{\text{SCHEME}}(\mathbf{H}) < R\right) \tag{5.13}$$

We can also characterize the performance of a given strategy by its *outage rate*:

$$R_{\text{OUT}}(\rho) = \sup\{R : \rho_{\text{OUT}}(R) \leq \rho\}. \tag{5.14}$$

We will now compare the outage performance of compute-and-forward to the performance of classical relaying strategies over a simple network. First, it is useful to note that usually, the exact choice of the coefficient vector at a relay is not important, so long as the resulting equations can be solved for the desired messages. Thus, to maximize the performance of our overall scheme, we should have each relay decode the equation that is available at the *highest rate*.

Consider the two user distributed MIMO network in Figure 5.5. There are two sources, two relays, and one destination. The relays see the transmitters through $\mathbf{H}$ whose entries are i.i.d. Rayleigh. Each relay is given a bit pipe with rate B bits per channel use to the destination. The destination would like to recover both message $\mathbf{w}_1$ and $\mathbf{w}_2$ at highest possible symmetric outage rate. Recall that for a symmetric rate point to be achievable, both transmitters must be able to communicate their messages with at least that rate. This a model of multiple antennas that are not co-located with the receiver and thus have rate constraints on communicating their observations. Several other groups have studied this problem including [36; 130].

The basic compute-and-forward strategy has each relay decode the equation with the highest rate and pass that to the destination. If the equations received by the destination are full rank, decoding is successful. Unfortunately, at low SNR, the probability that the equations are not full rank is quite high as shown in Figure 5.6. One simple solution is to force each relay to choose an equation with $a_{mm} \neq 0$. This results in equations that are far more likely to be solvable at the expense of slightly lower computation rates. The achievable rates for these two strategies are given below and are plotted in Figure 5.7 for $B = 2$ and outage probability $\rho = 1/4$.

[2] This section is drawn from [108].

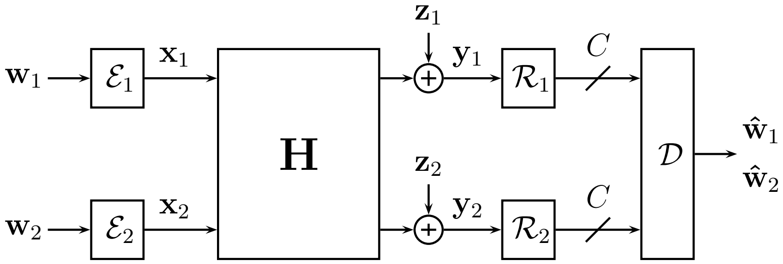

Figure 5.5: Two transmitters communicate to a distributed MIMO receiver with two antennas. Each antenna has a rate C bit pipe to the receiver.

$$\begin{aligned} \mathcal{A} &= \Big\{ \mathbf{a} \in \mathbb{Z}^M : \|\mathbf{a}\|^2 \le 1 + \|\mathbf{h}\|^2 \mathsf{SNR} \Big\} \\ R_{\text{MAX},m} &= \max_{\mathbf{a}_m} \log \left(\left(\|\mathbf{a}_m\|^2 - \frac{\mathsf{SNR}\ |\mathbf{h}_m^* \mathbf{a}_m|^2}{1 + \mathsf{SNR}\|\mathbf{h}_m\|^2} \right)^{-1} \right) \\ R_{\text{NZ},m} &= \max_{\substack{\mathbf{a}_m \\ a_{mm} \neq 0}} \log \left(\left(\|\mathbf{a}_m\|^2 - \frac{\mathsf{SNR}\ |\mathbf{h}_m^* \mathbf{a}_m|^2}{1 + \mathsf{SNR}\|\mathbf{h}_m\|^2} \right)^{-1} \right) \end{aligned} \tag{5.15}$$

$$R_{\text{COMP}}(\mathbf{H}) = \begin{cases} \min \Big(B, \ \ \min_m R_{\text{MAX},m} \Big) & \text{if } \mathbf{A} \text{ is full rank,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.16}$$

$$R_{\text{COMP,NZ}}(\mathbf{H}) = \begin{cases} \min \Big(B, \ \ \min_m R_{\text{NZ},m} \Big) & \text{if } \mathbf{A} \text{ is full rank,} \\ 0 & \text{otherwise.} \end{cases} \tag{5.17}$$

For decode-and-forward, we require that each relay is responsible for a single message. It attempts to recover this message either by treating the other message as noise or decoding both messages. The rate for this strategy is evaluated below and plotted in Figure 5.7. For more details on decode-and-forward for multiple relays (as well as compress-and-forward and cut-set upper bounds), see [79].

$$R_{\text{ignore},1} = \log\left(1 + \frac{|h_{11}|^2\mathsf{SNR}}{1 + |h_{12}|^2\mathsf{SNR}}\right) \tag{5.18}$$

$$R_{\text{ignore},2} = \log\left(1 + \frac{|h_{22}|^2\mathsf{SNR}}{1 + |h_{21}|^2\mathsf{SNR}}\right) \tag{5.19}$$

$$\begin{aligned} R_{\text{decode},m} = \min\Bigg(& \log\left(1 + |h_{m1}|^2\mathsf{SNR}\right), \\ & \log\left(1 + |h_{m2}|^2\mathsf{SNR}\right), \\ & \frac{1}{2}\log\left(1 + \|\mathbf{h}_m\|^2\mathsf{SNR}\right)\Bigg) \end{aligned} \tag{5.20}$$

$$R_{ii} = \min(R_{\text{ignore},1}, R_{\text{ignore},2}) \tag{5.21}$$

$$R_{id} = \min(R_{\text{ignore},1}, R_{\text{decode},2}) \tag{5.22}$$

$$R_{di} = \min(R_{\text{decode},1}, R_{\text{ignore},2}) \tag{5.23}$$

$$R_{dd} = \min(R_{\text{decode},1}, R_{\text{decode},2}) \tag{5.24}$$

$$R_{DF}(\mathbf{H}) = \min(\max(R_{ii}, R_{id}, R_{di}, R_{dd}), B) \tag{5.25}$$

For our upper bound, we use a cut-set bound that either groups the relays with the sources

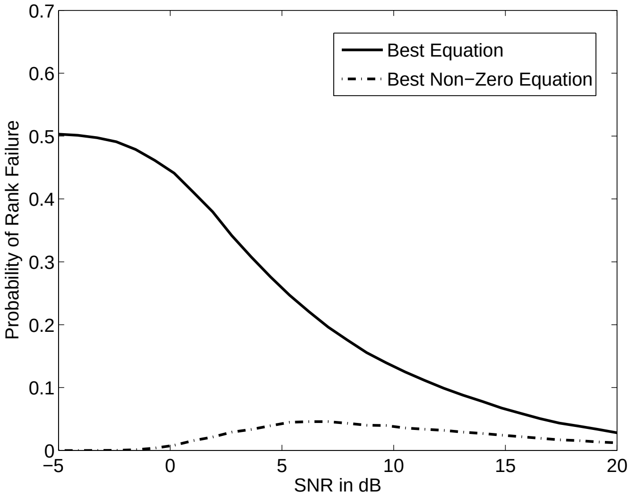

Figure 5.6: Probability of rank failure for the 2-user distributed MIMO multiple-access channel by having each relay decode the best equation and the best non-zero equation.

or with the destination. This yields the following bound on the symmetric rate:

$$\begin{aligned} R_{\text{MIMO}}(\mathbf{H}) = \min \Big(& \log\big(1 + (|h_{11}|^2 + |h_{21}|^2)\mathsf{SNR}\big), \\ & \log\big(1 + (|h_{12}|^2 + |h_{22}|^2)\mathsf{SNR}\big), \\ & \frac{1}{2}\log\det\left(\mathbf{I} + \mathbf{H}\mathbf{H}^*\mathsf{SNR}\right)\Big) \end{aligned} \tag{5.26}$$

$$R_{\text{UPPER}}(\mathbf{H}) = \min(R_{\text{MIMO}}(\mathbf{H}), B) \tag{5.27}$$

Finally, we consider the performance of compress-and-forward with i.i.d. Gaussian codebooks. The variance of the channel observation at relay m is $1 + \|\mathbf{h}_m\|^2\mathsf{SNR}$ and we have to compress this using B bits. At the destination, one can equivalently write this as a MIMO channel with channel matrix $\mathbf{H}_{\text{CF}}$:

$$\mathsf{SNR}_{\text{CF},m} = \frac{\mathsf{SNR}(2^B - 1)}{2^B + \mathsf{SNR}\|\mathbf{h}_m\|^2} \tag{5.28}$$

$$\mathbf{H}_{\text{CF}} = \begin{bmatrix} \sqrt{\mathsf{SNR}_{\text{CF},1}} & 0 \\ 0 & \sqrt{\mathsf{SNR}_{\text{CF},2}} \end{bmatrix} \mathbf{H} \tag{5.29}$$

$$R_{\text{CF}}(\mathbf{H}) = \min(R_{\text{MIMO}}(\mathbf{H}_{\text{CF}}), B). \tag{5.30}$$

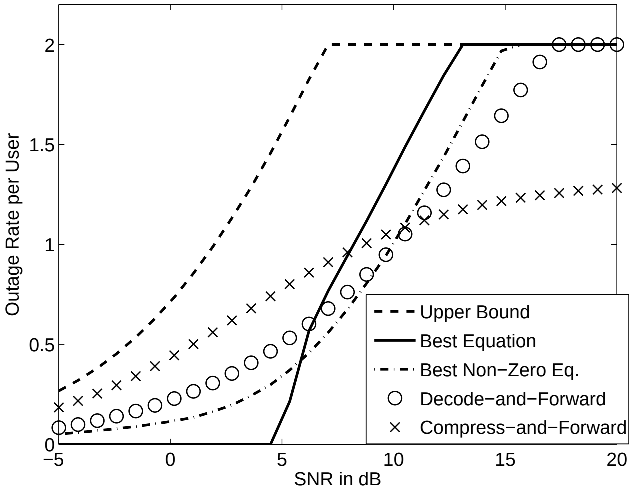

Figure 5.7: Symmetric outage rates for the 2-user distributed MIMO multiple-access channel with i.i.d. Rayleigh fading only known at the receivers. Here, we set $B = 2$ and outage probability $\rho = 1/4$.

From Figure 5.7, we can see that compute-and-forward (with the best equation) outperforms all other strategies starting at approximately 8dB. It also saturates the bit pipes to the destination at an SNR 5dB less than required for decode-and-forward. However, the gains are not as dramatic as observed in Example 11. For non-integer coefficients, we can only decode an integer combination and the remainder acts like additional noise. Despite this penalty, compute-and-forward is the best strategy in the moderate SNR regime. Compress-and-forward is a good strategy at low SNR but since it cannot saturate the bit pipe rate B since it introduces quantization noise. Decode-and-forward suffers at high SNR since it treats interference as noise.

Note that the encoding strategy for compute-and-forward does not depend on the choice of equation coefficients at the relay. Therefore, one can obtain the maximum of the best equation rate and the best non-zero equation rate with the same strategy simply by disallowing certain coefficients at the relays past an appropriate SNR.

Remark 27. Since the channel from the transmitters to the relays is essentially a 2-user interference channel, it may be useful to have each transmitter send out a public and a private message as in the Han-Kobayashi scheme [58]. Such a scheme might improve the performance of both the decode-and-forward strategy and the compute-and-forward strategy (by employing superposition as in Section 4.6).

5.3 Multicasting over Multiple-Access Networks

Consider a single source that communicates to a single destination across a network of relays connected by bit pipes. It is well known that for this *unicast problem* the capacity is given by the max-flow min-cut theorem and is achievable by routing [47; 40]. If the source wants to multicast the same message to multiple destinations, then clearly the best possible performance is upper bounded by the minimum over each unicast problem. As shown by Ahlswede et al., this upper bound is in fact the multicast capacity and is achievable through *network coding* [4]. In network coding, each relay transmits a function of its received packets, rather than simply routing them. This "mixing" of packets turns out to be necessary as routing cannot achieve the multicast capacity for all wired networks. Now, if we want to multicast a message over a wireless network, one approach is to first transmit packets to the relays and then have them compute the required functions for network coding. However, using the tools developed in Chapters 3 and 4, we can increase our overall throughput by performing the network coding directly on the channel. This way we do not have to fight the interference inherent to the physical layer and can instead turn it to our advantage.

In this section, we look at relay networks connected together by noisy multiple-access channels. For finite field channels, we will derive the multicast capacity and for AWGN channels, we will give an achievable rate that is a strict improvement over known strategies.[3]

5.3.0.1 Related Work

There has been considerable interest in using the physical layer for network coding. Bhadra, Gupta, Shakkottai considered finite field networks with both broadcast and multiple-access constraints and showed how to approach the multicast capacity as the field size tends to infinity [15]. In [164], Zhang, Liew, and Lam take a communications perspective and use uncoded transmission to send mod-2 sums of bits over a multiple-access channel at lower bit error rates than possible with separate transmissions. Katti, Gollakota, and Katabi simulated a practical system that uses a clever scheme to add bits uses the phases of transmitted signals [68].

In concurrent work to our own, Narayanan, Wilson, and Sprintson, showed how to use lattices for network coding over a two-way relay channel [101]. This was later extended by Nam, Chung, and Lee to the case of unequal power constraints [99].

We also point to the work of Ratnakar and Kramer that examines networks with only broadcast constraints and finds the capacity for deterministic channels [126]. As in our considerations, they find that it is not possible to treat channel coding and network coding separately, as gains are possible through a joint design.

5.3.1 Problem Statement

A channel network is usually thought of as a graph where the vertices are the encoders and decoders and the edges are the point-to-point channels with known capacities. For our problem, we will need a bit more notation to cleanly represent both point-to-point channels and MACs.

Definition 47. A *multiple-access network*, $\mathcal{G}_{\text{MAC}}$, consists of the following elements:

1. $\mathcal{V}_N$: the encoder/decoder nodes of the network. Each node, v, has a unique label taken from the positive integers, $v \in \mathbb{Z}_+$, and consists of a decoding function $g_{v_j v}$ for each incoming edge (v_j, v) and an encoding function $f_{v v_k}$ for each outgoing edge (v, v_k).

2. v^S: the source node. One element of $\mathcal{V}_N$. The source transmits the message, $w \in \{1, 2, \ldots, 2^{nR}\}$.

3. $(v_1^R, v_2^R, \ldots, v_L^R)$: the receiver nodes. Each one is an element of $\mathcal{V}_N$ and produces an estimate of the transmitted message, $\hat{w}_\ell$.

4. $\mathcal{V}_{\text{MAC}}$: the MACs in the network. Each MAC, m, has a unique integer label, $m \in \mathbb{Z}_+$.

[3] The results in this section originally appeared in [105; 106].

5. $\mathcal{C}_{NN}$: the directed point-to-point channels in the network. Each channel has a unique integer label, $c_{NN} \in \mathbb{Z}_+$, and the labels of its inputs and output nodes are given by the functions $v_{\text{IN}}(c_{NN})$ and $v_{\text{OUT}}(c_{NN})$ respectively.

6. $\mathcal{C}_{NM}$: the input edges from nodes to MACs. Each edge has a unique integer label, $c_{NM} \in \mathbb{Z}_+$, and the labels of its inputting node and destination MAC are given by the functions $v_{\text{IN}}(c_{NM})$ and $v_{\text{OUT}}(c_{NM})$ respectively.

7. $\mathcal{C}_{MN}$: the output edges from a MAC to a node. We assume that the output of a given MAC is only observed by a single node. Each edge has a unique integer label, $c_{MN} \in \mathbb{Z}_+$, and the label of its MAC and destination node are given by the functions $v_{\text{IN}}(c_{MN})$ and $v_{\text{OUT}}(c_{MN})$ respectively.

8. $X_{v_j v_k}[i]$: the channel input on the edge (v_j, v_k) at time i. The encoders are constrained to only produce channel inputs from time $i = 1$ to time $i = n$.

9. $Y_{v_j v_k}[i]$: the channel output on the edge (v_j, v_k) at time i.

We also assume that there are a finite number of nodes and channels in the network, $|\mathcal{V}_N| + |\mathcal{V}_{\text{MAC}}| + |\mathcal{C}_{NN}| + |\mathcal{C}_{NM}| + |\mathcal{C}_{MN}| < \infty$.

Definition 48. A multicast rate, R, is *achievable* if $\forall \epsilon > 0$ and n large enough there exist encoding and decoding functions for the network such that the average probability of error is less than ϵ:

$$\begin{aligned} &\hat{w}_\ell = f_{v_\ell^R}(Y^n_{v_\ell^R}) \\ &\Pr\left(\{\hat{w}_1 \neq w\} \cup \cdots \cup \{\hat{w}_L \neq w\}\right) < \epsilon, \end{aligned} \tag{5.31}$$

where $w \in \{1, 2, \ldots, 2^{nR}\}$ and $Y^n_{v_\ell^R}$ represents all the channel outputs observed by the ℓ^{th} receiver.

Definition 49. The *multicast capacity* is the supremum of all achievable multicast rates.

Definition 50. A *point-to-point network*, $\mathcal{G}_{\text{POINT}} = (\mathcal{V}_N, \mathcal{C}_{NN})$, is just a multiple-access network without any multiple-access nodes, $\mathcal{V}_{\text{MAC}} = \mathcal{C}_{NM} = \mathcal{C}_{MN} = \emptyset$.

Definition 51. A *unit bit pipe network*, $\mathcal{G}_{\text{PIPE}} = (\mathcal{V}, \mathcal{C})$, is just a point-to-point network except all of the channels, $\mathcal{C}$, are taken to be noiseless bit pipes with unit capacity. The encoding/decoding nodes are given by the set $\mathcal{V}$.

Our scheme will give achievable rates for multiple-access networks comprised of either discrete linear or Gaussian MACs. We express the achievable rate through a new point-to-point network that results from an appropriate transformation of our original network. The achievable rate is then given by the multicast capacity of the point-to-point network. We will also demonstrate that in some cases our achievable rates coincide with the simple upper bound due to the max-flow min-cut theorem of Ford and Fulkerson [48].

We now briefly review some results for multicasting over point-to-point channel networks using linear codes. In [84] and [76], it was shown that linear encoding and decoding over a finite field is sufficient to achieve the multicast capacity. Bounds are also given on the required field size. It was independently and concurrently shown by Ho et al. in [60], Jaggi et al. in [66], and Sanders et al. [132] that the field size only needs to be larger than the number of receivers. We reproduce the version from [60] below as it will be useful to us in proving our main theorems.

Definition 52. Let $\mathcal{G}_{\text{PIPE}} = (\mathcal{V}, \mathcal{C})$ and let $\mathbb{F}_q$ be a finite field of size q. An *algebraic network code* is a set of linear functions for a unit bit pipe network. Specifically, the encoding function from node v_j to node v_k is constrained to be a linear function of its observations from each incoming edge:

$$X_{v_j v_k}[i] = \sum_{v_r} \alpha_{v_r v_j} Y_{v_r v_j}[i]. \tag{5.32}$$

where $Y_{v_r v_j}[i]$ is the value seen by node v_j at time i on the incoming edge from node v_r and $Y_{v_r v_j}[i], \alpha_{v_r v_j} \in \mathbb{F}_q$ for all $v_r \in \mathcal{V}$.

Lemma 17 (Ho et al.)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{C})$ be a unit bit pipe network with a single source and L receivers. The multicast capacity is given by the max-flow min-cut bound and can be achieved by an algebraic network code over any finite field larger than L ($\mathbb{F}_q, \quad q > L$).*

For a full proof, see [61]. This result can be easily extended to point-to-point networks as shown below.

Lemma 18. *The multicast capacity of any point-to-point channel network with a single source and L receivers, $\mathcal{G}_{POINT} = (\mathcal{V}_N, \mathcal{C}_{NN})$, is achievable by combining point-to-point channel codes with an algebraic network code with field size larger than L.*

Proof. Let the capacity of each channel $c_{NN} \in \mathcal{C}_{NN}$ be given by $R(c_{NN})$. Choose capacity-achieving codes for each channel such that with probability $1 - \frac{\delta}{|\mathcal{C}_{NN}|}$ we get a noiseless channel with rate $\hat{R}(c_{NN}) = R(c_{NN}) - \frac{\delta}{2|\mathcal{C}_{NN}|}$. Now choose $\gamma > 0$ such that:

$$\max_{c_{NN} \in \mathcal{C}_{NN}} \left(\hat{R}(c_{NN}) - \gamma \left\lfloor \frac{\hat{R}(c_{NN})}{\gamma} \right\rfloor \right) < \frac{\delta}{2|\mathcal{C}_{NN}|} \tag{5.33}$$

Create a γ bit pipe network, $\mathcal{G}_{\text{PIPE}} = (\mathcal{V}, \mathcal{E})$, where the nodes are the same as in $\mathcal{G}_{\text{POINT}}$, $\mathcal{V} = \mathcal{V}_N$. For each channel $c_{NN} \in \mathcal{C}_{NN}$ with capacity $\hat{R}(c_{NN})$ in $\mathcal{G}_{\text{POINT}}$, place $\left\lfloor \frac{\hat{R}(c_{NN})}{\gamma} \right\rfloor$

noise-free channels with capacity γ in the bit pipe network with the same connectivity. Since all channels in $\mathcal{G}_{\text{PIPE}}$ have the same capacity, we are free to generate an algebraic code that achieves the multicast capacity using Lemma 17. This algebraic code is sent over the channels in $\mathcal{G}_{\text{POINT}}$ using the channel codes chosen above and a timesharing approach. For instance, for a channel with capacity $R(c_{NN})$, we consider the n total channel uses in chunks of $\left(\left\lfloor \frac{R(c_{NN})}{\gamma} \right\rfloor\right)^{-1} n$. Each of these chunks can be used to send γn bits reliably and thus can be used to send one function. Over all cuts in the max-flow min-cut characterization, the largest reduction in rate is at most $\frac{\delta}{2}$ (due to the gap to capacity). Considering the channels only in units of γ also causes at most a $\frac{\delta}{2}$ rate reduction over the worst possible cut. Thus, as $\delta \to 0$, we can approach the multicast capacity. □

Now we will use computation codes to map an algebraic network code onto multiple-access channels in the network. We first explore two examples that are a variant of the butterfly network given in [4]. Then, we state our theorems on multicasting over general finite field and AWGN multiple-access networks.

5.3.2 Motivating Examples

We will focus on two variants of the butterfly network from [4, Figure 7] to demonstrate our coding idea. First, we review the network coding scheme for the original butterfly network which is depicted in Figure 5.8. The goal is for the source node (at the top of the graph) to send two bits a and b to the two receiver nodes (at the bottom of the graph). Each edge is a bit pipe with unit capacity. It can be checked that there is no routing assignment that allows both receivers to recover both bits. However, by computing the mod-2 sum of the bits and sending this down the center path, both receivers can infer both bits. Thus, some form of network coding is necessary to achieve the multicast capacity of networks.

In our examples, we will put a multiple-access channel in the center path and use computation codes to transmit the sum of the bits to the receiver. The first example considers the binary alphabet case and the second looks at the Gaussian case.

5.3.2.1 Mod-2 Adder MAC

Consider the channel network in Figure 5.9(a). Each vertex on the graph represents a decoder/encoder pair. The sender is at the top of the graph and the two receivers are at the bottom. The labeled edges represent noiseless bit pipes each with capacity C. At the center of the graph is a MAC with inputs X_1 (from the left) and X_2 (from the right) and output $Y = X_1 \oplus X_2 \oplus Z$ where Z is an i.i.d. Bernoulli(p) sequence. Note that the sum rate of this MAC is upper bounded by $1 - h_B(p)$.

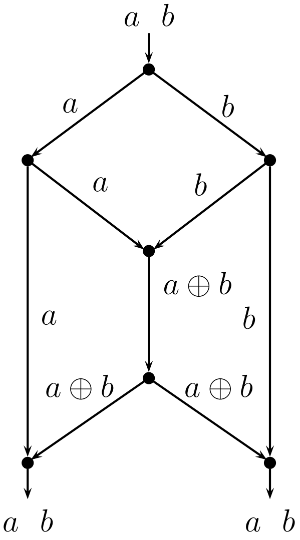

Figure 5.8: Butterfly network introduced by Ahlswede et al.. The multicast capacity of this network is not achievable through routing but is achievable through network coding.

Theorem 24. *For the channel graph from Figure 5.9 (a) the multicast capacity is:*

$$R = B + \min(B, 1 - h_B(p)) \tag{5.34}$$

Proof. (*Converse.*) Applying the cutset bound gives that the rate to each receiver is upper bounded by:

$$R \leq B + \min(B, 1 - h_B(p)) \tag{5.35}$$

(*Achievability.*) We have a block $\mathbf{w}$ of $n(B + \min(B, 1 - h_B(p)))$ bits to transmit to both receivers. We will break up $\mathbf{w}$ in two ways. For the first, we write $\mathbf{w} = [\mathbf{w}_{11}\ \mathbf{w}_{12}]$ where the first chunk is of length nB and the second is of length $n(\min(B, 1 - h_B(p)))$. For the second, we write $\mathbf{w} = [\mathbf{w}_{21}\ \mathbf{w}_{22}]$ where the first chunk is of length $n(\min(B, 1 - h_B(p)))$ and the second is of length nB. We transmit $\mathbf{w}_{11}$ down the left path and $\mathbf{w}_{22}$ down the right path. From $\mathbf{w}_{11}$ we automatically know $\mathbf{w}_{21}$ and from $\mathbf{w}_{22}$ we know $\mathbf{w}_{12}$. We send the mod-2 sum $\mathbf{u} = \mathbf{w}_{21} \oplus \mathbf{w}_{12}$ reliably across the MAC using the linear code from Theorem 8. Finally, this mod-2 sum is conveyed to the receivers. The left receiver can compute $\mathbf{w}_{12} = \mathbf{u} \oplus \mathbf{w}_{21}$ and the right receiver can compute $\mathbf{w}_{21} = \mathbf{u} \oplus \mathbf{w}_{12}$ to fully recover $\mathbf{w}$. □

Standard random coding arguments cannot attain the optimal performance over the network in Figure 5.9 (a). The decode-and-forward and compress-and-forward rates are

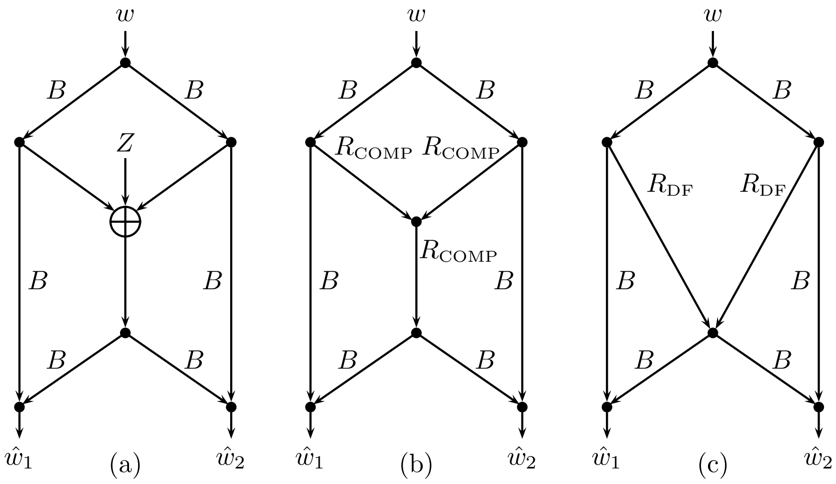

Figure 5.9: (a) Binary multiple-access variant of the butterfly network. The MAC in the center is a noisy mod-2 adder with i.i.d. Bernoulli(p) noise Z (b) Using a linear code, we can achieve the multicast capacity of the MAC butterfly network which is equivalent to the multicast capacity of this transformed network. Here, $R_{\textsf{COMP}} = 1 - h_B(p)$. (c) With a decode-and-forward strategy, we can only achieve rates on the original network that are achievable on this network. Here, $R_{\textsf{DF}} = \frac{1}{2}(1 - h_B(p))$

given by:

$$R_{\text{DF}} = B + \min\left(B, \frac{1 - h_B(p)}{2}\right) \tag{5.36}$$

$$R_{\text{CF}} = B + \min\left(B(1 - h_B(p)), (1 - h_B(p))\right) \tag{5.37}$$

If the capacity B of the point-to-point links is small enough, then fully decoding the input messages to the MAC is sufficient. If the capacity B is large enough, forwarding the output of the MAC is sufficient. However, in the intermediate regime, structured codes outperform both strategies.

