

RICE UNIVERSITY

Cooperative Strategies for Near-Optimal Computation in
Wireless Networks

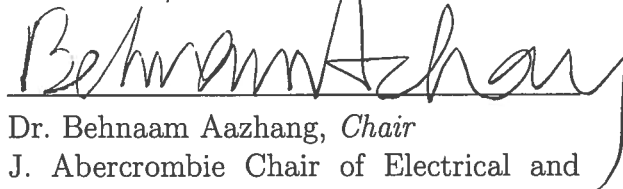
by

Matthew Nokleby

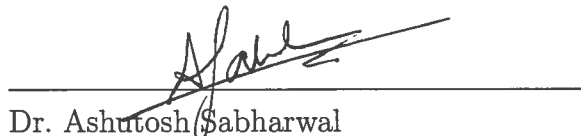
A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Doctor of Philosophy

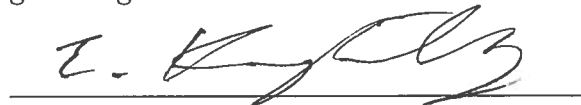
APPROVED, THESIS COMMITTEE:



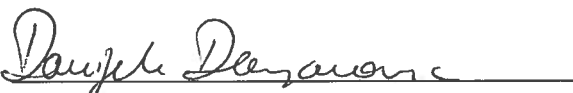
Dr. Behnaam Aazhang, *Chair*
J. Abercrombie Chair of Electrical and
Computer Engineering



Dr. Ashutosh Sabharwal
Professor of Electrical and Computer En-
gineering



Dr. Edward Knightly
Professor of Electrical and Computer En-
gineering



Dr. Danijela Damjanović
Assistant Professor of Mathematics

HOUSTON, TEXAS
NOVEMBER 2012

ABSTRACT

Cooperative Strategies for Near-Optimal Computation in Wireless Networks

by

Matthew Nokleby

Computation problems, such as network coding and averaging consensus, have become increasingly central to the study of wireless networks. Network coding, in which intermediate terminals compute and forward functions of others' messages, is instrumental in establishing the capacity of multicast networks. Averaging consensus, in which terminals compute the mean of others' measurements, is a canonical building block of distributed estimation over sensor networks. Both problems, however, are typically studied over graphical networks, which abstract away the broadcast and superposition properties fundamental to wireless propagation. The performance of computation in realistic wireless environments, therefore, remains unclear.

In this thesis, I seek after near-optimal computation strategies under realistic wireless models. For both network coding and averaging consensus, cooperative communications plays a key role. For network coding, I consider two topologies: a single-layer network in which users may signal cooperatively, and a two-transmitter, two-receiver network aided by a dedicated relay. In the former topology, I develop a decode-and-forward scheme based on a linear decomposition of nested lattice codes. For a network having two transmitters and a single receiver, the proposed

scheme is optimal in the diversity-multiplexing tradeoff; otherwise it provides significant rate gains over existing non-cooperative approaches. In the latter topology, I show that an amplify-and-forward relay strategy is optimal almost everywhere in the degrees-of-freedom. Furthermore, for symmetric channels, amplify-and-forward achieves rates near capacity for a non-trivial set of channel gains.

For averaging consensus, I consider large networks of randomly-placed nodes. Under a path-loss wireless model, I characterize the resource demands of consensus with respect to three metrics: energy expended, time elapsed, and time-bandwidth product consumed. I show that existing consensus strategies, such as gossip algorithms, are nearly order optimal in the energy expended but strictly suboptimal in the other metrics. I propose a new consensus strategy, tailored to the wireless medium and cooperative in nature, termed *hierarchical averaging*. Hierarchical averaging is nearly order optimal in all three metrics for a wide range of path-loss exponents. Finally, I examine consensus under a simple quantization model, showing that hierarchical averaging achieves a nearly order-optimal tradeoff between resource consumption and estimation accuracy.

ACKNOWLEDGEMENTS

Over the course of graduate school, one becomes indebted to far too many people to thank properly in the preface to a thesis. I will do my best.

I express my appreciation to Behnaam Aazhang for his guidance and savvy. To the many fine instructors at Rice University, including Mark Embree, Don Johnson, and Ashutosh Sabharwal for their challenging and enlightening courses. To Waheed U. Bajwa, Robert Calderbank, Natahsa Devroye, and Bobak Nazer for their fruitful collaborations. To Jared Anderson, John Dehlin, Greg Rockwell, Andrew Waters, and so many others for helping to smooth out the transitions. To Prince Chidyagwai and Justin Fritz for their mental toughness. To David Ramirez for the last-minute sprints. To David Kao for Puzzle. To Evan Everett for bearing the brunt of hipster criticism. To Raajen Patel for his bottomless well of charisma. To Gareth Middleton for the bourbon-infused hours spent unravelling the duality of kitty and COM. To Jim Henson for providing a model of the modern scientific powerhouse.