We can think about our compute-and-forward strategy as transforming the multiple-access network into an equivalent point-to-point network. The multiple-access channel is replaced with a relay node with incoming and outgoing edge capacities given by the computation rate (see Figure 5.9 (b)). Usually, we try to convert interference networks into point-to-point equivalents by decoding individual messages (see Figure 5.9 (c)). This results in direct links across the multiple-access channel but with lower rates which results in a lower overall throughput.

5.3.2.2 Gaussian MAC

Consider the AWGN channel network in Figure 5.10 (a). Each vertex on the graph represents a decoder/encoder pair. The sender is at the top of the graph and the two receivers are at the bottom. We assume that all channel inputs and outputs are real-valued and that all encoders must satisfy an average power constraint, $\frac{1}{n}\sum_{i=1}^{n} x_m[i]^2 \leq P$. The $Z_m,\ m = 1, 2, \ldots, 7$ are drawn i.i.d. according to a Gaussian distribution with mean 0 and variance N.

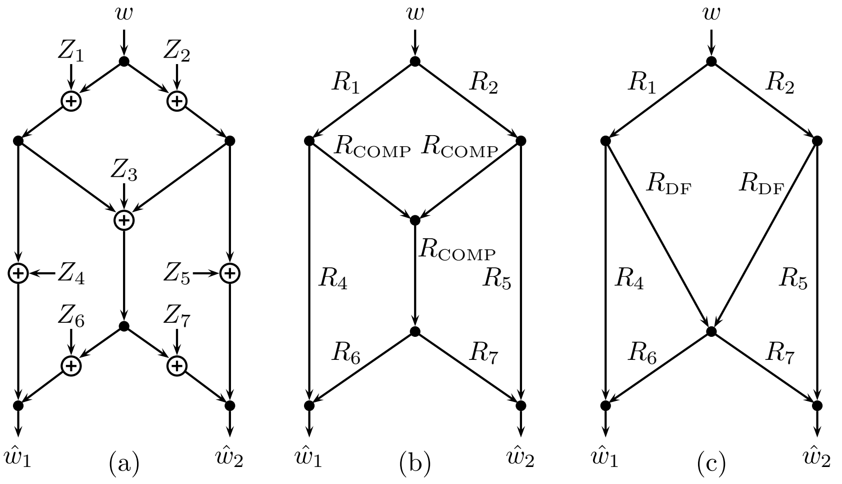

Figure 5.10: (a) Gaussian multiple-access variant of the butterfly network. (b) Using a lattice-based code, we can achieve any rate on the MAC butterfly network that is achievable on this transformed network. Here, $R_{\text{COMP}} = \max\left(\frac{1}{4}\log\left(1+\frac{2P}{N}\right), \frac{1}{2}\log\left(\frac{1}{2}+\frac{P}{N}\right)\right)$ where P is the per user power of the MAC. (c) With a decode-and-forward strategy, we can only achieve rates on this network where $R_{\text{DF}} = \frac{1}{4}\log\left(1+\frac{2P}{N}\right)$

Theorem 25. *The following multicast rate is achievable on the channel network in Figure 5.10(a):*

$$R = \frac{1}{2}\log\left(1+\frac{P}{N}\right) + \max\left(\frac{1}{4}\log\left(1+\frac{2P}{N}\right), \frac{1}{2}\log\left(\frac{1}{2}+\frac{P}{N}\right)\right) \tag{5.38}$$

Proof. Let $\mathbf{w} \in \mathbb{F}_p^k$ be the message vector where p is a prime. Using Corollary 2, we can reliably send the sum of messages over the AWGN multiple-access channel at a rate of:

$$R_{\text{COMP}} = \max\left(\frac{1}{4}\log\left(1+\frac{2P}{N}\right), \frac{1}{2}\log\left(\frac{1}{2}+\frac{P}{N}\right)\right). \tag{5.39}$$

The first term in the maximization comes from decoding the messages separately using coefficient vectors $\mathbf{a} = [1\ 0]^T$ and $\mathbf{a} = [0\ 1]^T$ combined with time-sharing and adding them together. The second term comes from directly decoding the sum using coefficient vector $\mathbf{a} = [1\ 1]^T$. We split the message into three pieces $\mathbf{w} = [\mathbf{w}_1\ \mathbf{w}_2\ \mathbf{w}_3]$ such that the first two pieces correspond to rate R_{COMP} and the last piece corresponds to rate $\frac{1}{2}\log\left(1+\frac{P}{N}\right)-R_{\text{COMP}}$. We send $\mathbf{w}_1$ and $\mathbf{w}_3$ down the left path and $\mathbf{w}_2$ and $\mathbf{w}_3$ down the right path. We transmit the sum of $\mathbf{w}_1$ and $\mathbf{w}_2$ over the multiple-access channel and onto the receivers. Now, the left receiver has $\mathbf{w}_1$, $\mathbf{w}_3$ and $\mathbf{w}_1 \oplus \mathbf{w}_2$ from it can recover the messages. Similarly, the right receiver has $\mathbf{w}_2$, $\mathbf{w}_3$ and $\mathbf{w}_1 \oplus \mathbf{w}_2$ and can decode as well. □

As in the discrete case, random coding arguments will not suffice for attaining this performance. The encoder following the MAC either decodes the messages in their entirety or quantizes the observed signal and forwards it. Below we give the best achievable rates for decode-and-forward and compress-and-forward with Gaussian codebooks:

$$R_{\text{DF}} = \frac{1}{2}\log\left(1+\frac{P}{N}\right) + \frac{1}{4}\log\left(1+\frac{2P}{N}\right)$$

$$R_{\text{CF}} = \frac{1}{2}\log\left(1+\frac{P}{N}\right) + \frac{1}{2}\log\left(1+\frac{P}{N}\left(\frac{P}{3P+N}\right)\right)$$

Note that for this example, compress-and-forward and amplify-and-forward yield the same performance.

Again, we can view our strategy as a network transformation as shown in Figure 5.10 (b). As before, decoding the messages individually results in a different network topology with a lower end-to-end throughput.

5.3.3 General Networks

We now give two multicasting results for multiple-access networks. First, we give the multicast capacity for when the MACs are constrained to be noisy linear functions over a finite field. Second, we give achievable rates for any Gaussian multiple-access network.

Our results can be viewed as a transformation of the original multiple-access network into a point-to-point network (see Figure 5.11). Any multicast rate that is achievable on the point-to-point network is achievable on the original network. For the finite field case, this will yield the multicast capacity and, for the Gaussian case, this is a strict improvement over known strategies.

Definition 53. Let $\mathcal{G}_{\text{MAC}}$ be a Gaussian multiple-access network. An equivalent point-to-point network, $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, is constructed from the original network by the following steps:

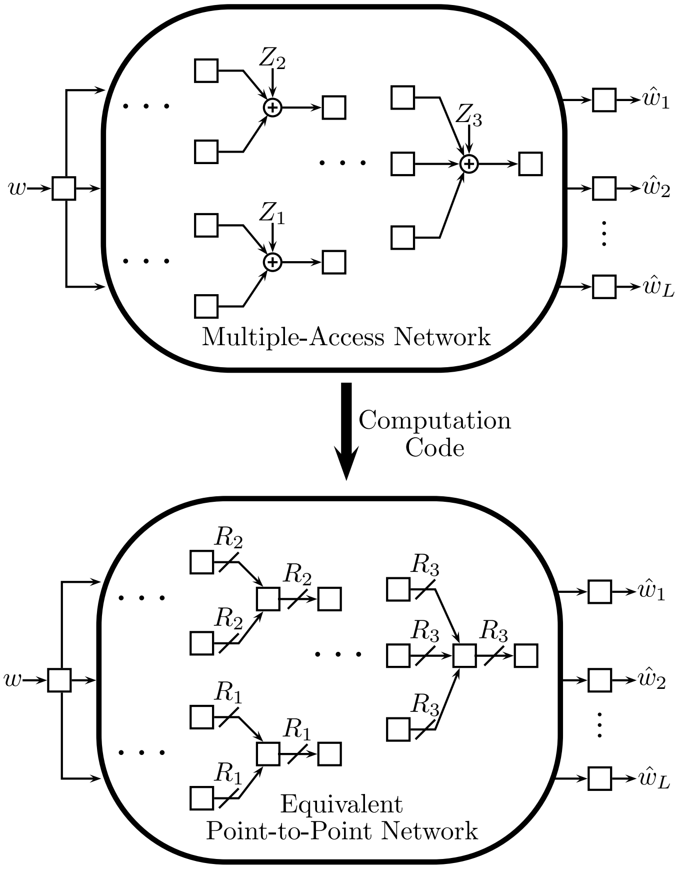

Figure 5.11: Using computation codes, we can convert every MAC in a network into a reliable linear relay with incoming and outgoing rate links given by the computation rate.

- Let the set of encoder/decoder nodes, $\mathcal{V}'$, in the new network be given by the original encoder/decoder nodes as well as the original MACs, $\mathcal{V}' = \mathcal{V}_N \cup \mathcal{V}_{\text{MAC}}$.
- Let the channels in the new network, $\mathcal{C}'$, be given by the original point-to-point channels as well as the input and output edges to the MACs, $\mathcal{C}' = \mathcal{C}_{NN} \cup \mathcal{C}_{NM} \cup \mathcal{C}_{MN}$. The connectivity of these edges is the same as in the original network.
- Set the capacities of the edges taken from $\mathcal{C}_{NM}$ and $\mathcal{C}_{MN}$ to be the computation rates of the associated MACs for sending the sum of the inputs, $R_{\text{COMP}}(m)$.

5.3.3.1 Finite Field Multiple-Access Networks

We assume all of the MACs in our channel network $\mathcal{G}_{\text{MAC}}$ are linear with respect to $\mathbb{F}_p$ for some field size $p \in \mathbb{Z}_+$ (see Definition 19). Each MAC in the network $m \in \mathcal{V}_{\text{MAC}}$ has a maximum sum rate $R_{\text{SUM}}(m)$ (see Definition 15).

Theorem 26. *The max-flow min-cut bound for multicasting is achievable for a finite field multiple-access network, $\mathcal{G}_{MAC}$, if $q > L$, where q is the MAC field size and L is the number of receivers in the network.*

Proof. (*Achievability.*) First, we will transform our network of noisy MACs and point-to-point channels, $\mathcal{G}_{\text{MAC}}$, into an equivalent point-to-point network, $\mathcal{G}'$, using Definition 53. We will then find an appropriate algebraic network code which we will map back to the original network using computation coding. Note that for finite-field multiple-access channels the computation rate R_{COMP} is equal to the maximum sum rate R_{MAX}.

The new point-to-point network, $\mathcal{G}'$ has the same max-flow min-cut characterization as our original network, $\mathcal{G}_{\text{MAC}}$. Choose $\epsilon > 0$ and let $R = C - \epsilon$ where C is the multicast capacity of $\mathcal{G}'$. Using Lemma 18, we can find a set of channel codes and an algebraic network code over the field $\mathbb{F}_q$ that achieves a multicast rate R over $\mathcal{G}'$ with an error probability less than $\frac{\epsilon}{2}$. We now map this strategy back to the multiple-access network $\mathcal{G}_{\text{MAC}}$. Recall that in the proof of Lemma 18, we treat a link with capacity B as $\lfloor \frac{B}{\gamma} \rfloor$ separate links (with the same connectivity) each with capacity γ. This results in a γ bit pipe network for which we can find an algebraic network code and, by choosing γ small enough, we can approach the multicast capacity of the point-to-point network.

Each MAC $m \in \mathcal{V}_{\text{MAC}}$ in the original network is replaced with a node in the transformed network. If the maximum sum rate of the MAC is $R_{\text{SUM}}(m)$, then we have $\lfloor \frac{R_{\text{SUM}}(m)}{\gamma} \rfloor$ outgoing links in the γ bit pipe network. Each of these outgoing links carries a linear function of the incoming links to the node. On the original MAC, we duplicate these linear functions using a computation code from Theorem 8 with probability of error no greater than $\frac{\epsilon}{2|\mathcal{V}_{\text{MAC}}|}$. Each function is allocated $\lfloor \frac{R_{\text{SUM}}(m)}{\gamma} \rfloor n$ channel uses. For the point-to-point channels, we use the same encoding and decoding as in $\mathcal{G}'$. Thus, we can achieve any multicast rate on the original network that is achievable on the transformed network.

(*Converse.*) Since our network transformation can only increase the multicast capacity of the network and we can achieve any rate less than the transformed capacity, we get that our scheme meets the max-flow min-cut bound. □

The network transformation performed above is not accessible using standard random coding arguments. To duplicate the same topology, to each node $m \in \mathcal{V}'$ that represents a MAC, we could only provide a total rate of $R_{\text{MAX}}(m)$ to all incoming links. Using a computation code, each incoming link receives a rate of $R_{\text{MAX}}(m)$ all to itself.

5.3.3.2 Gaussian Multiple-Access Networks

We now assume that all channels are either AWGN point-to-point channels or Gaussian MACs (with inputs and outputs over the reals). We further assume each user faces an identical power constraint and that the channel quality is controlled by the noise variance $N_m \in \mathbb{R}_+$.

More formally, *all* channels in our channel network $\mathcal{G}_{\text{MAC}}$ are constrained to be of the form $Y_m = X_1 + X_2 + \ldots X_J + Z_m$ where Z_m is an i.i.d. Gaussian sequence of mean 0 and variance N_m. Furthermore, the X_j must satisfy power constraints of the form $\frac{1}{n}\sum_{i=1}^{n}(x_j[i])^2 \leq P$.

Theorem 27. *Let $\mathcal{G}_{MAC}$ be a Gaussian multiple-access network. Any multicast rate achievable on the equivalent point-to-point network is achievable on the original network for the following computation rate:*

$$R_{COMP} = \frac{1}{2}\log\left(\frac{1}{J_m} + \frac{P}{N_m}\right) \tag{5.40}$$

where N_m is the noise variance of the MAC m associated to these edges in the original network and J_m is the number of inputs to these MACs.

Proof. The proof is nearly identical to that of Theorem 26. First, we find the equivalent point-to-point network using Definition 53 and find a multicast capacity-achieving algebraic network code with field size p. Next, we use computation codes for each MAC using Corollary 2 with coefficient vector $\mathbf{a} = [1 \ 1 \ \cdots \ 1]^T$. We then map the linear functions from the appropriate node in $\mathcal{V}'$ onto the MAC. Note that although the computation code is targeted at the sum but we can simply pre-multiply by the desired coefficient at each transmitter to replicate the network code. □

Using a purely random code, we can also achieve the same network transformation but only with a lower computation rate given by:

$$R_{DF} = \frac{1}{2J_m}\log\left(1 + \frac{J_m P}{N_m}\right). \tag{5.41}$$

In general, higher rates are possible through a more involved network transformation. Rather than simply associating the computation rate in (5.40) with each incoming link, we can allow for any coefficient vector and only connect those incoming links with non-zero coefficients. This strategy, combined with time-sharing and successive cancellation, allows us to recover the usual solution (which allocates rates according to the multiple-access rate region) as well as the performance in Theorem 27. Thus, our approach results in higher rates than those possible with the classical random coding approach. Unlike the finite field case, we are unable to reach the multicast capacity partly because our computation codes for AWGN channels do not meet the upper bound.

It is also possible to use the compute-and-forward strategy for network coding over AWGN networks with fading and broadcast constraints. However, these generalizations are cumbersome and do not yield any additional insight as they are just achievable strategies. In the next section, we give an example of wireless network coding over a two-way relay channel.

5.4 Wireless Network Coding

As shown by [61], linear network coding lends itself to a distributed, random implementation in which nodes evaluate random linear combinations of incoming packets and send them along. For a sufficiently large field size, this strategy achieves the multicast capacity of bit pipe networks. To apply it to a wireless network, we need a physical layer interface to communicate between transmitters and receivers. Compute-and-forward provides an ideal interface for linear network coding as it can make the interference part of the network code. Rather than choosing the network coding coefficients according to a uniform distribution [61], we choose the coefficients to closely match the channel realization. This allows the relay to operate at higher computation rates and, as a result, increases the end-to-end performance of the overall network code.

Another possibility is to directly use the noisy linear combinations provided by the channel as an "analog network code" [164; 68]. This is essentially an amplify-and-forward strategy and works well at high SNR. However, over a large network, noise can build up and lower the overall performance. We will compare our compute-and-forward strategy to amplify-and-forward as well as decode-and-forward through an example.

For our example, we use a slight modification of the well-studied two-way relay network [164; 68; 115; 101; 99]. In the usual considerations, two users want to exchange messages through a relay that sees a noisy combination of their signals. The relay can communicate to the users through a broadcast channel and the goal is to maximize the symmetric rate. It is well-known that for the Gaussian two-way relay channel case without fading, nested lattice codes rapidly approach the upper bound as the SNR increases [101; 99]. Here, we consider a two-way relay channel with i.i.d. Rayleigh fading on all links (see Figure 5.12). We assume that the channel coefficients are only known at the receiving end and evaluate the outage performance as in Section 5.2. The two users have power constraint SNR and the relay has power constraint $\mathsf{SNR}_{\mathrm{BC}}$.

The multicast rate from the relay to two users for a given channel realization is just the minimum of the two point-to-point rates:

$$R_{\mathrm{BC}} = \log\Big(1 + (\min(|h_3|^2, |h_4|^2))\mathsf{SNR}_{\mathrm{BC}}\Big). \tag{5.42}$$

For compute-and-forward, we can guarantee that each user can solve for its desired message

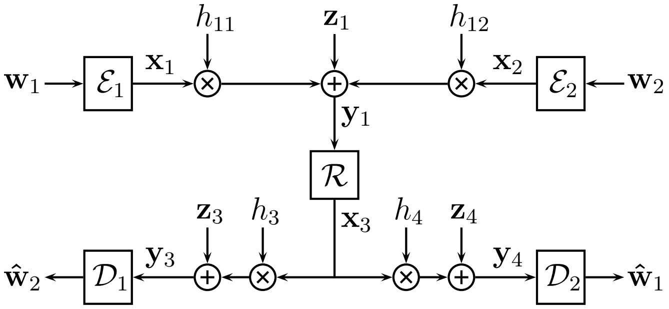

Figure 5.12: Two-way relay channel with fading. Two users exchange messages with the help of an intermediate relay.

by decoding a message with non-zero coefficients at the relay. We also let the relay perform successive cancellation to recover the messages individually. This allows it to access the corner points of the multiple-access region. For each channel realization, the relay selects the strategy that results in the highest symmetric rate.

$$\begin{aligned}
R_{\text{NZ}} &= \max_{a_{11},a_{12}\neq 0} \log\left(\left(\|\mathbf{a}_1\|^2 - \frac{\mathsf{SNR}\ |\mathbf{h}_1^*\mathbf{a}_1|^2}{1+\mathsf{SNR}\|\mathbf{h}_1\|^2}\right)^{-1}\right)\\
R_{\text{corner},1} &= \min\left(\log\left(1+\frac{|h_{11}|^2\mathsf{SNR}}{1+|h_{12}|^2\mathsf{SNR}}\right),\right.\\
&\qquad\left.\log\left(1+|h_{12}|^2\mathsf{SNR}\right)\right)\\
R_{\text{corner},2} &= \min\left(\log\left(1+\frac{|h_{12}|^2\mathsf{SNR}}{1+|h_{11}|^2\mathsf{SNR}}\right),\right.\\
&\qquad\left.\log\left(1+|h_{11}|^2\mathsf{SNR}\right)\right)\\
R_{\text{COMP}}(\mathbf{H}) &= \min\left(R_{\text{BC}}, \max\left(R_{\text{NZ}}, R_{\text{corner},1}, R_{\text{corner},2}\right)\right)
\end{aligned}$$

The relay is required to recover both message for decode-and-forward and the rate is simply the maximum symmetric rate in the multiple-access capacity region.

$$\begin{aligned} R_{\text{decode}} = \min\Bigg(&\log\left(1+|h_{11}|^2\mathsf{SNR}\right), \\ &\log\left(1+|h_{12}|^2\mathsf{SNR}\right), \\ &\frac{1}{2}\log\left(1+\|\mathbf{h}_1\|^2\mathsf{SNR}\right)\Bigg) \end{aligned} \tag{5.43}$$

$$R_{\text{DF}}(\mathbf{H}) = \min\Big(R_{\text{BC}}, R_{\text{decode}}\Big) \tag{5.44}$$

The amplify-and-forward scheme has the relay scale its observation to meet the power constraint and retransmit.

$$\begin{aligned} R_{\text{AF}}(\mathbf{H}) = \min\Bigg(&\log\left(1+\frac{|h_{12}h_3|^2\mathsf{SNR}\ \mathsf{SNR}_{\text{BC}}}{1+\|\mathbf{h}_1\|^2\mathsf{SNR}+|h_3|^2\mathsf{SNR}_{\text{BC}}}\right), \\ &\log\left(1+\frac{|h_{11}h_4|^2\mathsf{SNR}\ \mathsf{SNR}_{\text{BC}}}{1+\|\mathbf{h}_1\|^2\mathsf{SNR}+|h_4|^2\mathsf{SNR}_{\text{BC}}}\right)\Bigg) \end{aligned}$$

Finally, since we are only interested in the symmetric rate, the upper bound is just the minimum over all channel strengths.

$$\begin{aligned} R_{\text{UPPER}}(\mathbf{H}) = \min\Bigg(&R_{\text{BC}}, \log\left(1+|h_{11}|^2\mathsf{SNR}\right), \\ &\log\left(1+|h_{12}|^2\mathsf{SNR}\right)\Bigg) \end{aligned} \tag{5.45}$$

In Figure 5.13, we plot the performance of these schemes for $\mathsf{SNR}_{\text{BC}} = 2\mathsf{SNR}$ and outage probability $\rho = 1/3$. Note that here there is no possibility of rank failure for compute-and-forward and it is always at least as good as decode-and-forward. However, unlike the no fading case, compute-and-forward does not approach the upper bound as the SNR increases. In fact, past a certain SNR (in this case 25dB), compress-and-forward is the best strategy. This is primarily due to the noise penalty for decoding an integer combination from non-integer channel coefficients. For the no fading case, the channels coefficients are assumed to be unity and this penalty does not appear.

The basic insight is that, for certain network topologies, the non-integer penalty can become significant. This penalty can be managed in several ways, including:

- Adjusting the channel at the transmitters if channel state information is available.

- Using multiple antennas at the relay to steer the coefficients towards integers [163].

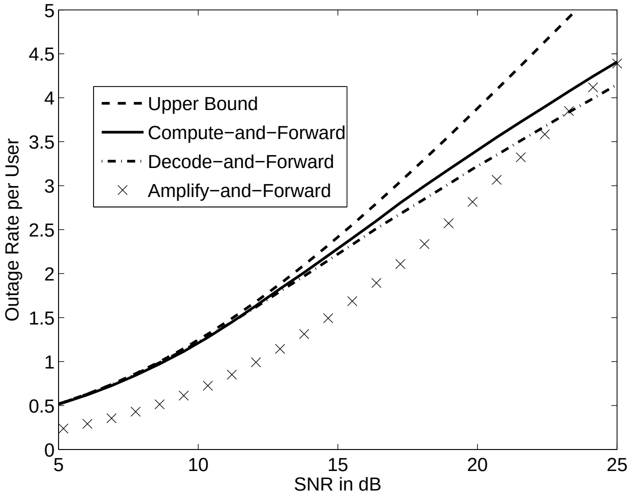

Figure 5.13: Outage rate per user for the two way relay channel with i.i.d. Rayleigh fading only known at the receivers. Here, we set $\mathsf{SNR}_{\mathsf{BC}} = 2\mathsf{SNR}$ and outage probability $\rho = 1/3$.

- Selecting transmitters opportunistically based on how close their coefficients are to integers.

However, despite the non-integer penalty, compute-and-forward offers a strict improvement over classical strategies for reliably recovering equations over a finite field. This, combined with the fact that amplify-and-forward requires fairly high values of SNR to overcome noise build-up, makes compute-and-forward a promising strategy for physical-layer network coding for multi-hop networks.

Chapter 6

Distributed Signal Processing Applications

In many sensor network applications, only certain functions of the sensor readings may be of interest. These functions may be thought of as sufficient statistics for the detection or estimation problem task at hand. For example, if the sensors each measure the local temperature, we may only want the average temperature, for which the sufficient statistic is the sum of the measurements. One way to compute these sufficient statistics is to first send all of the observed data to a fusion center. As we have seen in previous chapters, we can often compute more efficiently by exploiting the natural function of the channel using a computation code.

If the sensor observations and the desired function take values on a finite set, then we can simply employ the techniques developed in Chapters 3 and 4. However, for continuous-valued observations and functions, we need new tools. Unlike in the discrete case, where functions can be reliably recovered, here we will have to tolerate some *distortion*. In this chapter, we develop a computation code for sending the sum of Gaussian sources over a Gaussian multiple-access channel. We then explore an application to a simple sensor network scenario first studied by Gastpar and Vetterli [54].

We also apply our tools to construct a new gossip algorithm for distributed averaging. The standard gossip algorithm computes pairwise averages between sensors until all sensors converge to the global average. This process is well-understood for the case where the sensors are connected by a graph of bit pipes [18]. In our variation, we take advantage of the wireless channel to average over many sensors at once. We show that this can result in both lower energy usage and delay.

6.1 Source-Channel Computation Codes

We will now develop a scheme for sending the sum of independent Gaussian sources over a Gaussian multiple-access channel (see Figure 6.1).[1] If we are given one channel use per source sample, then uncoded transmission is optimal. With more channel uses, we use a lattice-based scheme to refine our estimate of the sum.

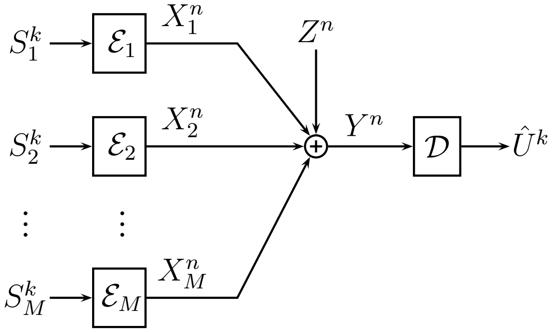

Figure 6.1: Refining the sum of Gaussian sources over a Gaussian multiple-access channel.

Definition 54 (Sources)**.** Each transmitter observes a length k i.i.d. Gaussian *source* S_m^k with mean zero and variance σ_S^2. Let $U[i] = S_1[i] + S_2[i] + \cdots + S_M[i]$ be the sum of the sources.

For ease of analysis, we will assume that we are allocated $n = k\ell$ channel uses for some positive integer $\ell \in \mathbb{Z}_+$.

Definition 55 (Encoders)**.** Each transmitter is equipped with an *encoding function* $\mathcal{E}_m$ that maps its length k observation into a length n channel input X_m^n:

$$\mathcal{E}_m : \mathbb{R}^k \to \mathbb{R}^n. \tag{6.1}$$

The channel input is subject to the usual power constraint:

$$\frac{1}{n}\sum_{i=1}^{n} (X_m[i])^2 \leq P. \tag{6.2}$$

Definition 56 (Channel Model)**.** The *channel* simply takes the sum of its inputs and adds

[1] This section is drawn from material in [104].

i.i.d. Gaussian noise Z^n with mean zero and variance σ_Z^2. The channel output at time i is

$$Y[i] = \sum_{m=1}^{M} X_m[i] + Z[i]. \tag{6.3}$$

Definition 57 (Decoders)**.** The receiver has a *decoding function* $\mathcal{D}$ which it uses to make an estimate $\hat{U}^k$ of the sum U^k:

$$\mathcal{D} : \mathbb{R}^n \to \mathbb{R}^k \tag{6.4}$$

Definition 58 (Distortion)**.** We say that a *distortion* D is achievable if for any $\epsilon > 0$ and k large enough, there exist fixed encoders $\mathcal{E}_1, \ldots, \mathcal{E}_M$ and a decoder $\mathcal{D}$ such that the estimate $\hat{U}^k$ satisfies the following mean-squared error constraint:

$$\frac{1}{k}\sum_{i=1}^{k} E\Big[(U[i] - \hat{U}[i])^2\Big] \le D + \epsilon. \tag{6.5}$$

Our goal is to achieve the smallest possible distortion while still satisfying the power constraint. First, we will give a lower bound on the achievable distortion.

Lemma 19. *The distortion at which the receiver can estimate the sum of the sources is lower bounded as follows:*

$$D \ge M\sigma_S^2 \left(\frac{\sigma_Z^2}{\sigma_Z^2 + MP}\right)^{\ell}. \tag{6.6}$$

Proof. The sum U is a Gaussian random variable with mean 0 and variance $M\sigma_S^2$. Thus, the rate distortion function for U is given by

$$R_U(D) = \frac{1}{2}\log\left(\frac{M\sigma_S^2}{D}\right). \tag{6.7}$$

The maximum sum rate of the Gaussian multiple-access channel is

$$C_{\text{SUM}}(P) = \frac{1}{2}\log\left(1 + \frac{MP}{\sigma_Z^2}\right). \tag{6.8}$$

Using the data processing inequality, we can show $R_U(D) \le \ell C_{\text{SUM}}(P)$ must be satisfied for a distortion to be achievable. Solving for D yields the desired result. □

Lemma 20. *If $\ell = 1$, then uncoded transmission is optimal for sending the sum of i.i.d.*

Gaussian sources over a Gaussian MAC and achieves distortion

$$D_{UNC} = M\sigma_S^2 \frac{\sigma_Z^2}{\sigma_Z^2 + MP}. \tag{6.9}$$

Proof. At each encoder, simply feed a source symbol, scaled to meet the power constraint, into the channel at each time step. At the decoder, we compute the minimum-mean squared error (MMSE) estimate of U. This results in the following distortion:

$$\begin{aligned} D &= E[(U - \hat{U})^2] \\ &= M\sigma_S^2 - \frac{(E[UY])^2}{E[Y^2]} \\ &= M\sigma_S^2 \frac{\sigma_Z^2}{\sigma_Z^2 + MP} \end{aligned}$$

This matches the lower bound in Lemma 19 when $\ell = 1$. □

When the channel bandwidth is larger than the source bandwidth ($\ell > 1$), then uncoded transmission will not make use of all the channel resources. One easy fix is to use a repetition code and repeat each uncoded transmission ℓ times. This results in the following distortion.

$$D_{\text{REP}} = M\sigma_S^2 \frac{\sigma_Z^2}{\sigma_Z^2 + \ell MP}, \tag{6.10}$$

While this is an improvement over not using the extra bandwidth, we would like the distortion to fall exponentially with ℓ. We now develop a lattice-based strategy for refining the estimate of the sum. In [74; 75], Kochman and Zamir develop an elegant joint source-channel lattice scheme for sending a Wyner-Ziv Gaussian source over a dirty paper channel. Our distributed refinement scheme consists of two main steps. First, we use uncoded transmission to send a noisy sum to the decoder. Then, we have each encoder run a version of the Kochman-Zamir scheme targeted at the desired sum, U. Unfortunately, there is a penalty for this form of distributedness. The lattice at each encoder results in channel outputs that violate the power constraint by a factor of M. Therefore, we must scale down our inputs to meet the power constraint and accept the resulting increase in distortion at the decoder.

Theorem 28. *For $n = \ell k$, $\ell \in \mathbb{Z}_+$, the following distortion is achievable for sending a Gaussian sum over a Gaussian MAC so long as $P > \frac{M-1}{M}\sigma_Z^2$:*

$$D = M\sigma_S^2 \left(\frac{\sigma_Z^2}{\sigma_Z^2 + MP}\right)\left(\frac{M\sigma_Z^2}{\sigma_Z^2 + MP}\right)^{\ell - 1}. \tag{6.11}$$

Proof. We will first show the achievable scheme for $\ell = 2$. We thus have $2k$ channel uses to convey k sums. We will use the first k channel uses for an uncoded transmission phase as in Lemma 20. The decoder will then form an MMSE estimate $\hat{\mathbf{u}}$ of the sum $\mathbf{u} = \mathbf{s}_1 + \cdots + \mathbf{s}_M$ and use this as side information for the next phase. Thus, $\mathbf{u} = \mathbf{q} + \hat{\mathbf{u}}$ where $\mathbf{q}$ is an i.i.d. Gaussian sequence with mean 0 and variance σ_Q^2 where

$$\sigma_Q^2 = M\sigma_S^2 \frac{\sigma_Z^2}{\sigma_Z^2 + MP}. \tag{6.12}$$

Choose a sequence of good lattices, Λ_k, using Lemma 8 and scale them such that the second moment of the lattice is MP. Let $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_M$ be independent dither vectors drawn uniformly over the fundamental Voronoi region, $\mathbf{d}_m \sim \mathrm{Unif}(\mathcal{V}_0)$, and made available to the encoders and decoder.

Each encoder transmits $\mathbf{x}_m = \frac{1}{\sqrt{M}}\mathbf{v}_m$ where:

$$\mathbf{v}_m = [\gamma \mathbf{s}_m + \mathbf{d}_m] \bmod \Lambda_k. \tag{6.13}$$

for some $\gamma > 0$ to be specified later. The channel output is given by:

$$\mathbf{y} = \sum_{m=1}^{M} \mathbf{x}_m + \mathbf{z} = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} \mathbf{v}_m + \mathbf{z}.$$

The decoder then computes:

$$\begin{aligned}
\mathbf{t} &= \alpha \mathbf{y} - \left(\sum_{m=1}^{M} \mathbf{d}_m + \gamma \hat{\mathbf{u}} \right) \\
\mathbf{r} &= \mathbf{t} \bmod \Lambda_k \\
&= \left[\frac{\alpha}{\sqrt{M}} \sum_{m=1}^{M} \mathbf{v}_m + \alpha \mathbf{z} - \sum_{m=1}^{M} (\mathbf{d}_m + \gamma \mathbf{s}_m) + \gamma \mathbf{q} \right] \bmod \Lambda_k \\
&= \left[\left(\frac{\alpha}{\sqrt{M}} - 1 \right) \sum_{m=1}^{M} \mathbf{v}_m + \alpha \mathbf{z} + \gamma \mathbf{q} \right] \bmod \Lambda_k.
\end{aligned}$$

If the second moment of the term inside the modulo operation does not exceed MP, the second moment of the lattice, then we can guarantee that:

$$\lim_{k \to \infty} \Pr\left(\mathbf{r} = \left(\frac{\alpha}{\sqrt{M}} - 1 \right) \sum_{m=1}^{M} \mathbf{v}_m + \alpha \mathbf{z} + \gamma \mathbf{q} \right) = 1. \tag{6.14}$$

since Λ_k is AWGN good. The second moment can be controlled by requiring that:

$$\left(\frac{\alpha}{\sqrt{M}} - 1\right)^2 (M^2P) + \alpha^2\sigma_Z^2 + \gamma^2\sigma_Q^2 \leq MP. \tag{6.15}$$

This equation will be satisfied by our final choice of the constants α and γ. The decoder's estimate of the sum is given by:

$$\begin{aligned}
\hat{\hat{\mathbf{u}}} &= \beta\mathbf{r} + \hat{\mathbf{u}} \\
&= \beta\left(\left(\frac{\alpha}{\sqrt{M}} - 1\right)\sum_{m=1}^{M}\mathbf{v}_m + \alpha\mathbf{z} + \gamma\mathbf{q}\right) + \hat{\mathbf{u}} \\
&= \beta\left(\left(\frac{\alpha}{\sqrt{M}} - 1\right)\sum_{m=1}^{M}\mathbf{v}_m + \alpha\mathbf{z}\right) - (1-\beta\gamma)\mathbf{q} + \mathbf{u}.
\end{aligned}$$

This estimate gives the following mean-squared error:

$$D = \beta^2\left(\left(\frac{\alpha}{\sqrt{M}} - 1\right)^2 M^2P + \alpha^2\sigma_Z^2\right) + (1-\beta\gamma)^2\sigma_Q^2.$$

We define the following constants:

$$\begin{aligned}
\alpha &= \frac{MP\sqrt{M}}{MP + \sigma_Z^2} \\
\gamma_0 &= \sqrt{\frac{MP}{\sigma_Q^2}\left(1 - \frac{M\sigma_Z^2}{MP + \sigma_Z^2}\right)},
\end{aligned}$$

and let $\gamma \to \gamma_0$ from below as $k \to \infty$. This ensures that Equation (6.15) is always satisfied. We also set:

$$\beta = \frac{\sigma_Q^2\gamma}{MP}.$$

As $k \to \infty$, we get that the achieved distortion is:

$$D = M\sigma_S^2\frac{\sigma_Z^2}{\sigma_Z^2 + MP}\frac{M\sigma_Z^2}{\sigma_Z^2 + MP}. \tag{6.16}$$

This proves the theorem for $\ell = 2$. For all higher values of ℓ, the scheme can be repeated with the final estimate from the last refinement taken as side information for the next stage. □

Remark 28. A simple achievable scheme for situations where we do not have an integer number of channel uses per source symbol is to time share between two integers whose average gives the proper ratio.

Remark 29. This joint source-channel scheme can be easily generalized so that we can send a linear function of Gaussian sources instead of just a sum. Essentially, the γ coefficient in (6.13) should be replaced by $\phi_m \gamma$ at encoder j where ϕ_m is the desired coefficient for that source ($U = \sum_{m=1}^{M} \phi_m S_J$).

We now show that if we want to communicate the sum to the receiver through a separation-based strategy, the encoders can do no better than send their sources to the decoder.

Lemma 21. *There are M source encoders, $\mathcal{E}_m : \mathbb{R}^k \rightarrow \{1, 2, \ldots, 2^{nR_m}\}$, each observing one of the sources S_m^k and compressing it into bits. A decoder, $\mathcal{D} : \{1, 2, \ldots, 2^{nR_1}\} \times \cdots \times \{1, 2, \ldots, 2^{nR_M}\} \rightarrow \mathbb{R}^k$, is given these bits and makes an estimate $\hat{U}^k$ of the sum U^k. The minimum total rate required to reconstruct the sum at distortion D is*

$$\sum_{m=1}^{M} R_m = \frac{M}{2} \log\left(\frac{M\sigma_S^2}{D}\right). \tag{6.17}$$

Proof. (*Converse.*) Let $W_m = \mathcal{E}_m\left(S_m^k\right)$ be the message output by the m^{th} encoder for a length k block of source symbols. Given $\mathbf{W} = (W_1, W_2, \ldots, W_M)$ at the decoder, the minimum-mean squared estimate (MMSE) of U is given by the conditional expectation.

$$D = \frac{1}{k}\sum_{i=1}^{k} E[(U[i] - \hat{U}[i])^2] \tag{6.18}$$

$$\geq \frac{1}{k}\sum_{i=1}^{k} E[(U[i] - E[U(i)|\mathbf{W}])^2] \tag{6.19}$$

$$\overset{\text{(a)}}{=} \frac{1}{k}\sum_{i=1}^{k} E\left[\left(\sum_{m=1}^{M} S_m[i] - E\left[\sum_{m=1}^{M} S_m[i] \middle| \mathbf{W}\right]\right)^2\right] \tag{6.20}$$

$$\overset{\text{(b)}}{=} \frac{1}{k}\sum_{i=1}^{k} E\left[\left(\sum_{m=1}^{M} S_m[i] - E\left[\sum_{m=1}^{M} S_m[i] \middle| W_m\right]\right)^2\right] \tag{6.21}$$

$$\overset{(c)}{=} \frac{1}{k}\sum_{m=1}^{M}\sum_{i=1}^{k} E[(S_m[i] - E[S_m[i]|W_m])^2] \tag{6.22}$$

$$\overset{(d)}{\geq} \sum_{m=1}^{M} \sigma_S^2 2^{-2R_m} \tag{6.23}$$

(a) by linearity of expectation
(b), (c) by independence of S_{m_1} and S_{m_2} for all $m_2 \neq m_1$
(d) by the single source rate distortion converse (see [29, pp. 315-318])

Minimizing the function $\sum_{m=1}^{M} \sigma_S^2 2^{-2R_m}$ is just a convex optimization problem subject to the convex constraint $\sum_{m=1}^{M} R_m = R$. It easily follows that the minimizing solution satisfies $R_1 = R_2 = \cdots = R_M$. We obtain:

$$D \geq M\sigma_S^2 2^{-2\frac{R}{M}}$$
$$R(D) \geq \frac{M}{2}\log\left(\frac{M\sigma_S^2}{D}\right).$$

(*Achievability.*) Each encoder simply uses a standard Gaussian rate distortion code for its source with distortion target $D_m = \frac{D}{M}$. Such a code requires a rate of at least $\frac{1}{2}\log\left(\frac{M\sigma_S^2}{D}\right)$ per encoder. See [29, pp. 318-325] for the derivation of such a code. The decoder recovers each source and sums the individual estimates to get an estimate of the desired sum at distortion D. □

Theorem 29. *The best achievable distortion for a separation-based scheme for sending a Gaussian sum over a Gaussian MAC is*

$$D_{SEP} = M\sigma_S^2\left(\frac{\sigma_Z^2}{\sigma_Z^2 + MP}\right)^{\ell/M}. \tag{6.24}$$

Proof. From Lemma 21, we know the minimum sum rate required. We also know that the maximum sum rate of a Gaussian multiple-access channel is:

$$\frac{1}{2}\log\left(1 + \frac{MP}{\sigma_Z^2}\right). \tag{6.25}$$

Recall that we are allocated ℓ channel uses per source symbol. Thus, any separation-based scheme must satisfy:

$$\frac{M}{2}\log\left(\frac{M\sigma_S^2}{D}\right) \leq \frac{\ell}{2}\log\left(1+\frac{MP}{\sigma_Z^2}\right). \tag{6.26}$$

Solving for D gives the desired result. □

Computation coding can fully utilize the extra channel bandwidth while separation-based coding must split the extra bandwidth between the M users. In Figure 6.2, we plot the distortion for repetition coding (6.10), separation-based coding (Theorem 29), computation coding (Theorem 28), and our lower bound (Lemma 19) for $M = 5$ transmitters with source variance $\sigma_S^2 = 1$ communicating over a channel with power $P = 3$ and noise variance $\sigma_Z^2 = 1$. The logarithm of the distortion is plotted versus the number of channel uses per source symbol. As the channel bandwidth increases, computation coding performs exponentially better than the other strategies.

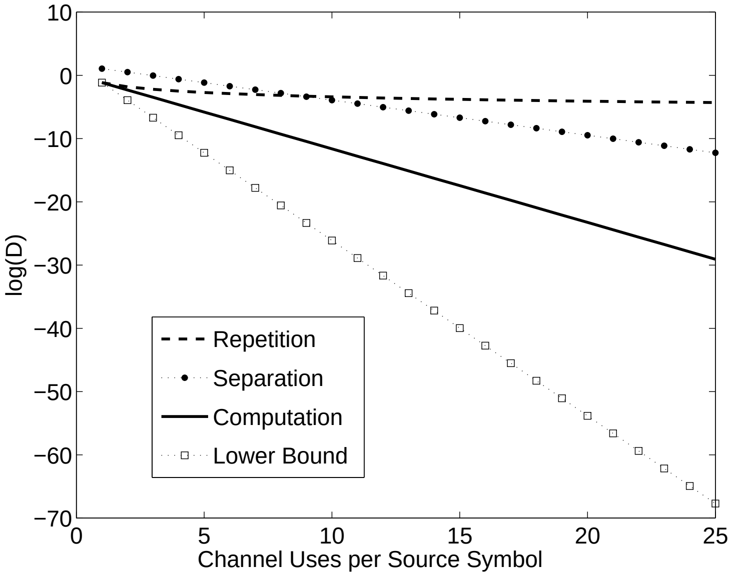

Figure 6.2: Refining the Sum of Gaussian Sources over a Gaussian MAC, $M = 5$, $P = 3$, $\sigma_Z^2 = 1$, $\sigma_S^2 = 1$

Recent work by Soundararajan and Vishwanath has examined a related problem where two transmitters want to send the difference of correlated Gaussian sources over a Gaussian MAC [147]. They showed that by using a common dither at both encoders, the performance of the lattice-based scheme can be improved.

6.2 Sensor Network Case Study

We now apply the computation code developed above to a simple sensor network model first proposed by Gastpar and Vetterli [54; 55].[2] The setup is essentially the same as in the previous section except that instead of independent sources, each sensor observes a noisy version of a single source (see Figure 6.3). We would like to reconstruct this source at the lowest possible distortion at the receiver. Specifically, the desired source U^k is drawn i.i.d. according to a Gaussian distribution with mean zero and variance σ_U^2. Each sensor observes a version of this source corrupted by i.i.d. Gaussian noise W^k with mean zero and variance σ_W^2:

$$S_m[i] = U[i] + W_m[i]. \tag{6.27}$$

We are given $n = k\ell$ channel uses and the goal is to make an estimate $\hat{U}^k$ at the receiver with the lowest possible distortion:

$$\frac{1}{k}\sum_{i=1}^{k} E\left[(U[i] - \hat{U}[i])^2\right] \leq D \tag{6.28}$$

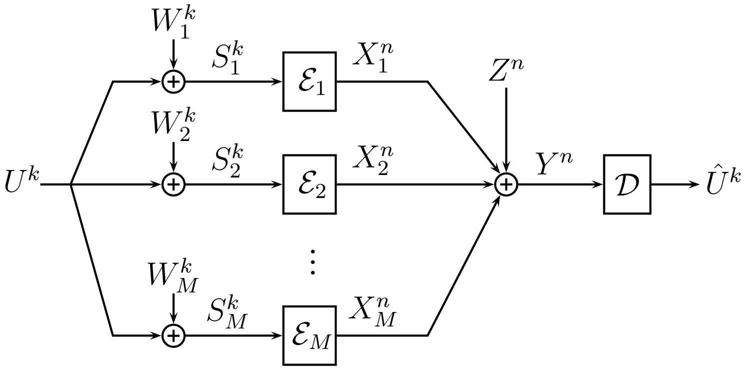

Figure 6.3: A simple Gaussian sensor network model.

6.2.1 A Converse Bound

We can slightly extend a bound first presented in [52] to obtain the following theorem:

[2] This section originally appeared as [107].

Theorem 30. *For the Gaussian sensor network, the incurred distortion must satisfy*

$$D_{LOWER} \geq \frac{\sigma_U^2 \sigma_W^2}{M\sigma_U^2 + \sigma_W^2} \left(1 + M \frac{\sigma_U^2}{\sigma_W^2} \left(\frac{\sigma_Z^2}{\frac{M\sigma_U^2 + \sigma_W^2}{\sigma_U^2 + \sigma_W^2} MP + \sigma_Z^2} \right)^{\ell} \right). \tag{6.29}$$

6.2.2 Separation-Based Coding

We will first consider a separation-based approach. The source coding problem corresponding to our sensor network example has been well studied, under the name of the *CEO problem.* This problem was introduced in [14; 154] and the quadratic Gaussian version described above was solved by Oohama [116], with some recent refinements [123; 27]. From this work, the *sum* rate (i.e., the total rate over all M encoders) in order to achieve a certain distortion D is determined as

$$R(D) = \log_2^+ \left(\frac{\sigma_U^2}{D} \left(\frac{D\sigma_U^2 M}{D\sigma_U^2 M - \sigma_U^2 \sigma_W^2 + D\sigma_W^2} \right)^M \right). \tag{6.30}$$

For our considerations, we will use Oohama's simpler lower bound, which can be obtained easily from the above, noting that $\sigma_U^2/D \geq 1$,

$$R(D) \geq M \log_2^+ \left(\frac{D\sigma_U^2 M}{D\sigma_U^2 M - \sigma_U^2 \sigma_W^2 + D\sigma_W^2} \right). \tag{6.31}$$

Conversely, the smallest achievable distortion satisfies

$$D(R) \geq \frac{\sigma_U^2 \sigma_W^2}{\sigma_U^2 M \left(1 - 2^{-R/M}\right) + \sigma_W^2}. \tag{6.32}$$

By noting that $1 - 2^{-R/M} \leq R/M$, this implies the lower bound

$$D(R) \geq \frac{\sigma_U^2 \sigma_W^2}{\sigma_U^2 R + \sigma_W^2}. \tag{6.33}$$

The total communication rate across the multiple-access channel in our system can be somewhat generously bounded by

$$R_{tot} \quad \leq \quad \frac{\ell}{2} \log_2 \left(1 + \frac{M^2 P}{\sigma_Z^2} \right), \tag{6.34}$$

where we recall that ℓ is the (average) number of channel uses per source sample. Note that the M^2 factor is due to the correlated observations made by the sensors. Using these correlations, the sensors could potentially generate channel inputs that combine coherently. Combining this with the source coding bound yields the following result.

Theorem 31. *For the Gaussian sensor network, a separation-based scheme incurs a distortion of at least*

$$D_{SEP} \geq \frac{\sigma_U^2 \sigma_W^2}{\ell \frac{\sigma_U^2}{2} \log_2 \left(1 + M^2 P / \sigma_Z^2\right) + \sigma_W^2}. \tag{6.35}$$

6.2.3 Uncoded Transmission

For the special case $\ell = 1$ (equal bandwidth), the simple sensor network has been thoroughly investigated. Gastpar showed in [52] that an optimal strategy is for each sensor to transmit its observation in an uncoded fashion:

$$X_m[n] \quad = \quad \sqrt{\frac{P}{\sigma_U^2 + \sigma_W^2}} U_m[n]. \tag{6.36}$$

Earlier work by Gastpar and Vetterli established that uncoded transmission is asymptotically optimal as the number of users goes to infinity [53; 54; 55].

Theorem 32. *For the Gaussian sensor network with $\ell = 1$, uncoded transmission attains the smallest possible distortion, given by*

$$D_{UNC} = \frac{\sigma_U^2 \sigma_W^2}{M \sigma_U^2 + \sigma_W^2} \left(1 + \frac{M(\sigma_U^2 \sigma_Z^2 / \sigma_W^2)}{\frac{M\sigma_U^2 + \sigma_W^2}{\sigma_U^2 + \sigma_W^2} MP + \sigma_Z^2} \right). \tag{6.37}$$

For a proof, see [52]. As before, repetition coding is unable to fully exploit the extra channel uses:

$$D_{\text{REP}} = \frac{\sigma_U^2 \sigma_W^2}{M \sigma_U^2 + \sigma_W^2} \left(1 + \frac{M(\sigma_U^2 \sigma_Z^2 / \sigma_W^2)}{\frac{M\sigma_U^2 + \sigma_W^2}{\sigma_U^2 + \sigma_W^2} \ell MP + \sigma_Z^2} \right). \tag{6.38}$$

6.2.4 Structured Codes

As the converse bound in Theorem 30 shows, we would ideally like the distortion to fall *exponentially* with increasing channel bandwidth (or increasing ℓ). However, repetition coding only provides a linear descent so we must turn to more clever strategies for $\ell > 1$. We now derive the performance of our computation code.

Theorem 33. *For the Gaussian sensor network, the following distortion is achievable for any $\ell > 1$:*

$$D_{LAT} = \frac{\sigma_U^2\sigma_W^2}{M\sigma_U^2+\sigma_W^2}\left(1+\left(\frac{M(\sigma_U^2\sigma_Z^2/\sigma_W^2)}{\frac{M\sigma_U^2+\sigma_W^2}{\sigma_U^2+\sigma_W^2}MP+\sigma_Z^2}\right)\left(\frac{M\sigma_Z^2}{MP+\sigma_Z^2}\right)^{\ell-1}\right).$$

Proof. (Sketch.) We first use uncoded transmission to communicate our observation sequences across the channel to get an MMSE estimate of their sum $\sum_{m=1}^{M} S_m[i]$ at distortion $D_1^{(U)}$ where

$$D_1^{\text{SUM}} = \frac{(M^2\sigma_U^2 + M\sigma_W^2)\sigma_Z^2}{\frac{P}{\sigma_U^2+\sigma_W^2}(M^2\sigma_U^2+M\sigma_W^2)+\sigma_Z^2} \tag{6.39}$$

Denote this MMSE estimate of the sum of observations by $V^{(1)}[i]$.

Now we employ our lattice-based scheme from Theorem 28 to refine this estimate of our sum with the remaining $(\ell-1)k$ channel uses. Note that due to the dither step in the proof of Theorem 28, we do not require the sources to be independent. We can reduce our distortion down to D_ℓ^{SUM} where

$$D_\ell^{\text{SUM}} = D_1^{\text{SUM}}\left(\frac{M\sigma_Z^2}{MP+\sigma_Z^2}\right)^{\ell-1} \tag{6.40}$$

We then use this estimate of the sum of observations to make an MMSE estimate $E[\mathbf{u}|\mathbf{v}^{(1)}]$ of the original source S. Let $\mathbf{s}_{\text{SUM}} = \sum_{m=1}^{M}\mathbf{s}_m$. The distortion for this estimate is given by:

$$\begin{aligned}
D_\ell &= \frac{1}{N}E\left[\|\mathbf{u}-E[\mathbf{u}|\mathbf{v}^{(\ell)}]\|^2\right] && (6.41)\\
&= \frac{1}{N}E\left[\|\mathbf{u}-E[\mathbf{u}|\mathbf{s}_{\text{SUM}}]+E[\mathbf{u}|\mathbf{s}_{\text{SUM}}]-E[\mathbf{u}|\mathbf{v}^{(\ell)}]\|^2\right]\\
&\overset{(a)}{=} \frac{1}{N}E\left[\|\mathbf{u}+E[\mathbf{u}|\mathbf{s}_{\text{SUM}}]\|^2\right]+\frac{1}{N}E\left[\|E[\mathbf{u}|\mathbf{s}_{\text{SUM}}]-E[\mathbf{u}|\mathbf{v}^{(\ell)}]\|^2\right]\\
&\overset{(b)}{=} \frac{\sigma_U^2\sigma_W^2}{M\sigma_U^2+\sigma_W^2}+\left(\frac{\sigma_U^2}{M\sigma_U^2+\sigma_W^2}\right)^2\frac{1}{N}E\left[\|\mathbf{s}_{\text{SUM}}-E[\mathbf{s}_{\text{SUM}}|\mathbf{v}^{(\ell)}]\|^2\right]\\
&= \frac{\sigma_U^2\sigma_W^2}{M\sigma_U^2+\sigma_W^2}+\left(\frac{\sigma_U^2}{M\sigma_U^2+\sigma_W^2}\right)^2 D_\ell^{\text{SUM}} && (6.42)
\end{aligned}$$

where (a) follows by the orthogonality principle, and (b) is due to the fact that MMSE estimation for Gaussian sources is just a rescaling.