Finally, to Amanda for having sufficient humor to laugh at me, but sufficient kindness not to laugh too hard.

Contents

Abstract	ii
Acknowledgements	iv
1 Introduction	1
1.1 Motivation	1
1.2 State of the Art	6
1.3 Contributions	10
1.4 Notation	14
2 Lattices and Lattice Codes	15
2.1 Lattices	15
2.2 Lattice Codes	18
2.3 The AWGN Channel	20
2.4 Lattice subspaces	22
2.5 A Diversion: Lattice Coding over the Relay Channel	27
3 User Cooperation	35
3.1 Preliminaries	36
3.2 Computation Rates	41
3.3 Diversity-multiplexing Tradeoff	57
3.4 Numerical Examples	73
4 Relay Cooperation	81
4.1 Preliminaries	81

4.2	Standard Relay	86
4.3	Instantaneous relay	92
5	Consensus	102
5.1	Preliminaries	103
5.2	Inner Bounds	109
5.3	Gossip Algorithms	112
5.4	Hierarchical averaging	116
5.5	Quantization	124
6	Conclusions	135
6.1	Summary	135
6.2	Future Directions	137
A	Mutual information of dithered lattices over the multiple-access channel	141
	References	144

List of Figures

2.1	Nested lattice codes. White dots are elements of the coding lattice, and black dots are elements of the shaping lattice. Each lattice point inside the shaded Voronoi region \mathcal{V}_s is a member of the codebook.	20
2.2	Lattice subspace decomposition. Each lattice codeword in $\mathcal{C}^{(n)}$ is the sum of a point in $\mathcal{C}_r^{(n)}$ (upper) and a point in $\mathcal{C}_v^{(n)}$ (lower). The shaded region \mathcal{V}_s defines the codebook, whereas the strip-shaped Voronoi regions \mathcal{V}_r and \mathcal{V}_v define the decoding regions of the resolution and vestigial codebooks, respectively.	26
2.3	The relay channel.	28
3.1	The cooperative computation channel. L users cooperatively transmit to M receivers, which decode the desired linear functions.	36
3.2	Diversity-multiplexing tradeoff for $L = 2$, $L = 5$ transmitters and a single receiver.	72
3.3	A two-by-one computation network with symmetric channel gains.	73
3.4	Achievable rates as a function of inter-transmitter channel gains	74
3.5	Three users are placed along a segment of the unit circle, while the receiver is placed at the origin.	75
3.6	Average computation rate vs. angle between transmitters.	75
3.7	A two-by-one computation network with asymmetric channel gains.	76
3.8	Achievable rates as a function of h_{21} and P	78
3.9	A two-by-two computation network with asymmetric channel gains.	79
3.10	Achievable rates as a function of h_{21} and P	80

4.1	The relay computation channel.	82
4.2	Achievable rates as a function of h and P for the relay computation channel with a standard relay.	93
4.3	Achievable rates as a function of h and P for the relay computation channel with an instantaneous relay.	101
5.1	Hierarchical partition of the network. Each square cell is divided into four smaller cells, which are each divided into four smaller cells, and so on.	117
5.2	Transmit energy E_ϵ and time-bandwidth product B_ϵ for a variety of consensus algorithms.	133
5.3	Total energy E and mean-square error σ^2 for several quantized consensus algorithms.	134

List of Tables

2.1	Lattice block Markov encoding	29
3.1	Superposition Block Markov encoding for Theorem 3.3	50
4.1	Block Markov encoding for compress-and-forward	90

Introduction

1.1 Motivation

Communications tasks fall into one of two broad categories: the exchange of digital messages, and the (perhaps) lossy conveyance of digital or analog sources. In information theory, these tasks correspond to the *channel coding* problem and the *source coding* problem, respectively.

For wireless point-to-point networks, both problems are well understood. In his seminal paper, Shannon [1] solved the channel coding problem by showing that channel capacity is equal to the maximum mutual information between channel input and output and that random codebooks are asymptotically optimal. He further derived the capacity for the additive white Gaussian noise (AWGN) channel, the simplest model for wireless communications. In the ensuing decades, more sophisticated wireless models have been studied. *Frequency-selective* channels, in which multipath and other propagation effects result in a non-uniform frequency response, was studied in [2] and solved with the now-ubiquitous “water-filling” solution. *Fading* channels, in which user mobility causes the channel to vary temporally, has been studied extensively [3–5], and researchers have devised appropriate approaches such as *ergodic*

capacity and *outage capacity*. Finally, the spatial dimension of wireless channels has been exploited via multiple-antenna channels [6], and it has been shown that significant capacity improvements are achieved simply by adding antennas.