□

As desired, we now have a scheme for which distortion falls exponentially with increasing ℓ. Unfortunately, its performance does not match that of our lower bound from Theorem 30. It is unclear whether our scheme can be significantly improved upon or that there is a fundamental penalty for distributed encoding beyond the $\ell = 1$. It seems likely that any scheme that employs quantization at the encoders will face a penalty that keeps it away from the lower bound.

One shortcoming of our scheme is that it only results in a reduction in distortion for $\ell > 1$ if an SNR requirement is satisfied ($\frac{P}{\sigma_Z^2} > 1 - \frac{1}{M}$). This can be overcome by combining repetition coding with the lattice scheme in Theorem 33. For instance, if each transmission is repeated $\theta \in \mathbb{Z}_+$ times then we can run $\frac{\ell-1}{\theta}$ refinements to get the following distortion (assume $\frac{\ell-1}{\theta} \in \mathbb{Z}_+$):

$$D_\ell = \frac{\sigma_U^2\sigma_W^2}{M\sigma_U^2 + \sigma_W^2}\left(1 + \left(\frac{M(\sigma_U^2\sigma_Z^2/\sigma_W^2)}{\frac{M\sigma_U^2+\sigma_W^2}{\sigma_U^2+\sigma_W^2}MP + \sigma_Z^2}\right)\left(\frac{M\sigma_Z^2}{\theta MP + \sigma_Z^2}\right)^{\frac{\ell-1}{\theta}}\right).$$

6.3 Local Interference Can Accelerate Gossip Algorithms

Gossip algorithms are a completely decentralized approach to computing a global function, such as the average. These algorithms can be used as a key component in constructing more complicated signal processing and optimization algorithms on networks. The gossip protocol is quite simple to describe: a sensor randomly wakes up itself and a neighbor and they replace their current values with their local pairwise average. This process continues until all nodes converge to within an acceptable distance from the true average. Boyd et al. [18] give a comprehensive analysis of the convergence speed for gossip algorithms for any connectivity graph. The convergence time is connected to the mixing time of a Markov chain on the graph induced by the sensor network communication ranges.

Clearly, if there was no energy penalty for long-range wireless transmissions, each sensor would broadcast its observation to the entire network and there would be no advantage to gossiping locally. However, such transmissions are expensive in terms of the energy required, and in addition generate significant interference which can delay the averaging process. We will show that with computation codes, we can exploit the interference to average over a neighborhood of sensors in one shot. This form of computation code relies on a certain minimum channel knowledge at the transmitters and we capture this by the somewhat simplistic model of the *local neighborhood* of a node. We assume that within a local neighborhood, each node knows its respective channel (fading) parameters towards the center node of the neighborhood. We further assume that within this local neighborhood, nodes can operate in a synchronous manner.

Outside of the local neighborhood, no channel state information or synchronization is required. The size of the local neighborhood is determined by the spatial and temporal coherence of the particular wireless infrastructure at hand. Our analysis suggests that if local neighborhoods of a certain size are physically possible, computation codes can yield exponentially large savings in the required energy, for a fixed averaging time.

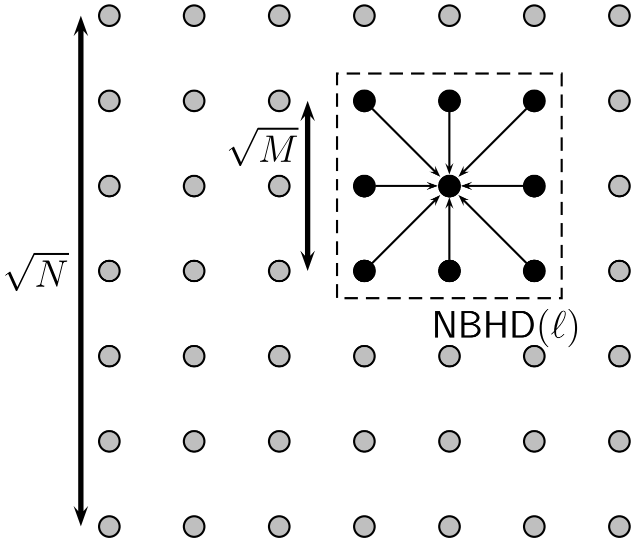

Figure 6.4: Node ℓ efficiently collects the average from its local neighborhood, NBHD(ℓ), using a computation code.

Computation coding allows us to *reliably* add numbers in a local neighborhood with concurrent transmissions over noisy channels. We will use these neighborhood averages as part of a gossip algorithm on a sensor network. In each round of the algorithm, one randomly selected node will wake up, collect the average from its local neighborhood, and distribute the result back to its local neighborhood. First, we show that if each gossip round is computing averages over a larger neighborhood, we can dramatically reduce the number of required gossip rounds. At one end of the scale is the case where the spatial and temporal coherence is so good that the "local" neighborhood includes, in fact, the entire network, and thus, consensus is achieved in a single "neighborhood gossip" step. At the other end of the scale is the case where there is almost no coherence at all, and thus, the local neighborhood only includes the nearest neighbor, and we are back to standard nearest-neighbor gossip. The interesting question is for which neighborhood size is there a benefit to using interference.

When averaging over some large neighborhood, sensors will have to transmit over longer distances and will have to operate at a higher power level, to overcome path loss. The key idea is that since the neighborhood gossip algorithm requires fewer rounds to converge, *each round can afford to take more time*, which can, under some conditions, yield a reduction in the total energy consumption. As we show, if we allow more time to the nearest-neighbor gossip it will always consume less energy. However, when we fix the total convergence time, neighborhoods that are large enough will yield exponential energy gains.

We perform our analysis on the simple topology of a grid network.[3] Our techniques can be extended to more realistic models of wireless network topologies like random geometric graphs (which can possibly change the results up to polylogarithmic factors) but in this section we will only address the simplest case.

Related Work

Distributed averaging can be used as a fundamental building block for *distributed signal processing* over networks, where the goal is to achieve a global objective (e.g., computing the global average of all observations) based on purely local computations (in this case, message-passing between pairs of adjacent nodes). Deterministic variations of gossip algorithms (i.e. each node communicating with all the one-hop neighbors as opposed to a randomly selected one) are often called *consensus algorithms* and their behavior and analysis are very similar. Gossip and consensus averaging is very useful because it can be easily converted into a more general algorithm that computes any linear projection of the sensor measurements (as long as each sensor knows the corresponding coefficient of the projection vector). Recently, such algorithms have been proposed for distributed filtering and optimization as well as distributed detection in sensor networks [148; 158; 128].

In a series of papers [18; 17], Boyd et al. have analyzed the performance of standard gossip algorithms on arbitrary graphs and shown how the gossip parameters can be optimized by solving an optimization problem to reduce convergence time. Unfortunately, for graphs that correspond to realistic sensor network topologies (like grids or random geometric graphs) standard gossip algorithms (even with optimal parameters) are very inefficient and require $\Theta(N^2)$ radio transmissions to converge where N is the number of sensors.

Mosk-Aoyama and Shah [94] use an algorithm based on the work of Flajolet and Martin [46] to compute averages and bound the averaging time in terms of a "spreading time" associated with the communication graph. Dimakis, Sarwate, and Wainwright [34] proposed a modified gossip algorithm that uses geographic information of the sensors to reduce the convergence time to $O(N^{1.5}\sqrt{\log N})$ for random geometric graphs. Very similar performance can also be achieved with only partial geographic information as shown by Li and Dai [85]. Geographic gossip was subsequently used to compute random linear projections and perform

[3] This section originally appeared as [103].

distributed compressive sensing [124] for sensor network measurements. Benezit et al. [13] showed that an extension of geographic gossip that averages along the routed paths can further reduce the convergence time to $O(N \log N)$ which is optimal for random geometric graphs and grids. In this work we assume that no geographic information is available at the nodes so such schemes are not applicable.

The issue of noise and quantization in the gossip messages has received significant attention recently [120; 10; 112; 125] and schemes that achieve quantized consensus and tight convergence bounds can be found in these papers. Sundaram and Hadjicostis [150] show how infinite accuracy can be achieved in finite number of rounds by extrapolating the consensus value through appropriate computation.

Other groups have also studied how to best exploit the physical-layer for consensus. For instance, Aysal et al. exploit the broadcast aspect in [11] and Kirti, Scaglione, and Thomas exploit the multiple-access aspect in [73]. Dimakis and Sarwate characterized the impact of sensor mobility on gossip convergence in [134].

6.3.1 Problem Statement

6.3.1.1 Wireless Channel Model

There is a sensor network composed of N nodes. Each node has a unique index $\ell \in \{1, 2, \ldots, N\}$ and a unique position $p \in \{1, 2, \ldots, \sqrt{N}\} \times \{1, 2, \ldots, \sqrt{N}\}$ on the extended grid. We assume that the wireless channel has a finite bandwidth so a discrete-time model is sufficient and we index time (or channel uses) using i. At time i, the received signal at node ℓ is:

$$Y_\ell[i] = \sum_{k \in \mathsf{NBHD}(\ell)} h_{\ell k}[i] X_k[i] + Z_\ell[i] \tag{6.43}$$

$$h_{\ell k} = r_{\ell k}^{-\frac{\alpha}{2}} e^{j\theta_{\ell k}[i]} \tag{6.44}$$

where $r_{\ell k}$ is distance between nodes ℓ and k, $\alpha \in \mathbb{R}_+$ is the power path loss coefficient, the $\theta_{\ell k}[i]$ are phases chosen randomly according to some distribution over the interval $[0, 2\pi]$, $X_k[i]$is the signal transmitted by the k^{th} node at time i, and $z_\ell[i]$ is i.i.d. Gaussian noise with mean zero and variance σ_Z^2.

Finally, $\mathsf{NBHD}(\ell) \subset \{1, \ldots, N\}$ is the local neighborhood of node ℓ. For ease of analysis, we assume that the local neighborhood are the nodes in the $\sqrt{M}$ by $\sqrt{M}$ square around node ℓ.[4] Ignoring boundary effects, each local neighborhood contains M nodes. In general, nodes do not know the phases, $\theta_{lk}[i]$, governing the channel to other nodes in the wireless network. However, we will assume that nodes do know the channels in their local neighborhood.

[4] Since we are only interested in the scaling law, we can safely ignore integer effects, i.e. assume that $\sqrt{M}$ is always odd and that $\sqrt{\frac{N}{M}}$ is an integer.

6.3.1.2 Time Model

We will assume that the nodes wake up according to the asynchronous time model in [18]. Each node observes a rate λ Poisson process and wakes up upon an arrival. The rate can be set such that no two nodes wake up in a given time interval with high probability. We also assume that the nodes are completely synchronized with respect to their channel uses; the Poisson clocks only determine when they wake up.

Furthermore, we will count time on two scales, channel uses and gossip rounds, to avoid confusion between our channel code and our gossip algorithm. Gossip rounds are simply a count of how many steps the gossip algorithm has taken (see Definition 60). We assume that within each round we have T_R channel uses.

6.3.1.3 Distributed Averaging

We now provide a precise notion of convergence for a gossip algorithm. First, we will review the standard formulation used in the literature. Since we are including noisy channels in our analysis, we must use long blocklengths to ensure reliable communication. Thus, we will allow for a vector of observations at each node, rather than a scalar, and this will allow us to communicate in a reliable fashion.

6.3.1.4 Standard Formulation

The standard formulation of a gossip algorithm is as follows. Each node k starts out with a scalar observation $s_k[0] \in \mathbb{R}$ for $k = 1, 2, \ldots, N$. Our goal is to have each node learn the global average of these observations:

$$s_{\text{AVG}} = \frac{1}{N} \sum_{k=1}^{N} s_k[0] \tag{6.45}$$

At time t, node k has an estimate $s_k[t]$ of the global average. Let $\mathbf{s}[t]$ denote the N-vector of these estimates at round t.

Definition 59. Choose $\epsilon > 0$. Let $R^{\text{AVG}}(N, \epsilon)$ be the minimum number of gossip rounds required to get all nodes estimates of the average to within ϵ of the true average with probability greater than $1 - \epsilon$.

$$R_{\text{AVG}}(N, \epsilon) = \sup_{\mathbf{s}[0]} \inf \left\{ t : \mathbb{P}\left(\frac{\|\mathbf{s}[t] - s_{\text{AVG}} \overrightarrow{1}\|}{\|\mathbf{s}[0]\|} \geq \epsilon \right) \leq \epsilon \right\}$$

6.3.1.5 Vector Formulation

We slightly modify the standard gossip problem statement by having each node k start out with a length-L vector observation $\mathbf{v}_k = (s_{k1},\ s_{k2}, \ldots,\ s_{kL}) \in \mathbb{R}^L$ for $k = 1, 2, \ldots, N$. Our goal is now to have each node learn the global average of these vectors:

$$\mathbf{v}_{\mathrm{AVG}} = \left(\frac{1}{N} \sum_{k=1}^{N} s_{k1}[0], \ldots, \frac{1}{N} \sum_{k=1}^{N} s_{kL}[0] \right) \tag{6.46}$$

To ensure finite transmission energies, we will also assume that the measurement vectors $\mathbf{v}_k$ have bounded ℓ_2 norm:

$$\|\mathbf{v}_k\|^2 \leq \Gamma L \tag{6.47}$$

where $\Gamma \in \mathbb{R}_+$ is a constant.

At time t, node k has an estimate $s_{kq}[t]$ of the global average of the q^{th} element. Let $\mathbf{s}_q[t]$ denote the N-vector of these estimates at round t. We use the following definition for convergence of the vector gossip algorithm.

Definition 60. Choose $\epsilon > 0$. Let $R^{\mathrm{AVG}}(N, M, \epsilon)$ be the minimum number of gossip rounds with neighborhood size M required to get all nodes estimates of the average vector to within ϵ of the true average with probability greater than $1 - \epsilon$.

$$\beta = \frac{\sum_{q=1}^{L} \|\mathbf{s}_q[t] - s_{\mathrm{AVG}q} \overrightarrow{1}\,\|}{\sum_{q=1}^{L} \|\mathbf{s}_q[0]\|} \tag{6.48}$$

$$R_{\mathrm{AVG}}(N, M, \epsilon) = \sup_{\mathbf{s}_q[0]} \inf \left\{ t : \mathbb{P}\left(\beta \geq \epsilon\right) \leq \epsilon \right\} \tag{6.49}$$

The total time spent by our algorithm is easily computed by multiplying the number of gossip rounds by the amount of channel uses used per gossip round T_R. However, it may be possible to schedule multiple gossip rounds simultaneously and therefore we divide this quantity by reuse factor $\mathcal{F}$:

$$T_{\mathrm{TOTAL}} = \frac{T_R R_{\mathrm{AVG}}(N, M, \epsilon)}{\mathcal{F}}. \tag{6.50}$$

Note that the reuse factor might be different for different neighborhood sizes and we bound this quantity in a subsequent section.

6.3.1.6 Energy Model

We assume that energy consumption is dominated by wireless transmissions and measure total energy consumption, E_{TOTAL}, by the sum of of the squared amplitudes of all transmissions in the network:

$$E_{\text{TOTAL}} = \sum_{i=1}^{T_{\text{TOTAL}}} \sum_{\ell=1}^{N} (x_\ell[i])^2 \tag{6.51}$$

By construction, each gossip round will consume the same amount of energy, E_{R}. Thus, the total energy consumption can also be computed by multiplying this quantity by the number of gossip rounds:

$$E_{\text{TOTAL}} = E_{\text{R}} R_{\text{AVG}}(N, M, \epsilon) \tag{6.52}$$

6.3.1.7 Time-Energy Tradeoff

Our goal is to minimize both the total time and the transmit energy cost for making the global average available at each node. Clearly, there is a tradeoff between these two quantities. Intuitively, if we demand the average in smaller amount of time, it will cost more energy. Thus, our goal is to find the best possible time-energy tradeoff curve and the algorithm that provides it. In the next section, we will provide a high-level description of our gossip algorithm.

6.3.2 Algorithm Sketch

Our algorithm operates at two levels of abstraction: At the higher level, we show how to select a good sequence of "neighborhood gossip" rounds in such a way as to attain global consensus as quickly as possible. More precisely, we show that a random sequence of uniformly chosen nodes performs well with high probability. At the lower level, we provide (physical-layer) algorithms that permit to efficiently perform "neighborhood gossip," exploiting the structure and coherence of the local interference, and leading to local consensus within the neighborhood. In this section, we give an overview and rough outline of the two key steps in the resulting "neighborhood gossip" algorithm.

6.3.2.1 Neighborhood Gossip

Assume node ℓ wakes up for the t^{th} gossip round. The following steps describe the gossip round:

1. Node ℓ wakes up all of the nodes in its local neighborhood, $\mathsf{NBHD}(\ell)$.

2. All nodes in the local neighborhood transmit their estimates to node ℓ using a computation code. The computation code is designed such that node ℓ receives only the average of these values.

3. Node ℓ uses the received information and its own value to compute the average of the estimates from its local neighborhood. It replaces its current estimate for the q^{th} element with this new estimate for the next gossip round:

$$s_{\ell q}[t+1] = \frac{1}{M} \sum_{k \in \mathsf{NBHD}(\ell)} s_{kq}[t] \tag{6.53}$$

   for $q = 1, 2, \ldots, L$.

4. Node ℓ broadcasts its updated estimate to all nodes in its local neighborhood. All local neighborhood nodes replace their current estimate with the transmitted one for the next gossip round:

$$s_{uq}[t+1] = \frac{1}{M} \sum_{k \in \mathsf{NBHD}(\ell)} s_{kq}[t] \quad \forall u \in \mathsf{NBHD}(\ell) \tag{6.54}$$

As one might expect, the convergence time of such an algorithm is highly dependent on the topology of the network and the choice of the local neighborhoods. In Section 6.3.3, we will examine a network where the nodes are placed on a $\sqrt{N} \times \sqrt{N}$ extended grid and the local neighborhoods are squares of size $\sqrt{M} \times \sqrt{M}$ centered around the nodes and show that the algorithm converge in $O\left(\frac{N^2}{M^2} \log\left(\frac{1}{\epsilon}\right)\right)$ rounds.

6.3.2.2 Computation Coding

The critical step in the neighborhood gossip algorithm is Step 2 in the description given above: All nodes in the local neighborhood need to communicate to the center node. It may be tempting at first to implement this using some form of orthogonal accessing where each node communicates to the center node on a separate channel. However, this approach would consume virtually all the potential advantages of neighborhood gossip. The key insight is that the center node does not need to know the exact data at each of the nodes in the neighborhood. Rather, it only needs to know the *average*. We show how this can be achieved very efficiently using computation codes. To give an intuition as to where this efficiency is coming from, consider the following two-step procedure:

1. By our definition of a local neighborhood, every node $k \in \mathsf{NBHD}(\ell)$ knows the channel characteristics $(r_{\ell k}, \theta_{\ell k}[i])$ (as in Equations (6.43, 6.44)) from itself to the center node ℓ. Exploiting this knowledge, the nodes in the local neighborhood can transform the

actual multiple-access channel between them and the center node ℓ into the following simple multiple-access channel:

$$Y_\ell[i] = \sum_{k \in \mathsf{NBHD}(\ell) \backslash \{\ell\}} X_{\ell k}[i] + Z_\ell[i]. \tag{6.55}$$

2. (Computation Coding) All nodes simultaneously encode and transmit their values using *identical* linear codebooks. The selected codewords will be added on the channel and node ℓ will receive the sum of the codewords. Since the codebook is linear, the sum of the codewords is also a codeword and is actually the codeword corresponding to the desired average.

In Section 6.3.4, we characterize the tradeoff between the number of channel uses, the precision of the received average, and the expended energy for computation coding.

6.3.3 Neighborhood Gossip on an Extended Grid

Assume that the nodes are placed on an $\sqrt{N} \times \sqrt{N}$ grid with unit distance between both rows and columns. Furthermore, assume that the local neighborhood, $\mathsf{NBHD}(\ell)$, of node ℓ is the $\sqrt{M} \times \sqrt{M}$ square of nodes centered on itself.

Therefore, for each gossip round, a random node ℓ activates and when the round is over, everyone in its local neighborhood $\mathsf{NBHD}(\ell)$ has replaced their value with the average of that neighborhood. In particular, we assume that after the computation coding phase of a gossip round has finished, the average of the local neighborhood estimates of the global average is available at the active node up to precision δ where δ is much smaller than ϵ.

Without loss of generality, we focus on a scalar observation ($L = 1$). Recall that $\mathbf{s}[t]$ is the vector of node estimates of the global average at round t and that $\mathbf{s}[0]$ is just the vector of the nodes' initial observations. Every time a node ℓ activates, all the nodes in $\mathsf{NBHD}(\ell)$ get averaged while other nodes stay invariant, and this can be written compactly as:

$$\mathbf{s}[t+1] = \mathbf{W}[t]\mathbf{s}[t] \tag{6.56}$$

where $\mathbf{W}(t)$ is the matrix that corresponds to averaging nodes in $\mathsf{NBHD}(\ell)$. When the selection of which node activates is i.i.d. random, the corresponding $\mathbf{W}[t]$ matrices for each round t are also i.i.d.

Now let $\bar{\mathbf{W}}$ denote the mean of the i.i.d. $\mathbf{W}[t]$ matrices. The distribution on the random matrices is such that $\bar{\mathbf{W}}$ satisfies the following three properties:

$$\mathbf{1}^T\bar{\mathbf{W}} = \mathbf{1}^T \tag{6.57}$$

$$\bar{\mathbf{W}}\mathbf{1} = \mathbf{1} \tag{6.58}$$

$$\rho\left(\bar{\mathbf{W}} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right) < 1 \tag{6.59}$$

where $\mathbf{1}$ is the all ones vector and $\rho(\cdot)$ is the spectral radius of a matrix.

These conditions guarantee convergence of the gossip algorithm[18] will converge to the true average. The main result of this section is abound on the number of gossip rounds require to converge:

Theorem 34. *The averaging time of neighborhood gossip on an extended grid of size $\sqrt{N} \times \sqrt{N}$, and neighborhoods of size $\sqrt{M} \times \sqrt{M}$ in gossip rounds, satisfies*

$$R^{AVG}(N, M, \epsilon) \leq c\frac{N^2}{M^2}\log\left(\frac{1}{\epsilon}\right), \tag{6.60}$$

where c is a fixed constant.

Proof. To bound the averaging time, we will use the following lemma, that uses the second eigenvalue of the *expected* matrix $\bar{\mathbf{W}}$. The main technical problem is that computing the second eigenvalue of this expected matrix is quite complicated, since even determining the expected matrix itself is not straightforward. We are going to be able to provide a good bound on the second eigenvalue of $\bar{\mathbf{W}}$ without actually computing the entries of the matrix, but rather bounding the *conductance* of $\bar{\mathbf{W}}$.

We begin with a bound connecting the averaging time with the second eigenvalue of $\bar{\mathbf{W}}$:

Lemma 22 (Boyd et al.)**.** *The averaging time in gossip rounds, $R^{AVG}(N, \epsilon)$, of a gossip algorithm is upper bounded by:*

$$R^{AVG}(N, \epsilon) \leq \frac{3\log\left(\epsilon^{-1}\right)}{\log\left(\frac{1}{\lambda_2(\bar{\mathbf{W}})}\right)} \tag{6.61}$$

where $\lambda_2(\bar{\mathbf{W}})$ is the second eigenvalue of the expected matrix $\bar{\mathbf{W}}$.

Since $\log(1 + x) \leq x$ (this bound is tight for small x), we can instead use the following simpler upper bound for our analysis:

$$R^{\mathrm{AVG}}(N, \epsilon) \leq \frac{3\log\left(\epsilon^{-1}\right)}{1 - \lambda_2(\bar{\mathbf{W}})} \tag{6.62}$$

For our gossip algorithm, $\mathbf{W}[t]$ is drawn uniformly from the set $\mathcal{W}$:

$$\mathcal{W} = \{\mathcal{W}(\ell) : \ell = 1, 2, \ldots, N\} \tag{6.63}$$

$$\mathcal{W}(\ell)_{uv} = \begin{cases} \frac{1}{|\mathsf{NBHD}(\ell)|}, & u, v \in \mathsf{NBHD}(\ell); \\ 1, & u = v, \quad u, v \notin \mathsf{NBHD}(\ell); \\ 0, & \text{otherwise.} \end{cases} \tag{6.64}$$

where $\mathcal{W}(\ell)_{uv}$ is the entry in row u and column v in the matrix $\mathcal{W}(\ell)$. Essentially, if node ℓ is chosen, nodes in its local neighborhood compute their local average and the rest keep their values the same. Unfortunately, computing the mean, $\bar{\mathbf{W}}$, of matrices drawn from $\mathcal{W}$ is difficult. However, we will still be able to give bounds on the spectral gap, $1 - \lambda_2(\bar{\mathbf{W}})$. First, we will need a basic linear algebra lemma to connect our matrix, $\bar{\mathbf{W}}$, to one for which we can give a tighter bound on the spectral gap.

Lemma 23. *Let $\bar{\mathbf{W}}_{FAST}$ be chosen such that $\bar{\mathbf{W}} = p_{STAY}\mathbf{I} + (1 - p_{STAY})\bar{\mathbf{W}}_{FAST}$ where $p_{STAY} \in [0, 1]$. Let $\lambda_2(\bar{\mathbf{W}}_{FAST})$ be the second eigenvalue of $\bar{\mathbf{W}}_{FAST}$. Then, the spectral gap of $\bar{\mathbf{W}}$ is given by:*

$$1 - \lambda_2(\bar{\mathbf{W}}) = (1 - p_{STAY})(1 - \lambda_2(\bar{\mathbf{W}}_{FAST})) \tag{6.65}$$

Proof. Let $\mathbf{a}_2$ be the second eigenvector of $\bar{\mathbf{W}}$. We have that:

$$\bar{\mathbf{W}}_{\text{FAST}}\mathbf{a}_2 = \left(\frac{1}{1 - p_{\text{STAY}}}\bar{\mathbf{W}} - \frac{p_{\text{STAY}}}{1 - p_{\text{STAY}}} I\right)\mathbf{a}_2 \tag{6.66}$$

$$= \left(\frac{\lambda_2(\bar{\mathbf{W}}) - p_{\text{STAY}}}{1 - p_{\text{STAY}}}\right)\mathbf{a}_2 \tag{6.67}$$

$$\lambda_2(\bar{\mathbf{W}}_{\text{FAST}}) = \frac{\lambda_2(\bar{\mathbf{W}}) - p_{\text{STAY}}}{1 - p_{\text{STAY}}} \tag{6.68}$$

The lemma follows immediately. □

The above lemma connects our matrix to a matrix that is not "lazy." Now observe that $\bar{\mathbf{W}}$ is a stochastic and symmetric matrix and corresponds to a Markov chain on the $\sqrt{N} \times \sqrt{N}$ grid that is reversible and ergodic. We can therefore use techniques that bound mixing times of Markov chains [57; 13] to bound the spectral gap of $\bar{\mathbf{W}}$.

Definition 61. The conductance [144] of a stochastic matrix $\bar{\mathbf{W}}$ (that corresponds to a reversible Markov chain) is defined by:

$$\Phi(\bar{\mathbf{W}}) = \min_{\substack{\mathcal{S} \subset \{1, \ldots, N\} \\ 0 < \pi(\mathcal{S}) \leq \frac{1}{2}}} \frac{Q_{\mathbf{W}}(\mathcal{S}, \mathcal{S}^C)}{\pi(\mathcal{S})} \tag{6.69}$$

where $Q_{\mathbf{W}}(u,v) = \pi(u)\bar{\mathbf{W}}_{uv} = \pi(v)\bar{\mathbf{W}}_{uv}$, $\pi(\mathcal{S})$ is the probability density of $\mathcal{S}$ under the stationary distribution of π of $\bar{\mathbf{W}}$ and $Q_{\mathbf{W}}(\mathcal{S},\mathcal{S}^C)$ is the sum of $Q_{\mathbf{W}}(u,v)$ over all $(u,v) \in \mathcal{S} \times (\{1,\dots,N\} \setminus \mathcal{S})$

Now we use the fact [144; 57] that conductance can be used to provide a lower bound on the spectral gap:

Lemma 24. *The second eigenvalue of a reversible Markov chain with transition probabilities given by $\bar{\mathbf{W}}$ satisfies:*

$$\frac{1}{1-\lambda_2(\bar{\mathbf{W}})} \leq \frac{2}{(\Phi(\bar{\mathbf{W}}))^2}. \tag{6.70}$$

Finally, we will need a simple fact about conductance given by the following lemma.

Lemma 25. *Let $\bar{\mathbf{V}}$ be a matrix whose off-diagonal elements are less than or equal to those of $\bar{\mathbf{W}}$. Then the conductance of $\bar{\mathbf{W}}$ satisfies the following lower bound:*

$$\Phi(\bar{\mathbf{W}}) \geq \min_{\substack{\mathcal{S}\subset\{1,\dots,N\} \\ 0<\pi(\mathcal{S})\leq\frac{1}{2}}} \frac{Q_{\mathbf{V}}(\mathcal{S},\mathcal{S}^C)}{\pi(\mathcal{S})} \tag{6.71}$$

where $Q_{\mathbf{V}}(u,v) = \pi(u)\bar{\mathbf{V}}_{uv} = \pi(v)\bar{\mathbf{V}}_{uv}$, $\pi(\mathcal{S})$ is the probability density of $\mathcal{S}$ under the stationary distribution of π of $\bar{\mathbf{W}}$ and $Q_{\mathbf{V}}(\mathcal{S},\mathcal{S}^C)$ is the sum of $Q_{\mathbf{V}}(u,v)$ over all $(u,v) \in \mathcal{S} \times (\{1,\dots,N\} \setminus \mathcal{S})$

Proof. Since we are just reducing the numerator in every term inside the minimization then the result is no higher than the original. □

We are now ready to bound the spectral gap of $\bar{\mathbf{W}}$. First, define a non-lazy matrix $\mathbf{W}_{NL}$ with $p_{\text{STAY}} = 1 - \frac{M}{N}$. Each non-diagonal entry of this matrix will be $\frac{1}{M}$ if the indices are in the same neighborhood. Now consider a cut across the center axis of the grid. (It is not hard to see that any other cut will only yield larger conductance.) Clearly, $\pi(\mathcal{S}) = \frac{1}{2}$. We now obtain a *lower bound* on $Q_{\mathbf{W}_{NL}}(\mathcal{S},\mathcal{S}^C)$. Since ignoring nodes and edges only reduces Q, we will only consider the nodes in $\mathcal{S}$ who have distance $\sqrt{M}/4$ or less from the separating axis. There are $\sqrt{N} \times \sqrt{M}/4$ such nodes and each one has at least $\sqrt{M}/2 \times \sqrt{M}/4$ neighbors in $\mathcal{S}^C$. Since all these edges have weight $1/M$ and the stationary distribution is uniform, $\pi(u) = \frac{1}{N}$, we get:

$$Q_{\mathbf{W}_{NL}}(\mathcal{S},\mathcal{S}^C) \geq \frac{\sqrt{N}\sqrt{M}}{4}\frac{\sqrt{M}\sqrt{M}}{8}\frac{1}{N}\frac{1}{M} = \frac{\sqrt{M}}{32\sqrt{N}}, \tag{6.72}$$

so

$$\Phi(\mathbf{W}_{NL}) \geq \frac{M}{16N}. \tag{6.73}$$

which implies that the spectral gap of the non-lazy chain is bounded as follows

$$\frac{1}{1-\lambda_2(\mathbf{W}_{NL})} \leq \frac{2}{(\Phi(\mathbf{W}_{NL}))^2} \leq 512\frac{N}{M}. \tag{6.74}$$

So using to bound the spectral gap of the non-lazy matrix $\bar{\mathbf{W}}$ we simply need to multiply by $\frac{1}{p_{\text{STAY}}} = \frac{N}{M}$ which yields the result.

□

6.3.4 Computation Coding

Inside a local neighborhood, due to the channel knowledge at the transmitters, we can invert the phases and make the channel into a noisy sum:

$$Y_\ell[i] = \left(\frac{1}{M^{\alpha/2}}\right) \sum_{k \in \mathsf{NBHD}(\ell)} X_k[i] + Z_\ell[i] \tag{6.75}$$

where the $M^{-\alpha/2}$ factor comes from considering the worst path loss within the neighborhood.

As mentioned earlier, we will only be attempting to send our averages up to some specified precision. Our error metric for a real number is the usual mean-squared error criterion.

Recall that each sequence of observations has a bounded ℓ_2 norm: $\|\mathbf{v}_k\|^2 \leq L\Gamma$ for $k = 1, 2, \ldots, n$. Finally we say that node k consumes power P if the average energy during a transmission of length T satisfies:

$$\frac{1}{T}\sum_{i=1}^{T} (X_k[i])^2 = P \tag{6.76}$$

Theorem 35. *Choose $\epsilon > 0$. Assume each node in a local neighborhood of size m has a length L bounded real-valued observation vector, $\|\mathbf{v}_k\|^2 \leq \Gamma L$. For L large enough, there exists a coding scheme such that the receiving node can make an estimate $\hat{\mathbf{v}}_{AVG}$ of the average $\mathbf{v}_{AVG} = \frac{1}{M}\sum \mathbf{v}_k$ that satisfies:*

$$Pr\left(\|\mathbf{v}_{AVG} - \hat{\mathbf{v}}_{AVG}\|^2 \geq \frac{\Gamma}{M}2^{-2B}\right) < \epsilon \tag{6.77}$$

so long as:

$$\frac{T}{2}\log\left(\frac{1}{M} + \frac{P}{M^{\frac{-\alpha}{2}}\sigma_Z^2}\right) > B \tag{6.78}$$

for some choice of T channel uses (per observation symbol), power P and precision B bits.

Proof. This follows directly from the proof of Theorem 28 (without the uncoded transmission step). Note that due to the dither step, the sources are not required to be Gaussian, they only need to satisfy second moment bounds (given here by $\|\mathbf{v}_k\|^2 \leq \Gamma L$). Here, the variance of the desired average $\mathbf{v}_{\text{AVG}}$ is $\frac{\gamma L}{M}$ instead of $M\sigma_S^2$ and the distortion D is equivalent to $\frac{1}{L}\|\mathbf{v}_{\text{AVG}} - \hat{\mathbf{v}}_{\text{AVG}}\|^2$. Thus, we just need that:

$$\left(\frac{M\sigma_Z^2}{\sigma_Z^2 + MP}\right)^{\ell} = 2^{-2B}. \tag{6.79}$$

Solving for B yields the desired result. □

Note that this scheme performs significantly better than a standard multiple-access scheme that attempts to inform the receiver of all the individual observation vectors before it computes the sum.

In order to compare our neighborhood scheme, to a nearest neighbor scheme, we need to characterize the resources needed to send a bounded real-valued vector over a Gaussian channel. This easily follows as a corollary of the above theorem.

Corollary 4. *Choose $\epsilon > 0$. Assume node k has a length-L bounded real-valued observation vector, $\|vb_k\|^2 \leq \Gamma L$. A node at distance 1 away needs to make an estimate $\hat{\mathbf{v}}_k$ up to precision B. For L large enough, there exists a coding scheme such that:*

$$Pr\left(\|\mathbf{v}_k - \hat{\mathbf{v}}_k\|^2 \geq \Gamma 2^{-2B}\right) < \epsilon \tag{6.80}$$

so long as:

$$\frac{T}{2}\log\left(1 + \frac{P}{\sigma_Z^2}\right) > B \tag{6.81}$$

for some choice of T channel uses (per observation symbol) and power P.

Of course, the receiving node needs to communicate the average back to the sender(s). However, it can be shown that this only requires a constant factor more energy so we omit it from our analysis.

6.3.5 Performance Comparisons

Now that we have characterized the number of gossip rounds required for neighborhood gossip and the resources required for computation coding, we can determine the scaling laws for both the time and energy consumed by our scheme. First, for comparison purposes, we will calculate the total time (in a scaling law sense) it takes for nearest neighbor gossip to converge on the grid:

$$T_{\text{PAIR}} = c_1 \frac{R_{\text{AVG}}(N, \epsilon) T_1}{\mathcal{F}_{\text{PAIR}}} = c_1 \frac{N^2 T_1}{\mathcal{F}_{\text{PAIR}}}. \tag{6.82}$$

where T_1 is the number of channel uses per gossip round and c_1 is a constant.

Similarly, we can use our result from Theorem 34 to get the total time it takes for neighborhood gossip to converge on the grid.

$$T_{\text{NBHD}} = c_2 \frac{R_{\text{AVG}}(N, M, \epsilon) T_2}{\mathcal{F}_{\text{NBHD}}} = c_2 \frac{N^2}{M^2} \frac{T_2}{\mathcal{F}_{\text{NBHD}}} \tag{6.83}$$

where T_2 is the number of channel uses per gossip round and c_2 is a constant.

We can upper bound $\mathcal{F}_{\text{PAIR}}$ by allowing all N nodes to gossip concurrently in one round of nearest neighbor gossip, $\mathcal{F}_{\text{PAIR}} \leq N$. Also, we can lower bound $\mathcal{F}_{\text{NBHD}}$ by allowing only one neighborhood gossip to take place per round, $\mathcal{F}_{\text{NBHD}} \geq 1$. This clearly is an upper bound on the time savings and we get that the ratio of total time to converge is bounded as follows:

$$\frac{T_{\text{NBHD}}}{T_{\text{PAIR}}} \leq \frac{c_2 T_2 N}{c_1 T_1 M^2}. \tag{6.84}$$

Next, we calculate the total energy E_{PAIR} used by the nearest neighbor gossip scheme. First, we need to determine the energy used in a single gossip round using Corollary 4. Let P_1 be the average power per channel use and B_1 the precision in bits.

$$E_{R1} = T_1 P_1 = T_1 \sigma_Z^2 (2^{2B_1/T_1} - 1) \tag{6.85}$$

where the second step follows from solving for P_1 in Corollary 4.

$$E_{\text{PAIR}} = E_{R1} R_{\text{AVG}}(N, \epsilon) = c_3 N^2 T_1 \sigma_Z^2 (2^{2B_1/T_1} - 1) \tag{6.86}$$

where c_3 is some constant.

Finally, we calculate the total energy E_{NBHD} used by the neighborhood gossip scheme. First, we determine the energy used in a single gossip round using Theorem 35. Let P_2 be the average power per channel use and B_2 the precision in bits.

$$E_{R2} = T_2 P_2 = M^{\alpha/2} \sigma_Z^2 \left(2^{2B_2/T_2} - \frac{1}{M} \right) \tag{6.87}$$

where the second step follows from solving for P_1 in Corollary 4.

One can show that the above expression decreases as the number of channel uses T_2 is increased but then begins to increase again. This is due to the computation coding expending energy to overcome the path loss. At some point these expenditures overcome the savings from using the channel addition. Thus, past this critical T_2 we should not use any more channel uses in a round even if they are allowed.

$$E_{\text{NBHD}} = E_{R2} R_{\text{AVG}}(N, M, \epsilon) = c_4 \frac{N^2}{M^2} M^{\alpha/2} T_2 \sigma_Z^2 \left(2^{2B_2/T_2} - \frac{1}{M} \right)$$

where c_4 is some constant.

Finally, we take the ratio of the total expended energies

$$\frac{E_{\text{NBHD}}}{E_{\text{PAIR}}} = \frac{c_4 T_2}{c_3 T_1} M^{\frac{\alpha}{2}} \frac{2^{2B_2/T_2} - \frac{1}{M}}{2^{2B_1/T_1} - 1}. \tag{6.88}$$

Now we should choose B_1 and B_2 appropriately so that the gossip algorithms converge. As mentioned earlier, noise or quantization effects in gossip algorithms have been the topic of much recent study. For our purposes, we simply assume that if the nearest neighbor gossip uses a constant number of bits of precision in each round, the algorithm is "noise-free", $B_1 \in \mathbb{Z}_+$. Furthermore, we assume that our scheme requires a worst-case $\log N$ bits of precision per round for convergence, $B_2 = c_5 \log N$. This is equivalent to assuming that all of the quantization noises add up linearly (in other words, the noise is adversarial). See [112] for details. This serves as an upper bound on the energy ratio as it can only make our scheme look less favorable.

We will now examine the tradeoff on three operating points. First, we will fix the total time per gossip round by allowing the same number of channel uses per round to each algorithm $T_1 = T_2$. Next, we will fix the total convergence time by setting $T_{\text{NBHD}} = T_{\text{PAIR}}$. Finally, we look at the case where neighborhood gossip converges faster than pairwise gossip, $T_{\text{NBHD}} = N^{-\tau} T_{\text{PAIR}}$, and show where energy savings are possible.

6.3.5.1 Fixed Round Time

Assume that $T_1 = T_2$. In this case, so long as $M > \sqrt{N}$, neighborhood gossip converges faster.

Remark 30. Note that if we assume both algorithms get the same number of concurrent gossips per round ($\mathcal{F}_{\text{PAIR}} = \mathcal{F}_{\text{NBHD}}$), then neighborhood gossip always converges faster (if M increases with N).

The energy ratio can be written as:

$$\frac{E_{\text{NBHD}}}{E_{\text{PAIR}}} = c_6 M^{\alpha/2} \frac{2^{2c_5 \log N/T_1} - \frac{1}{M}}{2^{2B_1/T_1} - 1} = c_6 M^{\alpha/2} \frac{N^{2c_5/T_1} - \frac{1}{M}}{2^{2B_1/T_1} - 1} \tag{6.89}$$

It can be shown that this ratio is always larger than 1. Thus, neighborhood gossip is less energy efficient than nearest neighbor gossip if it is only given the same number of channel uses per gossip round. This is because the computation code must expend extra energy to overcome the long hop to the receiver.

6.3.5.2 Fixed Convergence Time

Assume that $T_{\text{NBHD}} = T_{\text{PAIR}}$ so that both algorithms are allowed the same amount of time to converge. In order to achieve this equality, we need to set $T_1 = \frac{T_2 N}{M^2}$ so that nearest neighbor gossip is permitted fewer channel uses per gossip round for M large enough. We will let T_2 be a constant.

Now we can write the energy ratio as:

$$\frac{E_{\text{NBHD}}}{E_{\text{PAIR}}} = \frac{c_4 M^2}{c_3 N} M^{\alpha/2} \frac{2^{2c_5 \log n/T_2} - \frac{1}{M}}{2^{2B_1 M^2/(T_2 N)} - 1} \tag{6.90}$$

Now we bring all of the terms up into the exponent (base 2) to get:

$$\frac{E_{\text{NBHD}}}{E_{\text{PAIR}}} < c_7 \exp\left[\left(\frac{2c_5}{T_2} - 1\right)\log N + \left(\frac{\alpha}{2} + 2\right)\log M - \frac{2B_1}{T_2}\frac{M^2}{N}\right] \tag{6.91}$$

From this equation, we can see that there is a phase transition for the neighborhood size. If the following condition is satisfied then neighborhood gossip uses exponentially less energy in N:

$$\left(\frac{2c_5}{T_2} - 1\right)\log N + \left(\frac{\alpha}{2} + 2\right)\log M < \frac{2B_1}{T_2}\frac{M^2}{N} \tag{6.92}$$

This condition can be satisfied if the number of nodes is increasing and the neighborhood size M is large enough. For instance, we can let M scale like $N^{1/2+\epsilon}$ for any $\epsilon > 0$.

6.3.5.3 Accelerated Gossip

Finally, we show that neighborhood gossip can simultaneously accelerate convergence and save energy. Assume that we would like a speedup factor of N^τ over pairwise gossip so that

$T_{\text{NBHD}} = N^{-\tau} T_{\text{PAIR}}$. The energy ratio is upper bounded as follows:

$$\frac{E_{\text{NBHD}}}{E_{\text{PAIR}}} < c_7 \exp\left[\left(\frac{2c_5}{T_2} - 1\right)\log N + \left(\frac{\alpha}{2} + 2\right)\log M - \frac{2B_1}{T_2}\frac{M^2}{N^{1+\tau}}\right] \tag{6.93}$$

If M scales like $N^{(1+\tau)/2+\epsilon}$ for any $\epsilon > 0$, then neighborhood gossip uses exponentially less energy in N. Thus, it is possible save both energy and time through the use of computation codes.

Chapter 7

Interference Alignment via Computation

Consider M transmitter-receiver pairs that communicate over a wireless channel on the same frequency band. If the users are not allowed to cooperate, it is clear that concurrent transmissions will interfere with one another. The key question is at what rate can each pair communicate in the presence of interference from all other pairs. If only one pair is active, this reduces to an interference-free point-to-point communication problem for which the capacity is known. Intuitively, it seems that the best possible scheme for M active pairs would allow each transmitter to operate at roughly $\frac{1}{M}$ its interference-free rate. Surprisingly, through a new strategy known as *interference alignment*, it is possible to have each transmitter operate all the way up to $\frac{1}{2}$ its interference-free rate.

Interference alignment was originally proposed by Maddah-Ali, Motahari and Khandani for the 2-user MIMO X channel [91] and subsequently applied to the M-user interference channel by Cadambe and Jafar [21]. The basic idea is that, from the viewpoint of each receiver, the interference should look as if it originated from a single user. For the interference channel, this can be accomplished by a vector space strategy over many parallel channels (which can be obtained by using multiple frequency bands or time instances). The end result is that each receiver sees its desired signal in half the dimensions while the interfering signals occupy the other half. As shown by Cadambe and Jafar, this allows each user to achieve $\frac{1}{2}$ its interference-free rate as the SNR goes to infinity. In this chapter, we develop a new alignment scheme for time-varying interference channels that permits each user to achieve at least half its interference-free rate at any SNR.

We now provide a high level description of our scheme. Assume that the M transmitters send out signals $X_1, X_2, \ldots, X_M$ at time t under channel matrix $\mathbf{H} = \{h_{m\ell}\}$ and that the M

receivers observe:

$$Y[t] = \sum_{\ell=1}^{M} h_{m\ell} X_\ell + Z_m[t] \tag{7.1}$$

where $Z_m[t]$ is i.i.d. Rayleigh additive noise. The transmitters wait until the complementary channel matrix $\mathbf{H}_C$ occurs at time t_C where

$$\mathbf{H}_C = \begin{bmatrix} h_{11} & -h_{12} & \cdots & -h_{1M} \\ -h_{21} & h_{22} & \cdots & -h_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ -h_{M1} & -h_{M2} & \cdots & h_{MM} \end{bmatrix} \tag{7.2}$$

and then resend $X_1, X_2, \dots, X_M$. This gives each receiver access to

$$Y_m[t_C] = h_{mm} X_m + \sum_{\ell \neq m} h_{m\ell} X_\ell + Z_m[t_C] \tag{7.3}$$

which it can add to $Y_m[t]$ to get

$$Y_m[t] + Y_m[t_C] = 2 h_{mm} X_m + Z_m[t] + Z_m[t_C]. \tag{7.4}$$

So, for the cost of two channel uses, we can get an interference-free channel. The observant reader will have noticed that, for most reasonable fading distributions, any single $\mathbf{H}_C \in \mathbb{C}^{M \times M}$ has measure zero and will effectively never occur. Fortunately, for our purposes, it is enough to wait until the channel matrix is fairly close to $\mathbf{H}_C$ to retransmit the signals. The description above is meant only to illustrate the key principles at work and we will make our analysis rigorous in the sequel.

Our primary focus is the M-user Gaussian interference channel with time-varying fading. We show that the above strategy allows each user to communicate at slightly more than half its interference-free rate regardless of M. We work with the standard information theoretic assumptions and only require that each user has causal knowledge of the channel matrix. Later, we will extend this strategy so that each receiver can recover more than one message. We also consider the X channel configuration where each transmitter has an independent message for each receiver. Here, we can only cope with 2 receivers and we comment on the difficulties of the N receiver case. Finally, we show how to employ this strategy beyond the Gaussian setting and derive the capacity region of a finite field interference channel with time-varying fading.[1]

[1] The material in this chapter is drawn from [109; 110].

Related Work

To date, the capacity region of the Gaussian interference channel is unknown except in some special cases. If the interference strength at each receiver is very strong, then it has been shown that it is optimal to first decode the interference and then extract the desired message [26; 135; 58; 133]. Conversely, if the interference strength is very weak, it is optimal to treat the interference as noise [96; 141; 8]. For the two-user case, Etkin, Wang, and Tse showed that a version of the Han-Kobayashi scheme [58] is approximately optimal and achieves the capacity region to within one bit [44].

For interference channels with more than two transmitter-receiver pairs, interference alignment can be used to mitigate the effects of interference. This strategy has been used to study capacity regions in the high SNR limit (i.e. degrees-of-freedom) for the interference channel [21] and X channel [91; 65; 22]. Recent work by Motahari, Gharan, Maddah-Ali, and Khandani has shown that alignment is possible by embedding vectors into scalar irrationals, thus circumventing the need for multiple frequency bands or time slots in the high SNR limit [95]. For static channels at finite SNR, lattice codes can be used for alignment. This was first shown by Bresler, Parekh, and Tse for the many-to-one interference channel [19]. Subsequent work has extended this strategy to fully connected, symmetric channels [149].

Our strategy makes use of the fact that parallel interference channels are *inseparable* [23; 133]. Usually, for parallel channels, we use a single code and simply optimize our resource allocation over the channels with water-filling. For the interference channel, this approach is insufficient as one can gain by coding over parallel channels.

Many of the interference alignment schemes make use of many independent channel realizations. For instance, for M users, the Cadambe-Jafar scheme requires 2^{M^2} time slots and our own scheme requires considerably more. For the three-user interference channel, two recent papers have characterized the degrees-of-freedom (when restricted to linear strategies) for limited channel realizations. First, Cadambe, Jafar, and Wang solved the case with a single complex-valued channel matrix [24]. Next, Bresler and Tse found the degrees-of-freedom for an arbitrary number of channel realizations [20].

In concurrent work to our own, Jeon and Chung have developed a similar alignment strategy for finite-field interference networks [67]. For a single-hop interference network, they match up pairs of channel matrices as we do to get interference-free channels. For a multi-hop network, they use subsequent hops to invert the channel matrix from the first hop. Another concurrent paper by Özgür and Tse examined the interference alignment scheme of Cadambe and Jafar [21] and found a lower bound on the rate at finite SNR for phase fading [119].

Finally, a recent paper by Wu and Dimakis has shown that interference alignment is also useful for designing codes for distributed storage with failures [156].

7.1 Time-Varying Gaussian Interference Channel

There are M transmitter-receiver pairs that communicate across a narrowband wireless channel over T time steps (see Figure 7.1).

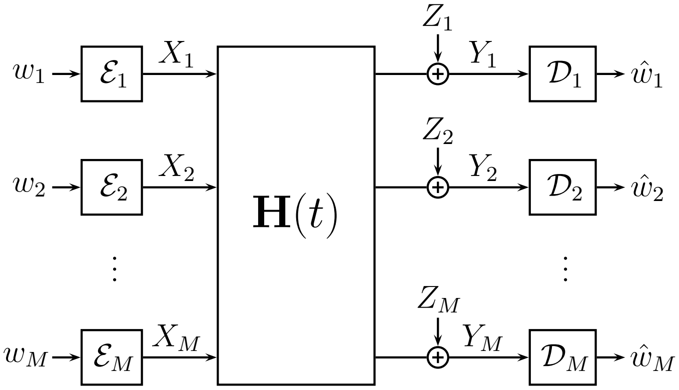

Figure 7.1: M-user interference channel.

Definition 62 (Messages)**.** Each transmitter has a *message* w_m chosen independently and uniformly from the set $\{1, 2, \ldots, 2^{n\tilde{R}_m}\}$ for some $\tilde{R}_m \geq 0$.

Definition 63 (Encoders)**.** Each transmitter has an *encoding function*, $\mathcal{E}_m$, that maps its message w_m into a length T channel input $X_m^T \in \mathbb{C}^T$ that satisfies the *power constraint*:

$$\frac{1}{T}\sum_{t=1}^{T} \big|X_m[t]\big|^2 \leq P_m. \tag{7.5}$$

Definition 64 (Channel Model)**.** The channel output observed by each receiver is a noisy linear combination of the inputs:

$$Y_m[t] = \sum_{\ell=1}^{K} h_{m\ell}[t]X_\ell[t] + Z_m[t] \tag{7.6}$$

where the $h_{m\ell}[t]$ are time-varying channel coefficients and $Z_m[t]$ is additive i.i.d. noise and drawn from a circularly symmetric complex Gaussian distribution with variance σ_n^2, $Z_m[t] \sim \mathcal{CN}(0, \sigma_m^2)$. We assume that at each time step each channel coefficient is drawn i.i.d. from a distribution with uniform phase[2]:

$$p_{h_{m\ell}}(a) = p_{h_{m\ell}}(ae^{jb}) \quad \forall a \in \mathbb{C}, b \in [0, 2\pi). \tag{7.7}$$

We also require that channel coefficients are independent of one another (although they may be drawn from different distributions). The transmitters and receivers are given access to the channel realizations causally. That is, before time t, each transmitter and receiver is given $h_{m\ell}[t]$ for all m and ℓ.

Definition 65 (Decoders)**.** Each receiver has a *decoding function* that maps its length T channel observations Y_m^T into an estimate $\hat{w}_m$ of its desired message w_m.

Definition 66 (Achievable Rate)**.** We say that a rate tuple $(R_1, R_2, \ldots, R_M)$ is *achievable* if for all $\epsilon > 0$ and n large enough there exist channel encoding and decoding functions $\mathcal{E}_1, \ldots, \mathcal{E}_M, \mathcal{D}_1, \ldots, \mathcal{D}_M$ such that:

$$\tilde{R}_m > R_m - \epsilon, \quad m = 1, 2, \ldots, M, \tag{7.8}$$

$$\Pr\left(\{\hat{w}_1 \neq w_1\} \cup \ldots \cup \{\hat{w}_M \neq w_M\}\right) < \epsilon. \tag{7.9}$$

Definition 67 (Capacity)**.** The *capacity region* is the closure of the set of all achievable rate tuples.

7.2 Channel Quantization

Our scheme relies on matching up time indices based on the phase and magnitude of the channel coefficients. In order to ensure that most channel coefficients are matched, we need strong typicality and for this we need the channel coefficients to take values on a finite set. We will accomplish this by quantizing the channel coefficients with a resolution determined by our desired gap to the target rate. By taking finer and finer quantizations, we can achieve the target rate in the limit.

First, we will threshold the channel coefficients by throwing out any time indices that contain a channel coefficient magnitude larger than h_{MAX}. This threshold is chosen such that the probability that one or more channel coefficients violate it in one time instant is τ.