Similarly, for source coding in [7] Shannon proved the *rate-distortion theorem*, showing the optimal tradeoff between the amount of information used to describe a memoryless source and the distortion induced by such a description. Further, he proved the *source-channel separation theorem*, which states that an optimal source code and an optimal channel code can be joined to form an optimal code for the lossy transmission of a source over a noisy channel. It is sufficient to encode the source at a rate just below the channel capacity and to transmit the result as a digital message over the channel.

Furthermore, the performance promised by these information-theoretic results has largely been borne out in practice. The development of turbo codes [8], the rediscovery and rehabilitation of low-density parity check (LDPC) codes [9, 10], and the recent advent of polar codes [11] have ushered in an era of near-capacity performance in practical systems. Source coding has fared even better. Near-optimum lossless approaches, such as Lempel-Ziv encoding [12] have been in use for decades. Combining these results with the source-channel separation theorem, it is possible to construct practical, near-optimal source-channel codes over point-to-point wireless channels.

Once one ventures away from a single source transmitted over a point-to-point channel, however, the situation quickly becomes complicated. Barring a few exceptions—such as Gaussian broadcast and multiple-access channels [13]—the capacity of multi-terminal channels remains unsolved. As evidenced by the three-terminal relay channel [14, 15], even the addition of even a single terminal can be enough to prevent a single-letter capacity expression. In recent years, researchers have succeeded in characterizing the approximate capacity of a few classes of multi-terminal channels. The

capacity of two-transmitter, two-receiver interference channels has been characterized to within one bit [16], and reasonably tight upper and lower bounds on the capacity of larger *symmetric* interference channels have been obtained [17]. The development of deterministic channel models [18] has inspired a recent flurry of approximate capacity results, and in particular the capacity of large wireless *multicast* networks – that is, networks in which receiver nodes demand the messages of every source node – has been characterized to within a constant gap [18, 19]. Even these approximate results, however, give little insight into the capacity of more general multi-flow wireless networks.

Similar difficulties arise with source coding. The Slepian-Wolf theorem [20] established the limits of lossless encoding of correlated sources, while Berger and Tung [21, 22] derived inner and outer bounds on the rate-distortion region of lossy encoding. Later Wagner et al. showed that the Berger-Tung bounds coincide for the special case of Gaussian sources and quadratic distortion [23], and very recently Courtade and Weissman proved a similar result for arbitrary sources under logarithmic distortion [24]. However, beyond these and a few other cases, distributed source coding problems remain unsolved. Furthermore, source-channel separation in general fails in multi-terminal settings. Even in scenarios in which capacity is known and optimal source codes exist, therefore, the optimal joint source-channel code is in general unknown. Despite decades of intense study, there yet lacks a comprehensive information theory for wireless networks.

The primary difficulty in network information theory is that signals interact in complicated ways. In wireless networks, this phenomenon is manifest as the *broadcast* and *superposition* natures of the wireless medium. A transmitter’s signal does not merely arrive at its intended destination; it is broadcast to every terminal in the vicinity. Similarly, a receiver hears more than the signal from its intended source; it

receives the superposition of transmissions from any terminal in the vicinity. These signal interactions are usually taken to be disadvantageous, and, as discussed in the previous paragraphs, the optimal means to counteract such disadvantages is largely a mystery.

Consequently, most practical wireless networks operate by eliminating signal interactions. Cellular networks operate via time-division multiple access (TDMA) or code-division multiple access (CDMA). Local-area Wi-Fi (802.11) networks operate via carrier sense medium access (CSMA) to avoid any two nearby transmitters from broadcasting simultaneously. Large-area WiMax (802.16) networks operate via orthogonal frequency-division multiple access (OFDMA). In each case, transmissions are orthogonalized in an appropriate signal space so that, in effect, signals do not interfere. The result is a network composed of parallel point-to-point channels, each of which is well-understood both in theory and in practice.

Indeed, once one orthogonalizes away interference, analysis of even large networks simplifies considerably. Graphical networks, in which links between terminals are represented as lossless links of fixed capacity, suffice to model wireless networks composed of orthogonal point-to-point channels [25]. And there is much one can say about graphical networks. On the channel coding side, the capacity of graphical multicast networks is characterized precisely by the cut-set bound. Ahlswede et al. proved that *network coding*—in which intermediate nodes compute and forward *functions* of incoming data—is sufficient to achieve the cut-set bound in arbitrary graphical multicast networks [26]. On the source coding side, the picture is somewhat less clear, simply because there exists a large variety of problems to consider. In this thesis I will focus on estimation problems targeted at sensor networks: how well can a network infer some function of sources collected at each terminal? In particular, I will address the problem of *averaging consensus*, in which each terminal intends to

compute the arithmetic mean of the sources. For graphical networks, consensus is well understood. There exist efficient averaging algorithms, such as path averaging and multi-scale gossip, that are approximately optimal with respect to the number of transmissions required to achieve consensus [27, 28].