Each channel coefficient is quantized as follows. The complex plane (up to radius h_{MAX}) is divided up into κ disjoint rings of equal width. These rings are further subdivided into equal segments based on η angles spaced equally between 0 and 2π where η. Each segment is a quantization cell for the channel coefficients. The parameters κ and η are chosen such that the maximum distance between any two points in any segment is ν where $\nu > 0$ will

[2] In the literature, a fading process that varies rapidly over the duration of the codeword is often called *fast fading*.

be specified later. We also assign all channel coefficients with magnitude larger than h_{MAX} to an erasure symbol.

The following lemma will allow us to show that for ν small enough, matching up channel coefficients based on their quantization cells has a negligible effect on the overall rate.

Lemma 26. *Given $h_k \in \mathbb{C}$ satisfying $|h_k| < h_{MAX}$ for $k = 1, 2, \ldots, K$, let $\hat{h}_k$ be any other element of the quantization cell of h_k. For any $a_k \in \mathbb{C}$, the following upper bound holds:*

$$\left| \sum_{k=1}^{K} a_k \hat{h}_k \right| \leq \left| \sum_{k=1}^{K} a_k h_k \right| + \nu \sum_{k=1}^{K} |a_k|. \tag{7.10}$$

Furthermore, if a_k is chosen such that $\left| \sum_{k=1}^{K} a_k h_k \right| > \nu \sum_{k=1}^{K} |a_k|$, then the following lower bound holds:

$$\left| \sum_{k=1}^{K} a_k \hat{h}_k \right| \geq \left| \sum_{k=1}^{K} a_k h_k \right| - \nu \sum_{k=1}^{K} |a_k|. \tag{7.11}$$

Proof. First, write each $\hat{h}_k = h_k + e_k$ where $|e_k| < \nu$. Now, we have by the triangle inequality:

$$\begin{aligned} \left| \sum_{k=1}^{K} a_k \hat{h}_k \right| = \left| \sum_{k=1}^{K} a_k (h_k + e_k) \right| &\leq \left| \sum_{k=1}^{K} a_k h_k \right| + \left| \sum_{k=1}^{K} a_k e_k \right| \\ &\leq \left| \sum_{k=1}^{K} a_k h_k \right| + \nu \sum_{k=1}^{K} |a_k|. \end{aligned}$$

Similarly, by the reverse triangle inequality, we have that:

$$\left| \sum_{k=1}^{K} a_k (h_k + e_k) \right| \geq \left| \sum_{k=1}^{K} a_k h_k \right| - \left| \sum_{k=1}^{K} a_k e_k \right| \tag{7.12}$$

$$\geq \left| \sum_{k=1}^{K} a_k h_k \right| - \nu \sum_{k=1}^{K} |a_k|. \tag{7.13}$$

□

For the remainder of this chapter, we will treat all channel coefficients as if they are quantized. Thus, we can treat them as if drawn from a discrete set where the probability of each quantization cell is given by the total probability of all channel coefficients in that cell. By construction, all quantization cells at a given radius have the same probability. Note that this depends strongly on the assumption of uniform phase.

We now recall the notion of strong typicality for sequences of discrete random variables. Let $\mathbf{H} = \{h_{nm}\}$ be the matrix of (quantized) channel coefficients which takes values in the set $\mathcal{H}$, $P(\mathbf{H})$ the probability of drawing $\mathbf{H}$ under the channel model, and $\mathbf{H}^{[T]}$ denote the sequence of such matrices over T channel uses. Let $\#(\mathbf{H}|\mathbf{H}^{[T]})$ denote the number of times the channel matrix $\mathbf{H}$ occurs in the sequence $\mathbf{H}^{[T]}$.

Definition 68. A sequence of channel matrices, $\mathbf{H}^{[T]}$, is γ-typical if:

$$\left| \frac{1}{T} \#(\mathbf{H}|\mathbf{H}^{[T]}) - P(\mathbf{H}) \right| \leq \gamma \quad \forall \mathbf{H} \in \mathcal{H}. \tag{7.14}$$

Let A_γ^T denote the set of all γ-typical channel matrix sequences.

Lemma 27 (Csiszar-Körner 2.12)**.** *For any i.i.d. sequence of channel matrices,* $\mathbf{H}^{[T]}$*, the probability of the set of all* γ*-typical sequences,* A_γ^T*, is lower bounded by:*

$$P(A_\gamma^T) \geq 1 - \frac{|\mathcal{H}|}{4T\gamma^2} \tag{7.15}$$

For a proof, see [63]. Due to the channel quantization, the size of $\mathcal{H}$ is $|\mathcal{H}| = (\kappa\eta + 1)^{MN}$. We will only work with sequences of channel matrices that are γ-typical and declare errors on the rest. This ensures that nearly all time indices can be matched up appropriately.

7.3 Ergodic Interference Alignment

Since the receivers cannot cooperate, the highest rates are achieved when there is no interference between users. Specifically, when $h_{m\ell} = 0 \;\; \forall \ell \neq m$, each receiver sees a point-to-point channel from its transmitter and can achieve

$$R_m = E\left[\log\left(1 + |h_{mm}|^2 \frac{P_m}{\sigma_m^2} \right) \right]. \tag{7.16}$$

We call this the *interference-free rate* and we will use it as a benchmark to gauge our performance.

Remark 31. Note that this assumes a uniform power allocation across all time slots and one can do better by using the causal channel state information to optimize the power allocation [25]. For simplicity, we use a uniform power allocation throughout our derivations. The optimization techniques employed in the point-to-point case can be applied identically to our results as well. See [152] for a study of power allocation for fast fading 2-user interference channels.

A simple approach to interference management is to have transmitters take turns using the channel. For instance, if we partition the channel equally between transmitters, each one can achieve

$$R_m = \frac{1}{M} E\left[\log\left(1 + M|h_{mm}|^2 \frac{P_m}{\sigma_m^2}\right)\right]. \tag{7.17}$$

The extra M factor inside the log comes from saving up power while the transmitter is required to stay silent. Under this approach, the sum rate stays nearly constant as we add users to the network. With interference alignment, we can do significantly better. Cadambe and Jafar's alignment scheme allows for the sum rate to increase linearly with the number of users:

$$\lim_{P_m \to \infty} \frac{\sum_{m=1}^{M} R_m}{\log(1 + P_m)} = \frac{M}{2}. \tag{7.18}$$

This means that each user is guaranteed a fixed rate (at high SNR) regardless of the number of users in the network. We now develop a new technique, *ergodic interference alignment*, that allows each user to achieve at least half its interference-free rate at any SNR.

Theorem 36. *For the M-user Gaussian interference channel with fast fading and uniform phases, each transmitter can achieve the following rate:*

$$R_m = \frac{1}{2} E\left[\log\left(1 + 2|h_{mm}|^2 \frac{P_m}{\sigma_m^2}\right)\right]. \tag{7.19}$$

Proof. Choose $\epsilon > 0$. For ease of analysis, we divide up our T channel uses into two intervals at the halfway point $T/2$. Using Lemma 27, we have that for T large enough, both intervals will be γ-typical with probability at least $(1 - \frac{\epsilon}{2})$ (with γ to be specified later). By Definition 68, this means that the number of occurrences of each possible channel matrix in each interval is bounded as follows:

$$\frac{T}{2}\left(P(\mathbf{H}) - \gamma\right) \leq \#(\mathbf{H}|\mathbf{H}^{[T/K]}) \leq \frac{T}{2}\left(P(\mathbf{H}) + \gamma\right) \tag{7.20}$$

for all $\mathbf{H} \in \mathcal{H}$.

Each encoder uses a length λT codebook $\mathcal{C}_m$ for some $\lambda > 0$ with rate $\tilde{R}_m$. The codebook is generated elementwise i.i.d. from a circularly symmetric Gaussian distribution with variance $P_m - \epsilon$.

Assume that the intervals are γ-typical so that each matrix will occur at least $\frac{T}{2}\left(P(\mathbf{H}) - \gamma\right)$ times in each interval. A time slot t in an interval is useable unless:

1. The channel matrix $\mathbf{H}[t]$ contains one or more elements with magnitude larger than h_{MAX}.

2. The channel matrix $\mathbf{H}[t]$ does not violate the threshold but has already occurred at least $\frac{T}{2}\left(P(\mathbf{H}) - \gamma\right)$ times.

We give a lower bound on the number of useable time slots below which we set to be equal to the length of the codebook:

$$\lambda T = \frac{T}{2} \sum_{\mathbf{H}:|h_{m\ell}|<h_{\mathrm{MAX}}} \left(P(\mathbf{H}) - \gamma\right) \tag{7.21}$$

$$= \frac{T}{2}\left(1 - \tau - (\kappa\eta)^{M^2}\gamma\right). \tag{7.22}$$

Recall that τ is the probability the channel matrix contains an element larger than h_{MAX} and κ and ν are parameters in the channel quantization.

During the first interval, each transmitter sends out a new symbol from its length λT codeword during each useable time slot t_1 and records the channel matrix $\mathbf{H}[t_1]$. We match up each useable time slot t_1 from the first interval with a useable time slot t_2 from the second interval for which the channel matrix $\mathbf{H}[t_2]$ is *complementary*:

$$\mathbf{H}[t_2] = \begin{bmatrix} h_{11}[t_1] & -h_{12}[t_1] & \cdots & -h_{1M}[t_1] \\ -h_{21}[t_1] & h_{22}[t_1] & \cdots & -h_{2M}[t_1] \\ \vdots & \vdots & \ddots & \vdots \\ -h_{M1}[t_1] & -h_{M2}[t_1] & \cdots & h_{MM}[t_1] \end{bmatrix}. \tag{7.23}$$

Note that this can be done using only causal channel knowledge by greedily matching time slots from the first interval in the order in which they occur. At the end of T channel uses, each receiver has access to equations of the form

$$Y_m[t_1] = h_{mm}X_m[t_1] + \sum_{\ell \neq m} h_{m\ell}[t_1]X_m[t_1] + Z_m[t_1] \tag{7.24}$$

$$Y_m[t_2] = h_{mm}X_m[t_1] - \sum_{\ell \neq m} h_{m\ell}[t_1]X_m[t_1] + Z_m[t_2]. \tag{7.25}$$

Each receiver adds these two equations together to get a nearly interference-free channel. Since the channel matrices are paired up according to their quantization cells, some residual interference will remain which we will treat as noise. Using Lemma 28, it can be shown that the signal-to-interference-and-noise is bounded below as follows:

$$\mathsf{SINR}_m \geq \frac{P_m\left(2|h_{mm}| - 2\nu\right)^2}{4\nu^2 \sum_{\ell \neq m} P_\ell + 2\sigma_m^2}. \tag{7.26}$$

Note that by letting $\nu \to 0$, we get that $\mathsf{SINR}_m \geq \frac{2|h_{mm}|^2}{\sigma_m^2}$.By choosing ν, γ, and τ small enough and T large enough, we can guarantee that SINR_m and λ are such that we can find a good code with probability of error at most $\frac{\epsilon}{2}$ and rate at least

$$\frac{1}{2}E\left[\log\left(1+2|h_{mm}|^2\frac{P_m}{\sigma_m^2}\right)\right]-\epsilon. \tag{7.27}$$

Recall also that with probability $\frac{\epsilon}{2}$ the channel is not γ-typical. Since the total probability of error is less than ϵ, we get the desired result. □

Note that this achievable strategy does not, in general, yield the capacity region. For instance, if the cross-channel gains are very small, then it is better to treat the interference as noise, rather than spending two channel uses to cancel it out. Thus, for Rayleigh fading, we can achieve higher rates by using this weak interference strategy over certain channel matrices and the alignment strategy over the rest. We can also expand the achievable rate region by time-sharing. One user is given a fraction of the channel uses for its exclusive use while the rest are used for interference alignment. See the proof of Theorem 40 for an application to the finite field setting.

For some special cases, it turns out that our strategy is optimal. In [64], Jafar developed the concept of a "bottleneck state" and showed that ergodic alignment is capacity-achieving. Roughly speaking, in a bottleneck state, receivers see interference at equal strength to their desired signal.

Overall, our scheme shows that it is possible to benefit from interference alignment at finite SNR. Moreover, the analysis is considerably simpler than for the original scheme proposed in [21] (with the assumption of uniform phase). It remains an open question whether this performance can be attained using less channel diversity or more limited channel state information.

7.4 Recovering More Messages

For the standard interference channel, we assume that each receiver is only interested in one of the transmitted messages. Here, we generalize our alignment scheme to handle the case where each receiver attempts to decode more than one message. The problem setup is largely the same as in Section 7.1 except that now there are M transmitters, each with a single message w_m with rate R_m, and N receivers that want exactly L messages each. For simplicity, we will assume that all messages are requested by the same number of receivers. (Note that this implicitly assumes that $\frac{NL}{M}$ is an integer.) Denote the subset of receivers that want message m by $\mathcal{S}_m$. In Figure 7.2, we provide a block diagram of a case with $M = 4$ transmitters, $N = 4$ receivers, and message requests $\mathcal{S}_1 = \{1,2\}, \mathcal{S}_2 = \{2,3\}, \mathcal{S}_3 = \{3,4\}$, and $\mathcal{S}_4 = \{4,1\}$.

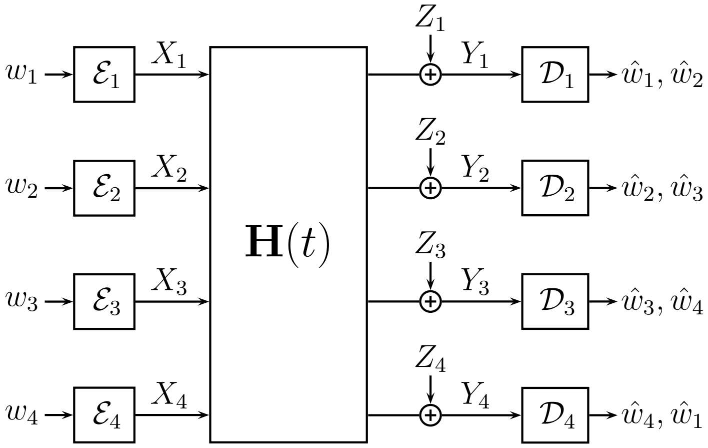

Figure 7.2: Interference channel where each receiver wants $L = 2$ messages.

7.4.1 Equation Coefficients

As before, we will create equations of the transmitted codewords at the receivers which can be solved for the desired messages. Essentially, at well-chosen time indices, all encoders retransmit symbols that were sent at an earlier time. This has the effect of giving the decoders equations with the symbols as the variables and the coefficients given by the channel. Here, it is insufficient to look for pairs of matrices that exactly cancel. In general, we will match up K matrices that allow all receivers to solve for their desired messages.

First, we assume that all channel coefficients are quantized as described in Section 7.2. In order to ensure that all channel coefficients can be appropriately matched, we only consider matchings between individual coefficients of the same magnitude. Since the phase of each coefficient is assumed to be uniform, all equations will have the same probability.

The goal is to specify a set of K equations such that the receiver can recover its desired messages. Each receiver is free to choose its own equations and repeat transmissions only occur when all receivers see the appropriate equations. These equations are fully specified by phase shifts $\phi_{nm}^{(k)}$ (with $\phi_{nm}^{(1)} = 1$ by default). For ease of analysis, these are restricted to take values on the set $\{e^{jb} : b = 0, \frac{2\pi}{\eta}, \frac{4\pi}{\eta}, \ldots, \frac{2\pi(\eta-1)}{\eta}\}$ so that they are in correspondence with the quantization cells. We write the phase shifts at each receiver for the k^{th} equation in matrix form below:

$$\mathbf{\Phi}^{(k)} = \begin{bmatrix} \phi_{11}^{(k)} & \phi_{12}^{(k)} & \cdots & \phi_{1M}^{(k)} \\ \phi_{21}^{(k)} & \phi_{22}^{(k)} & \cdots & \phi_{2M}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{N1}^{(k)} & \phi_{N2}^{(k)} & \cdots & \phi_{NM}^{(k)} \end{bmatrix} \tag{7.28}$$

We now show how to match up channel matrices based on these phase shifts. Let $\mathbf{A} \odot \mathbf{B} \triangleq \{a_{nm} b_{nm}\}$ denote the Hadamard product of $\mathbf{A}$ and $\mathbf{B}$. We divide up the T channel uses into K intervals of length T/K. Using Lemma 27, we have that for T large enough, all K intervals will be γ-typical with probability at least $(1 - \frac{\epsilon}{2})$. By Definition 68, this means that the number of occurrences of each possible channel matrix in each interval is bounded as follows:

$$\frac{T}{K}\left(P(\mathbf{H}) - \gamma\right) \leq \#(\mathbf{H}|\mathbf{H}^{[T/K]}) \leq \frac{T}{K}\left(P(\mathbf{H}) + \gamma\right) \tag{7.29}$$

for all $\mathbf{H} \in \mathcal{H}$.

Each encoder uses a length λT codebook $\mathcal{C}_m$ with rate $\tilde{R}_m$ generated i.i.d. from a circularly symmetric Gaussian distribution with variance $P_m - \epsilon$.

Assume that the intervals are γ-typical. Each matrix will occur at least $\frac{T}{K}\left(P(\mathbf{H}) - \gamma\right)$ times in each interval. During the first time interval, each encoder transmits a new symbol from its codeword at each time step t unless:

1. The channel matrix $\mathbf{H}[t]$ contains one or more elements with magnitude larger than h_{MAX}.
2. The channel matrix $\mathbf{H}[t]$ does not violate the threshold but has already occurred at least $\frac{T}{K}\left(P(\mathbf{H}) - \gamma\right)$ times.

We give a lower bound on the number of useable time slots below which we set to be equal to the length of the codebook:

$$\lambda T = \frac{T}{K} \sum_{\mathbf{H}: |h_{nm}| < h_{\text{MAX}}} \left(P(\mathbf{H}) - \gamma\right) \tag{7.30}$$

$$= \frac{T}{K}\left(1 - \tau - (\kappa\eta)^{MN}\gamma\right). \tag{7.31}$$

We then match up used time slots from the first interval with time slots in the remaining $K - 1$ intervals. During the k^{th} time interval, when the channel matrix $\mathbf{\Phi}^{(k)} \odot \mathbf{H}$ occurs, it is matched with the first unmatched time slot from the first interval that had channel matrix $\mathbf{H}$. The encoders retransmit the symbols from the first interval for all matched time slots. Since are intervals are assumed to be γ-typical, all $\frac{T}{K}\left(P(\mathbf{H})-\gamma\right)$ time indices for each matrix from the first interval can be successfully matched.

After T time steps, receiver n has access to equations of the form:

$$y_n^{(1)} = \sum_{m=1}^{M} h_{nm}x_m + z_n^{(1)} \tag{7.32}$$

$$y_n^{(2)} = \sum_{m=1}^{M} \phi_{nm}^{(2)} h_{nm}x_m + z_n^{(2)} \tag{7.33}$$

$$\vdots \tag{7.34}$$

$$y_n^{(K)} = \sum_{m=1}^{M} \phi_{nm}^{(K)} h_{nm}x_m + z_n^{(K)} \tag{7.35}$$

where x_m are the symbols from a single index in the chosen codewords, h_{nm} are fixed channel coefficients (up to the quantization cells), and $z_n^{(k)}$ are the noise terms from the matched time indices.

Given these equations, the receiver attempts to recover the symbols from its desired by applying linear transformations. For each desired symbol x_ℓ, the receiver forms an estimate:

$$u_{n\ell} = \sum_{k=1}^{K} a_{n\ell}^{(k)} y_n^{(k)} \tag{7.36}$$

$$= \sum_{m=1}^{M} h_{nm}x_m \sum_{k=1}^{K} a_{n\ell}^{(k)}\phi_{nm}^{(k)} + \sum_{k=1}^{K} a_{n\ell}^{(k)} z_n^{(k)} \tag{7.37}$$

for some choice of $a_{n\ell}^{(k)} \in \mathbb{C}$.

Let $\delta[\ell]$ be the Kronecker delta function. The following lemma establishes a worst-case signal-to-interference-and-noise ratio (SINR) for the channel between x_ℓ and u_ℓ.

Lemma 28. *Assume that $\phi_{nm}^{(k)}$ and $a_{n\ell}^{(k)}$ are chosen such that $\sum_{k=1}^{K} a_{n\ell}^{(k)}\phi_{nm}^{(k)} = \beta\delta[\ell - m]$ for some $\beta > 0$. Then, the AWGN channel between symbol x_ℓ and estimate $u_{n\ell}$ has an SINR that is lower bounded by:*

$$\mathsf{SINR} \geq \frac{P_\ell\left(\beta|h_{n\ell}| - \nu\sum_{k=1}^{K}|a_{n\ell}^{(k)}|\right)^2}{\nu^2\left(\sum_{k=1}^{K}|a_{n\ell}^{(k)}|\right)^2\sum_{m\neq\ell}P_m + \sigma_n^2\sum_{k=1}^{K}|a_{n\ell}^{(k)}|^2}.$$

Furthermore, as ν goes to zero, we have that:

$$\lim_{\nu \downarrow 0} \mathsf{SINR} \geq \frac{P_\ell \beta^2 |h_{n\ell}|^2}{\sigma_n^2 \sum_{k=1}^K |a_{n\ell}^{(k)}|^2}. \tag{7.38}$$

Proof. First, we lower bound the signal power which is slightly diminished due to channel quantization. By Lemma 26, the signal power is lower bounded as follows:

$$P_\ell \left| \sum_{k=1}^K a_{n\ell}^{(k)} h_{n\ell} \phi_{n\ell}^{(k)} \right|^2 \geq P_\ell \left(\beta |h_{n\ell}| - \nu \sum_{k=1}^K |a_{n\ell}^{(k)}| \right)^2 . \tag{7.39}$$

Now, we upper bound the power of the remaining interference due to quantization. Again, by Lemma 26, the power of each interferer $m \neq \ell$ at receiver n is upper bounded as follows:

$$P_m \left| \sum_{k=1}^K a_{nm}^{(k)} h_{nm} \phi_{nm}^{(k)} \right|^2 \leq P_m \left(0 + \nu \sum_{k=1}^K |a_{n\ell}^{(k)}| \right)^2 . \tag{7.40}$$

Finally, the noise terms $z_n^{(k)}$ are each weighted by $a_{n\ell}^{(k)}$ in $u_{n\ell}$. Since the noise is i.i.d. across time, we get that $\sigma_n^2 \sum_{k=1}^K |a_{n\ell}^{(k)}|^2$ as the power of the sum of the noise.

□

The requirements on $a_{n\ell}^{(k)}$ and $\phi_{nm}^{(k)}$ in Lemma 28 can be restated as a matrix condition. Let $\mathbf{A}_n$ and $\mathbf{\Phi}_n$ be defined as

$$\mathbf{A}_n = \begin{bmatrix} a_{n1}^{(1)} & a_{n1}^{(2)} & \cdots & a_{n1}^{(K)} \\ a_{n2}^{(1)} & a_{n2}^{(2)} & \cdots & a_{n2}^{(K)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{nM}^{(1)} & a_{nM}^{(2)} & \cdots & a_{nM}^{(K)} \end{bmatrix} \tag{7.41}$$

$$\mathbf{\Phi}_n = \begin{bmatrix} \phi_{n1}^{(1)} & \phi_{n2}^{(1)} & \cdots & \phi_{nM}^{(1)} \\ \phi_{n1}^{(2)} & \phi_{n2}^{(2)} & \cdots & \phi_{nM}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n1}^{(K)} & \phi_{n2}^{(K)} & \cdots & \phi_{nM}^{(K)} \end{bmatrix} . \tag{7.42}$$

Assume that receiver n wants messages with indices $\ell_1, \ell_2, \ldots, \ell_I$. Then the following condition is equivalent to $\sum_{k=1}^{K} a_{n\ell}^{(k)} \phi_{nm}^{(k)} = \beta\delta[\ell_i - m]$ for $i = 1, 2, \dots, I$:

$$\mathbf{A}_n \mathbf{\Phi}_n = \beta \begin{bmatrix} \delta[\ell_1 - 1] & \delta[\ell_1 - 2] & \cdots & \delta[\ell_1 - M] \\ \delta[\ell_2 - 1] & \delta[\ell_2 - 2] & \cdots & \delta[\ell_2 - M] \\ \vdots & \vdots & \ddots & \vdots \\ \delta[\ell_I - 1] & \delta[\ell_I - 2] & \cdots & \delta[\ell_I - M] \end{bmatrix}$$

Example 15. For the standard interference channel, we have that $N = M$ and $\mathcal{S}_m = \{m\}$. The framework above can be used to derive Theorem 36. Each receiver uses $K = 2$ equations to recover its desired messages w_m. The phase shifts are given by $\phi_{mm}^{(2)} = 1$ and $\phi_{nm}^{(2)} = -1$ for $n \neq m$. The messages are recovered using $a_{mm}^{(1)} = a_{mm}^{(2)} = 1$. It follows that $\sum_{k=1}^{2} a_{mm}^{(k)} \phi_{nm}^{(k)} = 2\delta[n - m]$. Applying Lemma 28, we get the desired result.

Let $\omega_K \triangleq e^{j2\pi/K}$ denote the K^{th} root of unity and let $\mathbf{W}_K$ be the size K discrete Fourier transform (DFT) matrix:

$$\mathbf{W}_K = \begin{bmatrix} \omega_K^0 & \omega_K^0 & \omega_K^0 & \cdots & \omega_K^0 \\ \omega_K^0 & \omega_K^1 & \omega_K^2 & \cdots & \omega_K^{K-1} \\ \omega_K^0 & \omega_K^2 & \omega_K^4 & \cdots & \omega_K^{2(K-1)} \\ \vdots & \vdots & \ddots & \vdots & \\ \omega_K^0 & \omega_K^{K-1} & \omega_K^{2(K-1)} & \cdots & \omega_K^{(K-1)^2} \end{bmatrix}. \tag{7.43}$$

Recall that the inverse DFT matrix has the following form:

$$\mathbf{W}_K^{-1} = \frac{1}{K} \begin{bmatrix} \omega_K^0 & \omega_K^0 & \cdots & \omega_K^0 \\ \omega_K^0 & \omega_K^{-1} & \cdots & \omega_K^{-(K-1)} \\ \omega_K^0 & \omega_K^{-2} & \cdots & \omega_K^{-2(K-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_K^0 & \omega_K^{-(K-1)} & \cdots & \omega_K^{-(K-1)^2} \end{bmatrix}. \tag{7.44}$$

Theorem 37. *For the fast fading Gaussian interference channel with uniform phase and N receivers that want L messages, each transmitter can achieve the following rate:*

$$R_m = \min_{n \in \mathcal{S}_m} \frac{1}{L+1} E\left[\log\left(1 + (L+1)|h_{nm}|^2 \frac{P_m}{\sigma_n}\right)\right]. \tag{7.45}$$

Proof. Without loss of generality, assume that receiver n is interested in messages from transmitters $1, 2, \dots, L$. (Otherwise, just reindex the transmitters.) To recover these messages, the receiver needs $L + 1$ equations: L for the messages and one for the interference. The phase coefficients are chosen from the DFT matrix of size $\mathbf{W}_{L+1}$:

$$\phi_{n\ell}^{(k)} = \begin{cases} \exp\left(\frac{j2\pi(\ell-1)(k-1)}{L+1}\right) & \ell = 1, 2, \ldots, L \\ \exp\left(\frac{j2\pi L(k-1)}{L+1}\right) & \ell = L+1, L+2, \ldots, M \end{cases}$$

and the recovery coefficients are chosen from the inverse DFT matrix $\mathbf{W}_{L+1}$ scaled by $L+1$:

$$a_{n\ell}^{(k)} = \exp\left(\frac{-j2\pi(\ell-1)(k-1)}{L+1}\right) \text{ for } \ell = 1, 2, \ldots, L$$

This immediately gives that $\mathbf{A}_n \mathbf{\Phi}_n = (L+1)\mathbf{I}$. We can now apply Lemma 28 to show that the resulting channel from each transmitter has $\mathsf{SINR}_{n\ell}$ no worse than:

$$\mathsf{SINR}_{n\ell} \geq \frac{P_\ell \left((L+1)|h_{n\ell}| - (L+1)\nu\right)^2}{(L+1)^2\nu^2 \sum_{\ell \neq m} P_\ell + (L+1)\sigma_n^2}.$$

Since the message from transmitter ℓ is multicast to several receivers, the rate is governed by the worst channel:

$$\mathsf{SINR}_\ell = \min_{n \in \mathcal{S}_\ell} \mathsf{SINR}_{n\ell} \tag{7.46}$$

By choosing ν, γ and τ small enough and T large enough, we can guarantee that SINR_ℓ is such that we can find a good code to all receivers with probability of error at most $\frac{\epsilon}{2}$ and rate at least

$$\min_{n \in \mathcal{S}_\ell} \frac{1}{L+1} E\left[\log\left(1 + (L+1)|h_{n\ell}|^2 \frac{P_\ell}{\sigma_n^2}\right)\right] - \epsilon. \tag{7.47}$$

Recall also that with probability $\frac{\epsilon}{2}$ the channel is not γ-typical. Since the total probability of error is less than ϵ, we get the desired result. □

One way to think about the $L+1$ equations used in this scheme is that we use one equation for each desired message and one for all of the remaining interference. The choice of the DFT matrix for our coefficients allows us to extract the maximum possible coherence gain out of these $L+1$ time slots. Note that if we simply extended the scheme from Theorem 36, to cancel out the interference from each desired message one-by-one, we could not achieve the same rates. Specifically, if we have

$$\mathbf{\Phi}_1 = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & \cdots & 1 \\ 1 & -1 & \cdots & -1 & -1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \cdots & 1 & -1 & \cdots & -1 \end{bmatrix} \tag{7.48}$$

then we only achieve a rate of

$$R_\ell = \min_{n \in \mathcal{S}_\ell} \frac{1}{L+1} E\left[\log\left(1 + 2|h_{n\ell}|^2 \frac{P_\ell}{\sigma_n^2}\right)\right]. \tag{7.49}$$

It can be shown that in the high SNR limit, this scheme achieves the sum degrees-of-freedom $\frac{M}{L+1}$ using similar upper bound techniques as found in [21].

7.5 X Message Set

We now turn to a variant of the interference channel, the X channel, that has garnered significant attention [65; 91; 22]. In this scenario, there are M transmitters and N receivers and each transmitter has an independent message for each receiver. For the single antenna case, Cadambe and Jafar showed that the sum degrees-of-freedom is $\frac{MN}{M+N-1}$ using interference alignment [22]. Here, we extend this result to the finite SNR regime for the special case of $N = 2$ receivers. Let $w_{m\ell}$ denote the message sent from the m^{th} transmitter to the ℓ^{th} receiver where ℓ takes values from 1 to N. Each message has rate $R_{m\ell}$. In Figure 7.3, we give a block diagram of an X message set for $M = 2$ transmitters and $N = 2$ receivers.

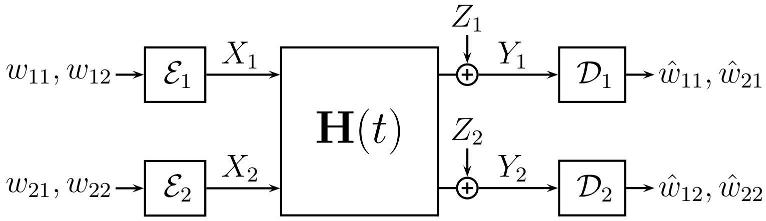

Figure 7.3: X message set for $M = 2$ transmitters and $N = 2$ receivers.

Unlike in our previous schemes, we cannot hope for the channel to generate an independent coefficient for every message. Transmitters must artificially separate their messages by premultiplying them by phases. This leaves us with fewer variables to work with to align the interference at every receiver.

For simplicity, we assume each transmitter splits its power equally between its messages w_{m1} and w_{m2}. The phase rotations at the transmitter for the k^{th} equation are given by $\theta_{m1}^{(k)}$ and $\theta_{m2}^{(k)}$. This results in the following channel input:

$$X_m^{(k)} = \theta_{m1}^{(k)} X_{m1} + \theta_{m2}^{(k)} X_{m2} \tag{7.50}$$

It is also convenient to represent these phases in matrix form:

$$\mathbf{\Theta}_n = \begin{bmatrix} \theta_{1n}^{(1)} & \theta_{2n}^{(1)} & \cdots & \theta_{Mn}^{(1)} \\ \theta_{1n}^{(2)} & \theta_{2n}^{(2)} & \cdots & \theta_{Mn}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{1n}^{(K)} & \theta_{2n}^{(K)} & \cdots & \theta_{Mn}^{(K)} \end{bmatrix}. \tag{7.51}$$

We are also free to choose the phases provided by the channel $\phi_{nm}^{(k)}$. The receivers see equations of all transmitted messages of the following form:

$$y_1^{(k)} = \sum_{m=1}^{M} \phi_{1m}^{(k)} h_{nm} (\theta_{m1}^{(k)} x_{m1} + \theta_{m2}^{(k)} x_{m2}) + z_1^{(k)} \tag{7.52}$$

$$y_2^{(k)} = \sum_{m=1}^{M} \phi_{2m}^{(k)} h_{nm} (\theta_{m1}^{(k)} x_{m1} + \theta_{m2}^{(k)} x_{m2}) + z_2^{(k)} \tag{7.53}$$

We can represent all of the phases seen at each receiver in a single matrix by ordering the messages as follows $w_{11}, w_{21}, \ldots, w_{M1}, w_{12}, w_{22}, \ldots, w_{M2}$. The matrix of phases is

$$\mathbf{B}_n = [\mathbf{\Theta}_1 \odot \mathbf{\Phi}_n \quad \mathbf{\Theta}_2 \odot \mathbf{\Phi}_n] \tag{7.54}$$

The key is to choose all of the phases such that the left half of $\mathbf{B}_n$ is full rank at receiver 1 and composed of identical columns at receiver 2 while at the same time ensuring the right half is full rank at receiver 2 and composed of identical columns at receiver 1. This is indeed possible as shown by the following theorem.

Theorem 38. *For the X message set with $N = 2$ receivers, the following rates are achievable:*

$$R_{nm} = \frac{1}{M+1} E \left[\log \left(1 + \frac{M+1}{2} |h_{nm}|^2 \frac{P_m}{\sigma_n^2} \right) \right] \tag{7.55}$$

Proof. We will show that it is possible to design the phases so that both receivers see a DFT matrix with independent columns for the desired messages and the same column for undesired messages. Choose the phases as follows:

$$\theta_{m1}^{(k)} = \exp\left(\frac{j2\pi(m-1)k}{M+1}\right) \tag{7.56}$$

$$\theta_{m2}^{(k)} = \exp\left(\frac{j2\pi Mk}{M+1}\right) \tag{7.57}$$

$$\phi_{1m}^{(k)} = 1 \tag{7.58}$$

$$\phi_{2m}^{(k)} = \exp\left(\frac{-j2\pi(m-1)k}{M+1}\right) \tag{7.59}$$

Let $\alpha = \exp(j2\pi/(M+1))$. We get that $\mathbf{B}_1$ is equal to:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{(M-1)} & \alpha^M & \alpha^M & \cdots & \alpha^M \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^M & \cdots & \alpha^{(M-1)M} & \alpha^{M^2} & \alpha^{M^2} & \cdots & \alpha^{M^2} \end{bmatrix}$$

and $\mathbf{B}_2$ is equal to:

$$\begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 & \alpha^M & \alpha^{M-1} & \cdots & \alpha \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 & \alpha^{M^2} & \alpha^{M(M-1)} & \cdots & \alpha^M \end{bmatrix}$$

Now, receivers 1 and 2 can treat this as a multicast problem and choose $\mathbf{A}_1$ to be the first M columns of the size $M+1$ inverse DFT matrix and $\mathbf{A}_2$ to be the last M columns. Following the remaining steps in the proof of Theorem 37 yields the desired result. □

It is not clear how this result can be extended beyond the two receiver case. Ideally, we would like to choose transmitter phase rotations and channel phases to get a matrix of phases

$$\mathbf{B}_n = [\mathbf{\Theta}_1 \odot \mathbf{\Phi}_n \quad \mathbf{\Theta}_2 \odot \mathbf{\Phi}_n \quad \cdots \quad \mathbf{\Theta}_N \odot \mathbf{\Phi}_n] \tag{7.60}$$

that allows us to appropriately align interference at all receivers. In the best case, we would only need M equations for our desired messages and $N-1$ for the interference from messages intended for other receivers.

7.6 Time-Varying Finite Field Interference Channel

For the Gaussian case, it has been sufficient to match up channel matrices and add up the resulting channel outputs. The simplicity of this strategy is in some ways an artifact of the Gaussian setting. In general, the receivers may need to perform a decoding step prior to combining the observed signals to avoid noise build-up. In this section, we consider a finite field interference channel with fast fading and derive the entire capacity region. Each receiver groups together time instances with the same channel coefficients and decodes a function of the messages, using a computation code from Chapter 3. By combining two appropriately chosen functions, the interference can be completely removed.

The problem statement is identical to that in Section 7.1 except for the channel model. We assume that all additions and multiplications are carried out over a finite field $\mathbb{F}_q$.

Definition 69 (Channel Model)**.** We assume that the channel inputs and outputs take values on the same finite field $\mathbb{F}_q$. The channel output observed by each receiver is a noisy linear combination of its inputs:

$$Y_m[t] = \sum_{\ell=1}^{M} h_{m\ell}[t] X_\ell[t] + Z_m[t] \tag{7.61}$$

where the $h_{m\ell}[t]$ are time-varying channel coefficients and $Z_m[t]$ is additive i.i.d. noise drawn from a distribution that takes values on uniformly on $\{1, 2, \ldots, q-1\}$ with probability ρ and is zero otherwise. We define the entropy of $Z_k(t)$ to be $0 \leq H(Z) \leq \log_2 q$. We assume that at each time step each channel coefficient is drawn independently and uniformly from $\mathbb{F}_q \setminus \{0\}$. The transmitters and receivers are given access to the channel realizations causally. That is, before time t, each transmitter and receiver is given $h_{m\ell}[t]$ for all m and ℓ.

Remark 32. Our results can be extended to the case where the channel coefficients are sometimes zero through simple counting arguments. However, this considerably complicates the description of the capacity region.

Lemma 29. *There exists a bijection,* $g : \mathbb{F}_q^{M \times M} \rightarrow \mathbb{F}_q^{M \times M}$ *such that* $\mathbf{H} + g(\mathbf{H}) = \mathbf{I}, \quad \forall \mathbf{H}$ *where* $\mathbf{I}$ *is the identity matrix.*

Proof. Let $f : \mathbb{F}_q \rightarrow \mathbb{F}_q$ be the bijection such that $f(\alpha) + \alpha = 1$ for all $\alpha \in \mathbb{F}_q$. Since $\mathbb{F}_q$ is a finite field, $f(\cdot)$ is guaranteed to exist. Then, define $g(\cdot)$ as follows:

$$g(\mathbf{H}) = \begin{bmatrix} f(h_{11}) & -h_{12} & \cdots & -h_{1M} \\ -h_{21} & f(h_{22}) & \cdots & -h_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ -h_{M1} & -h_{M2} & \cdots & f(h_{MM}) \end{bmatrix} \tag{7.62}$$

where $-h_{k\ell}$ is the additive inverse of $h_{k\ell}$. Clearly, $g(\mathbf{H}) + \mathbf{H} = \mathbf{I}$ and $g(\cdot)$ is a bijection. □

The basic idea underlying our scheme is to add together two well-chosen channel outputs such that the interference exactly cancels out. However, for the finite field model, if we do this in an uncoded fashion, we risk accumulating noise. Thus, we denoise the desired linear functions using the computation code from Theorem 11 prior to combining them together.

We will now show that all users can achieve half the single user rate simultaneously. Note if a transmitter-receiver pair had the channel to themselves, they could achieve an interference-free rate of $\log q - H(Z)$.

Theorem 39. *For the M-user finite field interference channel with fast fading, each transmitter can achieve the following rate :*

$$R_{SYM} = \frac{1}{2}(\log q - H(Z)) \tag{7.63}$$

Proof. For any $\epsilon > 0$, let γ be a small positive constant that will be chosen later to satisfy our rate requirement. Using Lemma 27, choose T large enough so that $P(A_\gamma^T) \geq 1 - \frac{\epsilon}{3}$. Assume that γ and T are chosen such that $T(\frac{1}{|\mathcal{H}|} - \gamma)$ is an even integer. Now condition on the event that the sequence of channel matrices, $\mathbf{H}^{[T]}$, is γ-typical. Since the channel coefficients are i.i.d. and uniform, the probability of any channel $\mathbf{H} \in \mathcal{H}$ is $\frac{1}{|\mathcal{H}|}$. Since $\mathbf{H}^{[T]}$ is γ-typical we have that for every $\mathbf{H} \in \mathcal{H}$:

$$T\left(\frac{1}{|\mathcal{H}|} - \gamma\right) \leq \#(\mathbf{H}|\mathbf{H}^{[T]}) \leq T\left(\frac{1}{|\mathcal{H}|} - \gamma\right) \tag{7.64}$$

We will only use the first $T(\frac{1}{|\mathcal{H}|} - \gamma)$ indices for each channel realization $\mathbf{H} \in \mathcal{H}$. This results in losing at most a γ fraction of the total rate. Group together all useable time indices that have channel realization $\mathbf{H}$ and call this set of indices $\mathcal{T}_{\mathbf{H}}$. We will encode for each $\mathcal{T}_{\mathbf{H}}$ separately. For each channel realization $\mathbf{H}$, transmitter ℓ generates a message $\mathbf{w}_{\ell\mathbf{H}} \in \mathbb{F}_q^k$ where $k = \frac{T}{2}(\frac{1}{|\mathcal{H}|} - \gamma)(\log q)^{-1}(R_{\text{SYM}} - \frac{\epsilon}{3})$.

Using a computation code from Theorem 11, each transmitter ℓ sends its message $\mathbf{w}_{\ell\mathbf{H}}$ during the first $\frac{T}{2}(\frac{1}{|\mathcal{H}|} - \gamma)$ time indices in $\mathcal{T}_{\mathbf{H}}$. Receiver m makes an estimate $\hat{\mathbf{u}}_{m\mathbf{H}}$ of $\mathbf{u}_{m\mathbf{H}} = \sum_{\ell=1}^{M} h_{m\ell}\mathbf{w}_{\ell\mathbf{H}}$.

For each channel realization $\mathbf{H} \in \mathcal{H}$, pair up the first $\frac{T}{2}(\frac{1}{|\mathcal{H}|} - \gamma)$ blocks with $\mathbf{H}$ with the last $\frac{T}{2}(\frac{1}{|\mathcal{H}|} - \gamma)$ blocks with $g(\mathbf{H})$ using $g(\cdot)$ from Lemma 29. Since g is a bijection, this procedure pairs up all of the channel indices. During the last $\frac{T}{2}(\frac{1}{|\mathcal{H}|} - \gamma)$ indices with channel $g(\mathbf{H})$, the transmitters resend the message $\mathbf{w}_{\ell\mathbf{H}}$ using a computation code from Theorem 11. The receivers make an estimate $\hat{\mathbf{v}}_{m\mathbf{H}}$ of $\mathbf{v}_{m\mathbf{H}} = \mathbf{v}_{m\mathbf{H}} = f(h_{mm})\mathbf{w}_{m\mathbf{H}} - \sum_{\ell \neq m} h_{m\ell}\mathbf{w}_{\ell\mathbf{H}}$ where $f(\cdot)$ is the function such that $f(h_{m\ell}) + h_{m\ell} = 1$.

For T large enough, the total probability of error for all computation codes is upper bounded by $\frac{\epsilon}{3}$. Receiver m makes an estimate of $\mathbf{w}_{m\mathbf{H}}$ by simply adding up the two equations to get $\hat{\mathbf{w}}_{m\mathbf{H}} = \hat{\mathbf{u}}_{m\mathbf{H}} + \hat{\mathbf{v}}_{m\mathbf{H}}$. Note that the transmitters do not know a priori which time indices will be successfully paired. To deal with this, the transmitters use an erasure code with rate at least $(1-\gamma)R_{\text{SYM}} - \frac{2\epsilon}{3}$ with probability of error no greater than $\frac{\epsilon}{3}$ over all transmissions. By choosing γ small enough, we finally get that each receiver can recover its message at a rate greater than $\frac{1}{2}(\log q - H(Z)) - \epsilon$ with probability of error less than ϵ as desired. □

Theorem 40. *For the M-user finite field interference channel with fast fading, any rate tuple $(R_1, \dots, R_M)$, satisfying the following inequalities is achievable:*

$$R_\ell + R_m \leq \log q - H(Z), \qquad \forall m \neq \ell. \tag{7.65}$$

First, we will give an equivalent description of this rate region and then show that any rate tuple can be achieved by time sharing the symmetric rate point from Theorem 39 and a single user transmission scheme.

Lemma 30. *Assume, without loss of generality, that the users are labeled according to rate in descending order, so that $R_1 \geq R_2 \geq \cdots \geq R_M$. The achievable rate region from Theorem 40 is equivalent to the following rate region:*

$$\begin{aligned} R_1 &\leq \log q - H(Z) \\ R_m &\leq \min\{\log q - H(Z) - R_1, \frac{1}{2}(\log q - H(Z))\}, \quad m \geq 2 \end{aligned} \tag{7.66}$$

Proof. The key idea is that only one user can achieve a rate higher than $\frac{1}{2}(\log q - H(Z))$. From (7.65), we must have that $R_1 + R_2 \leq \log q - H(Z)$ so if $R_1 > \frac{1}{2}(\log q - H(Z))$ all other users must satisfy $R_m \leq \log q - H(Z) - R_1$. If $R_1 \leq \frac{1}{2}(\log q - H(Z))$, then we have that $R_m \leq \frac{1}{2}(\log q - H(Z))$ for all other users since the rates are in descending order. □

Proof of Theorem 40. We show that the equivalent rate region developed by Lemma 30 is achievable by time-sharing. First, we consider the case where $R_1 > \frac{1}{2}(\log q - H(Z))$. Let $\alpha = 2(1 - \frac{R_1}{\log q - H(Z)})$. We allocate αT channel uses to the symmetric scheme from Theorem 39. For, the remaining $(1-\alpha)T$ channel uses, users 2 through M are silent, and user 1 employs a capacity-achieving point-to-point channel code. This results in user 1 achieving its target rate R_1:

$$\begin{aligned} &\frac{\alpha(\log q - H(Z))}{2} + (1-\alpha)(\log q - H(Z)) \\ &= \log q - H(Z) - R_1 - \log q + H(Z) + 2R_1 = R_1 \end{aligned} \tag{7.67}$$

and users 2 through M achieving $R_m = \log q - H(Z) - R_1$. If $R_1 \leq \frac{1}{2}(\log q - H(Z))$, we can achieve any rate point with the use of the symmetric scheme from Theorem 39. □

Finally, we will give an upper bound using the techniques in [21] to show that the achievable rate region in Theorem 40 is the capacity region.

Theorem 41. *For the M-user finite field interference channel with fast fading, the capacity region is the set of all rate tuple $(R_1, \dots, R_M)$ satisfying:*

$$R_\ell + R_m \leq \log q - H(Z), \qquad \forall m \neq \ell. \tag{7.68}$$

Proof. The required upper bound follows from steps similar to those in Appendix II of [21]. Without loss of generality, we upper bound the rates of users 1 and 2. Note that the capacity of the interference channel only depends on the noise marginals. Thus, we can assume that $Z_1[t] = h_{12}[t](h_{22}[t])^{-1}Z_2[t]$. Let $\tilde{Y}_2[t] = h_{12}[t](h_{22}[t])^{-1}Y_2[t]$.

We give the receivers full access to the messages from users 3 through M as this can only increase the upper bound. Let $\epsilon_T = (R_1 + R_2)P_e + h_B(P_e)$ where P_e is the probability of error. From Fano's inequality, we have that $T(R_1 + R_2)$ is upper bounded as follows:

$$\begin{aligned}
T(R_1 + R_2) &\leq I(w_1; Y_1^T) + I(w_2; w_1, \tilde{Y}_2^T) + T\epsilon_T \\
&= I(w_1; Y_1^T) + I(w_2; \tilde{Y}_2^T | w_1, X_1^T) + T\epsilon_T \\
&= I(w_1; Y_1^T) + I(w_2; \{h_{12}(t)X_2(t) + Z_1(t)\}_{t=1}^T | w_1, X_1^T) + T\epsilon_T \\
&= I(w_1; Y_1^T) \cdots \\
&\qquad \cdots + I(w_2; \{h_{11}(t)X_1(t) + h_{12}(t)X_2(t) + Z_1(t)\}_{t=1}^T | w_1, X_1^T) + T\epsilon_T \\
&= I(w_1; Y_1^T) + I(w_2; Y_1^T | w_1) + T\epsilon_T \\
&= I(w_1, w_2; Y_1^T) + T\epsilon_T \\
&\leq T(\log q - H(Z)) + T\epsilon_T
\end{aligned}$$

As the probability of error P_e tends to zero, $\epsilon_T \to 0$ which yields $R_1 + R_2 \leq \log q - H(Z)$. Similar outer bounds hold for all receiver pairs ℓ and m. Comparing these to the achievable region in Theorem 40 yields the capacity region. □

Overall, ergodic interference alignment shows how much can be gained by coding over parallel interference channels. While in the Gaussian case, we can simply add up two well-matched channel outputs, in general, we can think about this alignment scheme as organizing the computations naturally provided by the channel.

Chapter 8

Conclusions

In this thesis, we have shown that it is possible to efficiently and reliably evaluate a function over a noisy channel. Through codes with an appropriately matched algebraic structure, the redundancy required to overcome noise can be added in a distributed fashion. These computation codes are clearly useful in situations where we are interested in a function of the observed data. In many communication networks, the goal is to exchange messages between users, not functions thereof. In this case, we demonstrated that it is beneficial to view interference between transmitters as *implicit computation* and structure our communication strategy accordingly. This lead us to the compute-and-forward relaying strategy: intermediate nodes decode the function of messages that is available at the highest rate and pass it towards the destination which, given enough functions, can infer its desired messages. Since this strategy works with the interference inherent to a communication network, the end-to-end throughputs are often higher than for the usual routing strategies that fight the interference.

We close with a discussion of some research directions that are a natural extension of the problems considered in this thesis.

Directions

- **A New Architecture for Wireless Networks.** The current modality for most wireless networks is a hub-and-spoke architecture. Users in a network are assigned to a single basestation which uses multiple-access codes for the uplink (many-to-one) and broadcast codes for the downlink (one-to-many). The basestations are in turn linked up by a wired network. As the number of wireless devices increases, this paradigm may eventually be replaced by a many-to-many architecture that makes use of advances in cooperative communications. Compute-and-forward provides a modular, digital solution for the physical layer of a many-to-many architecture: relay nodes work with equations of bits (or symbols over a finite field). Moreover, the underlying computation codes are linear, a condition usually imposed for its practical merits. While in this thesis we focused on finding the linear codes with the highest rates, in practice we can use any linear code. Of course, there are many other issues to consider, such as synchronization, but we believe that these can be overcome using the same techniques used in current systems. From one point of view, the usual encoding and decoding process is essentially unchanged under compute-and-forward except that the encoders all use the same codebook and the decoder acts as if there were only one transmitter.

- **Necessity of Algebraic Structure.** Our results make a strong case that algebraic structure is useful for proving capacity theorems for networks. Yet, we do not know if such structure is *necessary.* Although new upper bound techniques will be needed to answer such a question, we conjecture that algebraic structure is indeed required. The answer would have both philosophical and practical consequences as it would shed some light on what, if not bits, should be the currency of distributed information.

- **Beyond Linear Functions.** The bulk of our results are for scenarios where both the channel and the desired function are linear with respect to some field. This is, in part, due to the availability of good linear and lattice code constructions. Ideally, we would like to find non-linear structured codes that can be used towards building computation codes for general channels and functions. It may be that without the symmetries inherent to linear codes, it may be very difficult to build and analyze a structured code. In the meantime, progress can be made by extending the hybrid computation coding strategy from Chapter 3 in which users first transmit uncoded to take advantage of the channel's natural (noisy) function and then send linear update bins.

- **Outer Bounds** Most outer bound arguments focus on the statistical dependencies between users in a network. Based on the network topology and the channel characteristics, they attempt to place limits on both the probability distributions that can be generated and the resulting mutual information expressions. Clearly, these arguments do not suffice for the problems considered in this thesis as they do not address the role of structural mismatch. If the channel provides a (noisy) function but we would like to evaluate another, what is the penalty? This requires new tools that focus on both the algebraic and the statistical constraints in the problem.

Overall, our aim has to been to challenge the current thinking about computation and noise. Usually, we try to eliminate noise first and then perform our desired computations. Our results provide a glimpse of what is possible if we allow for noisy, local computations while aiming for reliable, global objectives.

Bibliography

[1] Agrawal, S., and Vishwanath, S. On the secrecy rate of interference networks using structured codes. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2009)* (Seoul, South Korea, June 2009).

[2] Ahlswede, R. Group codes do not achieve Shannon's channel capacity for general discrete channels. *The Annals of Mathematical Statistics 42*, 1 (February 1971), 224–240.

[3] Ahlswede, R. Multi-way communication channels. In *Proceedings of the 2nd International Symposium on Information Theory, Prague* (1971), Publishing House of the Hungarian Academy of Sciences, pp. 23–52.

[4] Ahlswede, R., Cai, N., Li, S.-Y. R., and Yeung, R. W. Network information flow. *IEEE Transactions on Information Theory 46*, 4 (July 2000), 1204–1216.

[5] Ahlswede, R., and Han, T. S. On source coding with side information via a multiple-access channel and related problems in multi-user information theory. *IEEE Transactions on Information Theory 29*, 3 (May 1983), 396–412.

[6] Aleksic, M., Razaghi, P., and Yu, W. Capacity of a class of modulo-sum relay channels. *IEEE Transactions on Information Theory 55*, 3 (March 2009), 921–930.

[7] Alon, N., and Spencer, J. H. *The Probabilistic Method*, 2nd ed. Wiley-Interscience, New York, NY, 2000.

[8] Annapureddy, V. S., and Veeravalli, V. V. Gaussian interference networks: Sum capacity in the low-interference regime and new outer bounds on the capacity region. *IEEE Transactions on Information Theory 55*, 7 (July 2009), 3032–3050.

[9] Appuswamy, R., Franceschetti, M., Karamchandani, N., and Zeger, K. Network coding for computing. In *46th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2008).

[10] Aysal, T. C., Coates, M. J., and Rabbat, M. G. Rates of convergence of distributed average consensus using probabilistic quantization. In *45th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2007).

[11] Aysal, T. C., Yildiz, M. E., Sarwate, A. D., and Scaglione, A. Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal Processing 57*, 7 (July 2009), 2748–2761.

[12] Bajwa, W. U., Sayeed, A. M., and Nowak, R. Matched source-channel communication for field estimation in wireless sensor networks. In *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2005)* (Los Angeles, CA, April 2005), pp. 332–339.

[13] Benezit, F., Dimakis, A. G., Thiran, P., and Vetterli, M. Order-optimal consensus through randomized path averaging. In *Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2007).

[14] Berger, T., Zhang, Z., and Viswanathan, H. The CEO problem. *IEEE Transactions on Information Theory 42*, 3 (May 1996), 887–902.

[15] Bhadra, S., Gupta, P., and Shakkottai, S. On network coding for interference networks. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2006)* (Seattle, WA, July 2006).

[16] Borade, S., Zheng, L., and Gallager, R. Amplify-and-forward in wireless relay networks: Rate, diversity, and network size. *IEEE Transactions on Information Theory 53*, 10 (October 2007), 3302–3318.

[17] Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. Analysis and optimization of randomized gossip algorithms. In *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC 2004)* (Atlantis, Bahamas, December 2004).

[18] Boyd, S., Ghosh, A., Prabhakar, B., and Shah, D. Randomized gossip algorithms. *IEEE Transactions on Information Theory 52*, 6 (June 2006), 2508–2530.

[19] Bresler, G., Parekh, A., and Tse, D. The approximate capacity of a one-sided interference channel. In *45th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2007).

[20] Bresler, G., and Tse, D. N. C. 3 user interference channel: Degrees of freedom as a function of channel diversity. In *47th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2009).

[21] Cadambe, V. R., and Jafar, S. A. Interference alignment and the degrees of freedom for the K user interference channel. *IEEE Transactions on Information Theory 54*, 8 (August 2008), 3425–3441.

[22] Cadambe, V. R., and Jafar, S. A. Interference alignment and the degrees of freedom of wireless X networks. *IEEE Transactions on Information Theory 55*, 5 (May 2009), 2334–2344.

[23] Cadambe, V. R., and Jafar, S. A. Multiple access outerbounds and the inseparability of parallel interference channels. *IEEE Transactions on Information Theory 55*, 9 (September 2009), 3983–3990.

[24] Cadambe, V. R., Jafar, S. A., and Wang, C. Interference alignment with asymmetric complex signaling - settling the Host-Madsen-Nosratinia conjecture. In *IEEE Transactions on Information Theory* (Submitted April 2009). See http://arxiv.org/abs/0904.0274.

[25] Caire, G., and Shamai (Shitz), S. On the capacity of some channels with channel state information. *IEEE Transactions on Information Theory 45*, 6 (September 1999), 2007–2019.

[26] Carleial, A. B. Interference channels. *IEEE Transactions on Information Theory 21*, 5 (September 1975), 569–570.

[27] Chen, J., Zhang, X., Berger, T., and Wicker, S. An upper bound on the sum-rate distortion function and its corresponding rate allocation schemes for the CEO problem. *IEEE Journal on Selected Areas in Communications 22*, 6 (August 2004), 977–987.

[28] Cover, T., El Gamal, A., and Salehi, M. Multiple access channels with arbitrarily correlated sources. *IEEE Transactions on Information Theory 26*, 6 (November 1980), 648–657.

[29] Cover, T., and Thomas, J. *Elements of Information Theory*, 2nd ed. Wiley-Interscience, Hoboken, NJ, 2006.

[30] Cover, T. M. A proof of the data compression theorem of Slepian and Wolf for ergodic sources. *IEEE Transactions on Information Theory 21*, 3 (March 1975), 226–228.

[31] Cover, T. M. Comments on broadcast channels. *IEEE Transactions on Information Theory 44*, 6 (October 1998), 2524–2530.

[32] Cover, T. M., and El Gamal, A. Capacity theorems for the relay channel. *IEEE Transactions on Information Theory 25*, 5 (September 1979), 572–584.

[33] Dabora, R., and Servetto, S. D. On the role of estimate-and-forward with time sharing in cooperative communication. *IEEE Transactions on Information Theory 54*, 10 (October 2008), 4409–4431.

[34] Dimakis, A. G., Sarwate, A. D., and Wainwright, M. J. Geographic gossip: Efficient aggregation for sensor networks. *IEEE Transactions on Signal Processing 56*, 3 (March 2008), 1205–1216.

[35] Dobrushin, R. L. Asymptotic optimality of group and systematic codes for some channels. *Theory of Probability and its Applications 8*, 1 (1963), 47–59.

[36] Dohler, M., Gkelias, A., and Aghvami, H. 2-hop distributed MIMO communication system. *Electronics Letters 39*, 18 (September 2003), 1350–1351.

[37] Doshi, V., Shah, D., Médard, M., and Jaggi, S. Distributed functional compression through graph coloring. In *Data Compression Conference (DCC 2007)* (Snowbird, UT, March 2007).

[38] El Gamal, A., Hassanpour, N., and Mammen, J. Relay networks with delays. *IEEE Transactions on Information Theory 53*, 10 (October 2007), 3413–3431.

[39] Elias, P. Coding for noisy channels. *IRE Convention Record 4* (1955), 37–46.

[40] Elias, P., Feinstein, A., and Shannon, C. E. A note on the maximum flow through a network. *IRE Transactions on Information Theory 2*, 4 (December 1956), 117–119.

[41] Erez, U., Litsyn, S., and Zamir, R. Lattices which are good for (almost) everything. *IEEE Transactions on Information Theory 51*, 10 (October 2005), 3401–3416.

[42] Erez, U., and Zamir, R. Achieving $\frac{1}{2}\log(1+\mathrm{SNR})$ on the AWGN channel with lattice encoding and decoding. *IEEE Transactions on Information Theory 50*, 10 (October 2004), 2293–2314.

[43] Erez, U., and Zamir, R. A modulo-lattice transformation for multiple-access channels. In *Proceedings of the 25th Annual Convention of Electrical and Electronic Engineers in Israel* (Eilat, Israel, December 2008).

[44] Etkin, R. H., Tse, D. N. C., and Wang, H. Gaussian interference channel capacity to within one bit. *IEEE Transactions on Information Theory 54*, 12 (December 2008), 5534–5562.

[45] FENG, H., EFFROS, M., AND SAVARI, S. Functional source coding for networks with receiver side information. In *42nd Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2004).

[46] FLAJOLET, P., AND MARTIN, G. Probabilistic counting algorithms for data base applications. *Journal of Computer and System Sciences 31*, 2 (1985), 182–209.

[47] FORD, L. R., AND FULKERSON, D. R. Maximal flow through a network. *Canadian Journal of Mathematics 8* (1956), 399–404.

[48] FORD, L. R., AND FULKERSON, D. R. *Flows in Networks.* Princeton University Press, Princeton, NJ, 1962.

[49] GALLAGER, R. *Information Theory and Reliable Communication.* John Wiley and Sons, Inc., New York, 1968.

[50] GAMAL, H. E., CAIRE, G., AND DAMEN, M. O. Lattice coding and decoding achieve the optimal diversity-multiplexing tradeoff of MIMO channels. *IEEE Transactions on Information Theory 50*, 6 (June 2004), 968–985.

[51] GASTPAR, M. *To Code or Not To Code.* PhD thesis, EPFL, 2002.

[52] GASTPAR, M. Uncoded transmission is exactly optimal for a simple Gaussian "sensor" network. *IEEE Transactions on Information Theory 54*, 11 (November 2008), 5427–5251.

[53] GASTPAR, M., AND VETTERLI, M. On the capacity of wireless networks: The relay case. In *Proceedings of the 21st Annual International Conference on Computer Communications (INFOCOM 2002)* (New York, NY, June 2002), vol. 3, pp. 1577 – 1586.

[54] GASTPAR, M., AND VETTERLI, M. Source-channel communication in sensor networks. In *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2003)* (New York, NY, April 2003), L. J. Guibas and F. Zhao, Eds., Lecture Notes in Computer Science, Springer, pp. 167–177.

[55] GASTPAR, M., AND VETTERLI, M. On the capacity of large Gaussian relay networks. *IEEE Transactions on Information Theory 51*, 3 (March 2005), 765–779.

[56] GIRIDHAR, A., AND KUMAR, P. R. Computing and communicating functions over sensor networks. *IEEE Journal on Selected Areas in Communications 23*, 4 (April 2005), 755–764.

[57] Guruswami, V. Rapidly mixing markov chains: A comparison of techniques. Unpublished. See http://www.cs.cmu.edu/∼venkatg/pubs/papers/markov-survey.ps, 2000.

[58] Han, T. S., and Kobayashi, K. A new achievable rate region for the interference channel. *IEEE Transactions on Information Theory 27*, 1 (January 1981), 49–60.

[59] He, X., and Yener, A. Providing secrecy with structured codes: Tools and applications to two-user Gaussian channels. *IEEE Transactions on Information Theory* (Submitted July 2009). See http://arxiv.org/abs/0907.5388.

[60] Ho, T., Karger, D. R., Médard, M., and Koetter, R. Network coding from a network flow perspective. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2003)* (Yokohama, Japan, June 2003).

[61] Ho, T., Medard, M., Koetter, R., Karger, D. R., Effros, M., Shi, J., and Leong, B. A random linear network coding approach to multicast. *IEEE Transactions on Information Theory 52*, 10 (October 2006), 4413–4430.

[62] I. Csiszár. Linear codes for sources and source networks: Error exponents, universal coding. *IEEE Transactions on Information Theory 28*, 4 (July 1982), 585–592.

[63] I. Csiszár, and Körner, J. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, New York, 1982.

[64] Jafar, S. A. The ergodic capacity of interference networks. *IEEE Transactions on Information Theory* (Submitted July 2009). See http://arxiv.org/abs/0902.0838.

[65] Jafar, S. A., and Shamai (Shitz), S. Degrees of freedom region for the MIMO X channel. *IEEE Transactions on Information Theory 54*, 1 (January 2008), 151–170.

[66] Jaggi, S., Chou, P., and Jain, K. Low complexity algebraic network codes. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2003)* (Yokohama, Japan, June 2003).

[67] Jeon, S.-W., and Chung, S.-Y. Capacity of a class of multi-source relay networks. *IEEE Transactions on Information Theory* (Submitted July 2009). See http://arxiv.org/abs/0907.2510.

[68] Katti, S., Gollakota, S., and Katabi, D. Embracing wireless interference: Analog network coding. In *ACM SIGCOMM* (Kyoto, Japan, August 2007).

[69] Katti, S., Rahul, H., Hu, W., Katabi, D., Médard, M., and Crowcroft, J. XORs in the air: Practical wireless network coding. In *ACM SIGCOMM* (Pisa, Italy, September 2006).

[70] Kawadia, V., and Kumar, P. R. A cautionary perspective on cross-layer design. *IEEE Wireless Communications Magazine 12*, 1 (February 2005), 3–11.

[71] Khude, N., Prabhakaran, V., and Viswanath, P. Harnessing bursty interference. In *Proceedings of the IEEE Information Theory Workshop (ITW 2009)* (Volos, Greece, June 2009).

[72] Kim, Y.-H. Capacity of a class of deterministic relay channels. *IEEE Transactions on Information Theory 54*, 3 (March 2008), 1328–1329.

[73] Kirti, S., Scaglione, A., and Thomas, R. J. A scalable wireless communication architecture for average consensus. In *Proceedings of the 46th IEEE Conference on Decision and Control (CDC 2007)* (New Orleans, LA, December 2007).

[74] Kochman, Y., and Zamir, R. Analog matching of colored sources to colored channels. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2006)* (Seattle, WA, July 2006).

[75] Kochman, Y., and Zamir, R. Joint Wyner-Ziv/dirty-paper coding by modulo-lattice modulation. *IEEE Transactions on Information Theory 55*, 11 (November 2009), 4878–4889.

[76] Koetter, R., and Medard, M. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking 11* (October 2003), 782–795.

[77] Körner, J., and Marton, K. How to encode the modulo-two sum of binary sources. *IEEE Transactions on Information Theory 25*, 2 (March 1979), 219–221.

[78] Kotelnikov, V. A. On the carrying capacity of the ether and wire in telecommunications. In *Material for the First All-Union Conference on the Technological Reconstruction of the Communications Sector and Low-Current Engineering* (Moscow, Russia, 1933).

[79] Kramer, G., Gastpar, M., and Gupta, P. Cooperative strategies and capacity theorems for relay networks. *IEEE Transactions on Information Theory 51*, 9 (September 2005), 3037–3063.

[80] Krithivasan, D., and Pradhan, S. Lattices for distributed source coding: Jointly Gaussian sources and reconstruction of a linear function. *IEEE Transactions on Information Theory* (Submitted July 2007). See http://arxiv.org/abs/0707.3461.

[81] Krithivasan, D., and Pradhan, S. A proof of the existence of good lattices. Tech. rep., University of Michigan, July 2007. See http://www.eecs.umich.edu/techreports/systems/cspl/cspl-384.pdf.

[82] Krithivasan, D., and Pradhan, S. Distribued source coding using Abelian group codes. *IEEE Transactions on Information Theory* (Submitted August 2008). See http://arxiv.org/abs/0808.2659.

[83] Laneman, J. N., Tse, D. N. C., and Wornell, G. W. Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *IEEE Transactions on Information Theory 50*, 12 (December 2004), 3062–3080.

[84] Li, S.-Y. R., Yeung, R. W., and Cai, N. Linear network coding. *IEEE Transactions on Information Theory 49*, 2 (February 2003), 371–381.

[85] Li, W., and Dai, H. Location-aided fast distributed consensus. In *IEEE Transactions on Information Theory* (Submitted July 2007). See http://arxiv.org/abs/0707.0500.

[86] Liao, H. *Multiple access channels*. PhD thesis, University of Hawaii, Honolulu, 1972.

[87] Liu, P., Tao, Z., Narayanan, S., Korakis, T., and Panwar, S. S. CoopMAC: A cooperative MAC for wireless LANs. *IEEE Journal on Selected Areas in Communications 25*, 2 (February 2007), 340–354.

[88] Loeliger, H.-A. Averaging bounds for lattices and linear codes. *IEEE Transactions on Information Theory 43*, 6 (November 1997), 1767–1773.

[89] Ma, N., and Ishwar, P. Distributed source coding for interactive function computation. *IEEE Transactions on Information Theory* (Submitted November 2008). See http://arxiv.org/abs/0801.0756.

[90] Ma, N., Ishwar, P., and Gupta, P. Information-theoretic bounds for multiround function computation in collocated networks. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2009)* (Seoul, South Korea, June 2009).

[91] Maddah-Ali, M. A., Motahari, A. S., and Khandani, A. K. Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis. *IEEE Transactions on Information Theory 54*, 8 (August 2008), 3457–3470.

[92] Marsch, P., and Fettweis, G. On backhaul-constrained multi-cell cooperative detection based on superposition coding. In *Proceedings of the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2008)* (Cannes, France, September 2008).

[93] Mergen, G., and Tong, L. Type based estimation over multiaccess channels. *IEEE Transactions on Signal Processing 54*, 2 (February 2006), 613–626.

[94] Mosk-Aoyama, D., and Shah, D. Information dissemination via gossip: Applications to averaging and coding. http://arxiv.org/cs.NI/0504029, April 2005.

[95] Motahari, A. S., Gharan, S. O., Maddah-Ali, M.-A., and Khandani, A. K. Real interference alignment: Exploiting the potential of single antenna systems. *IEEE Transactions on Information Theory* (Submitted November 2009). See http://arxiv.org/abs/0908.2282.

[96] Motahari, A. S., and Khandani, A. K. Capacity bounds for the Gaussian interference channel. *IEEE Transactions on Information Theory 55*, 2 (February 2009), 620–643.

[97] Mudumbai, R., Wild, B., Madhow, U., and Ramchandran, K. Distributed beamforming using 1 bit feedback: From concept to realization. In *44th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2006).

[98] Murphy, P., Sabharwal, A., and Aazhang, B. Building a cooperative communications system. Tech. rep., Rice University, July 2007. http://arxiv.org/pdf/0707.2998.

[99] Nam, W., Chung, S.-Y., and Lee, Y. H. Capacity bounds for two-way relay channels. In *Proceedings of the International Zurich Seminar on Communications (IZS 2008)* (Zurich, Switzerland, March 2008).

[100] Nam, W., Chung, S.-Y., and Lee, Y. H. Nested lattice codes for Gaussian relay networks with interference. *IEEE Transactions on Information Theory* (Submitted February 2009). See http://arxiv.org/abs/0902.2436.

[101] Narayanan, K., Wilson, M. P., and Sprintson, A. Joint physical layer coding and network coding for bi-directional relaying. In *45th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2007).

[102] Nazer, B. Note on structured superposition decoding matrix. Unpublished, January 2009.

[103] Nazer, B., Dimakis, A. G., and Gastpar, M. Local inteference can accelerate gossip algorithms. In *46th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2008).

[104] Nazer, B., and Gastpar, M. Computation over multiple-access channels. *IEEE Transactions on Information Theory 53*, 10 (October 2007), 3498–3516.

[105] Nazer, B., and Gastpar, M. Lattice coding increases multicast rates for Gaussian multiple-access networks. In *45th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2007).

[106] Nazer, B., and Gastpar, M. The case for structured random codes in network capacity theorems. *European Transactions on Telecommunications 19*, 4 (June 2008), 455–474.

[107] Nazer, B., and Gastpar, M. Structured random codes and sensor network coding theorems. In *Proceedings of the International Zurich Seminar on Communications (IZS 2008)* (Zurich, Switzerland, March 2008).

[108] Nazer, B., and Gastpar, M. Compute-and-forward: Harnessing interference through structured codes. *IEEE Transactions on Information Theory* (Submitted August 2009). See http://arxiv.org/abs/0908.2119.

[109] Nazer, B., Gastpar, M., Jafar, S. A., and Vishwanath, S. Ergodic interference alignment. In *Proceedings of the International Symposium on Information Theory (ISIT 2009)* (Seoul, South Korea, June 2009).

[110] Nazer, B., Gastpar, M., Jafar, S. A., and Vishwanath, S. Interference alignment at finite SNR: General message sets. In *47th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2009).

[111] Nazer, B., Sanderovich, A., Gastpar, M., and Shamai (Shitz), S. Structured superposition for backhaul constrained cellular uplink. In *Proceedings of the International Symposium on Information Theory (ISIT 2009)* (Seoul, South Korea, June 2009).

[112] Nedic, A., Olshevsky, A., Ozdaglar, A., and Tsitsiklis, J. On distributed averaging algorithms and quantization effects. Tech. Rep. 2778, MIT, November 2007. See http://arxiv.org/abs/0711.4179.

[113] Nyquist, H. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers (AIEE) 47* (April 1928), 617–644.

[114] Ochiai, H., Mitran, P., Poor, H. V., and Tarokh, V. Collaborative beamforming for distributed wireless ad hoc sensor networks. *IEEE Transactions on Signal Processing 53*, 11 (November 2005), 4110–4124.

[115] Oechtering, T. J., Schnurr, C., Bjelakovic, I., and Boche, H. Broadcast capacity region of two-phase bidirectional relaying. *IEEE Transactions on Information Theory 54*, 1 (January 2008), 454–458.

[116] Oohama, Y. The rate-distortion function for the quadratic Gaussian CEO problem. *IEEE Transactions on Information Theory 44*, 3 (May 1998), 1057–1070.

[117] Orlitsky, A., and Roche, J. R. Coding for computing. *IEEE Transactions on Information Theory 47*, 3 (March 2001), 903–917.

[118] Özgür, A., Lévêque, O., and Tse, D. N. C. Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks. *IEEE Transactions on Information Theory 53*, 10 (October 2007), 3549–3572.

[119] Özgür, A., and Tse, D. N. C. Achieving linear scaling with interference alignment. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2009)* (Seoul, South Korea, June 2009).

[120] P. Frasca, R. Carli, F. F., and Zampieri., S. Average consensus on networks with quantized communication. *International Journal of Robust and Nonlinear Control 19*, 16 (November 2009), 1787–1816.

[121] Philosof, T., Zamir, R., Erez, U., and Khisti, A. Lattice strategies for the dirty multiple access channel. *IEEE Transactions on Information Theory* (Submitted April 2009). See http://arxiv.org/abs/0904.1892.

[122] Poltyrev, G. On coding without restrictions for the AWGN channel. *IEEE Transactions on Information Theory 40*, 2 (March 1994), 409–417.

[123] Prabhakaran, V., Tse, D., and Ramchandran, K. Rate region of the quadratic Gaussian CEO problem. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2004)* (Chicago, IL, July 2004).

[124] Rabbat, M., Haupt, J., A.Singh, and Nowak, R. Decentralized compression and predistribution via randomized gossiping. In *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2006)* (Nashville, TN, April 2006).

[125] Rajagopal, R., and Wainwright, M. J. Network-based consensus averaging with general noisy channels. In *45th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2007). See http://arxiv.org/abs/0805.0438.

[126] Ratnakar, N., and Kramer, G. The multicast capacity of deterministic relay networks with no interference. *IEEE Transactions on Information Theory 52*, 6 (June 2006), 2425–2432.

[127] Rogers, C. A. Lattice coverings of space. *Mathematica 6* (1959), 33–39.

[128] Saligrama, V., Alanyali, M., and Savas, O. Distributed detection in sensor networks with packet losses and finite capacity links. *IEEE Transactions on Signal Processing 54*, 11 (November 2006), 4118–4132.

[129] Sanderovich, A., Peleg, M., and Shamai (Shitz), S. Scaling laws in decentralized processing of interfered Gaussian channels. In *Proceedings of the International Zurich Seminar on Communications (IZS 2008)* (Zurich, Switzerland, March 2008).

[130] Sanderovich, A., Shamai (Shitz), S., and Steinberg, Y. Distributed MIMO receiver - achievable rates and upper bounds. *IEEE Transactions on Information Theory 55*, 10 (October 2009), 4419–4438.

[131] Sanderovich, A., Somekh, O., Poor, H. V., and Shamai (Shitz), S. Uplink macro diversity of limited backhaul cellular network. *IEEE Transactions on Information Theory 55*, 8 (August 2009), 3457–3478.

[132] Sanders, P., Egner, S., and Tolhuizen, L. Polynomial time algorithms for network information flow. In *15th ACM Symposium on Parallel Algorithms and Architectures* (2003), pp. 286–294.

[133] Sankar, L., Shang, X., Erkip, E., and Poor, H. V. Ergodic two-user interference channels: Is separability optimal? In *46th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2008).

[134] Sarwate, A. D., and Dimakis, A. G. The impact of mobility on gossip algorithms. In *Proceedings of the 28th Annual International Conference on Computer Communications (INFOCOM 2009)* (Rio de Janeiro, Brazil, April 2009).

[135] Sato, H. The capacity of the Gaussian interference channel under strong interference. *IEEE Transactions on Information Theory 27*, 6 (November 1981), 786–788.

[136] Schein, B., and Gallager, R. G. The Gaussian parallel relay network. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2000)* (Sorrento, Italy, June 2000).

[137] Sendonaris, A., Erkip, E., and Aazhang, B. User cooperation diversity - part I: System description. *IEEE Transactions on Communications 51*, 11 (November 2003), 1927–1938.

[138] Shamai (Shitz), S., Simeone, O., Somekh, O., Sanderovich, A., Zaidel, B., and Poor, H. V. Information theoretic implications of constrained cooperation in simple cellular models. In *Proceedings of the IEEE International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC 2008)* (Cannes, France, September 2008).

[139] Shamai (Shitz), S., and Verdú, S. Capacity of channels with uncoded side information. *European Transactions on Telecommunications 6*, 5 (September-October 1995), 587–600.

[140] Shamai (Shitz), S., Verdú, S., and Zamir, R. Systematic lossy source/channel coding. *IEEE Transactions on Information Theory 44*, 2 (March 1998), 564–579.

[141] Shang, X., Kramer, G., and Chen, B. A new outer bound and the noisy-interference sum-rate capacity for Gaussian interference channels. *IEEE Transactions on Information Theory 55*, 2 (February 2009), 689–699.

[142] Shannon, C. E. A mathematical theory of communication. *Bell System Technical Journal 27* (1948), 379–423, 623–656.

[143] Shannon, C. E. Communication in the presence of noise. *Proceedings of the Institute of Radio Engineers 37*, 1 (January 1949), 10–21.

[144] Sinclair, A. Improved bounds for mixing rates of markov chains and multicommodity flow. *Combinatorics, Probability and Computing 1*, 4 (December 1992), 351–370.

[145] Slepian, D., and Wolf, J. Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory 19*, 4 (July 1973), 471–480.

[146] Somekh, O., Zaidel, B. M., and Shamai (Shitz), S. Sum rate characterization of joint multiple cell-site processing. *IEEE Transactions on Information Theory 53*, 12 (December 2007), 4473–4497.

[147] Soundararajan, R., and Vishwanath, S. Communicating the difference of correlated Gaussian sources over a MAC. In *Data Compression Conference (DCC 2009)* (Snowbird, UT, March 2009).

[148] Spanos, D., Olfati-Saber, R., and Murray, R. Distributed Kalman filtering in sensor networks with quantifiable performance. In *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2005)* (Los Angeles, CA, April 2005).

[149] Sridharan, S., Jafarian, A., Vishwanath, S., Jafar, S. A., and Shamai (Shitz), S. A layered lattice coding scheme for a class of three user Gaussian interference channels. In *46th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2008).

[150] Sundaram, S., and Hadjicostis, C. N. Distributed consensus and linear functional calculation: An observability perspective. In *Proceedings of the ACM/IEEE*

International Conference on Information Processing in Sensor Networks (IPSN 2007) (Cambridge, MA, April 2007).

[151] Telatar, E. Capacity of multi-antenna Gaussian channels. *European Transactions on Telecommunications 10*, 6 (November - December 1999), 585–595.

[152] Tuninetti, D. Gaussian fading interference channels: Power control. In *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers* (Monterey, CA, October 2008).

[153] Ulukus, S., and Kang, W. A single-letter upper bound for the sum rate of multiple access channels with correlated sources. In *Proceedings of the 39th Asilomar Conference on Signals, Systems and Computers* (2005).

[154] Viswanathan, H., and Berger, T. The quadratic Gaussian CEO problem. *IEEE Transactions on Information Theory 43*, 5 (September 1997), 1549–1559.

[155] Wagner, A. B. On distributed compression of linear functions. In *46th Annual Allerton Conference on Communications, Control, and Computing* (Monticello, IL, September 2008).

[156] Wu, Y., and Dimakis, A. G. Reducing repair traffic for erasure coding-based storage via interference alignment. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT 2009)* (Seoul, South Korea, June 2009).

[157] Wyner, A. D. Shannon-theoretic approach to a Gaussian cellular multiple-access channel. *IEEE Transactions on Information Theory 40*, 6 (November 1994), 1713–1727.

[158] Xiao, L., Boyd, S., and Lall, S. A scheme for asynchronous distributed sensor fusion based on average consensus. In *Proceedings of the ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN 2005)* (Los Angeles, CA, April 2005).

[159] Yamamoto, H. Wyner-Ziv theory for a general function of the correlated sources. *IEEE Transactions on Information Theory 28*, 5 (September 1982), 803–807.

[160] Zamir, R. Lattices are everywhere. In *Proceedings of the 4th Annual Workshop on Information Theory and its Applications (ITA 2009)* (La Jolla, CA, February 2009).

[161] Zamir, R., and Feder, M. On lattice quantization noise. *IEEE Transactions on Information Theory 42*, 4 (July 1996), 1152–1159.

[162] Zamir, R., Shamai (Shitz), S., and Erez, U. Nested linear/lattice codes for structured multiterminal binning. *IEEE Transactions on Information Theory 48*, 6 (June 2002), 1250–1276.

[163] Zhan, J., Nazer, B., Gastpar, M., and Erez, U. MIMO compute-and-forward. In *Proceedings of the International Symposium on Information Theory (ISIT 2009)* (Seoul, South Korea, June 2009).

[164] Zhang, S., Liew, S., and Lam, P. Hot topic: Physical-layer network coding. In *Proceedings of the 12th Annual ACM International Conference on Mobile Computing and Networking (MobiCom 2006)* (Los Angeles, CA, September 2006).