A common thread unifying these two seemingly disparate problems is *function computation*. In network coding intermediate terminals compute functions of incoming messages, and in consensus terminals compute averages of other's data. In either case, the task of computing functions is substantially simplified in graphical networks, since signal interactions are eliminated. However, eliminating signal interactions is accomplished only by abstracting away broadcast and superposition, which are crucial features of the wireless medium. A natural question, then, is the extent to which such an abstraction impacts performance. Or, put slightly differently: Is it possible to leverage the insights gained over graphical networks to attain optimal performance in *wireless* computation?

My aim in this thesis is to provide at least partial answers to the preceding question. In the following chapters, I examine both network coding and averaging consensus under relatively realistic wireless models. In the case of network coding, I focus on smaller networks and study performance with respect to information-theoretic achievable rates. In the case of averaging consensus, I focus on large networks and study performance with respect to the scaling of resource expenditure in the size of the network. In both cases, a key to near-optimal performance is *cooperative communications*, in which terminals overhear each other's transmissions and make subsequent transmissions to assist in carrying out communications tasks. I propose cooperative strategies that, for certain topologies and under certain assumptions, are nearly optimal in the relevant metrics.

1.2 State of the Art

1.2.1 Network Coding

Network coding was introduced in [26], where it was shown to achieve the multicast capacity of wireline networks. It was later shown that (random) linear network codes are sufficient for multicast [29–31], and although linear codes are provably insufficient for general wireline networks [32] they remain popular due to their simplicity and effectiveness. Network coding has been applied to wireless networks by several means. Two information-theoretic techniques are the quantize-map-and-forward of [18] and the “noisy” network coding of [19], in which relays compress and re-encode the incoming superposition of signals. These approaches generalize the discrete-valued, noiseless combinations of wireline network coding to continuous-valued, noisy combinations over wireless links. For multicast networks, they come to within a constant gap of capacity. More practical approaches are COPE [33], in which a network coding “shim” is added to an otherwise-ordinary 802.11 system, and *analog network coding* [34], in which intermediate nodes amplify and forward the noisy superposition of received signals. Both techniques offer throughput gains over existing networking strategies.

A more direct approach to wireless network coding is to match the superposition of the wireless medium to linear codes. In physical-layer network coding [35], multiple terminals transmit uncoded constellation points. Each receiver obtains a noisy sum of constellation points, which is detected and mapped to the modulo sum of the underlying bits. In compute-and-forward [36, 37], this idea is given a thorough information-theoretic treatment. Each transmitter maps finite-field messages to lattice codewords, noisy linear combinations of which arrive at receivers. Each receiver decodes the incoming signal to an integer combination of lattice codewords, which

then is mapped to a finite-field linear combination of messages. Due to the linear structure of lattices, integer combinations of lattice points can be decoded almost as easily as a single lattice codewords; therefore, linear combinations of messages are often easier to decode than the several messages individually. In a network context, if enough linearly independent combinations are recovered by receivers, the individual messages can be disentangled from the combinations somewhere “downstream” in the network. Similar techniques have been applied to the two-way and multi-way relay channels, in some cases achieving rates within a constant gap of capacity [38–41]. Somewhat surprisingly, compute-and-forward has proven useful in interference channels, helping to establish the approximate capacity of symmetric interference channels [17].

Compute-and-forward as proposed in [36] requires a correspondence between wireless channel gains and the desired integer combinations. If the channels do not produce suitable linear combinations of transmitters’ signals, the receivers cannot easily recover suitable integer combinations of the lattice points. Several solutions to this challenge have been proposed. *Integer-forcing receivers* [42, 43], in which linear receivers are chosen to induce integer-valued equivalent channels, were developed for compute-and-forward over multiple-input multiple-output (MIMO) channels. In [44], a number-theoretic approach was developed to address this problem in the high-SNR regime. Matching real interference alignment [45–48] techniques to linear codes, an encoding strategy was proposed that achieves the full degrees of freedom. However, there are no performance guarantees at moderate SNR.

1.2.2 Averaging Consensus

Averaging consensus is often described as a simple, canonical example of distributed signal processing over sensor networks. A common narrative is that each node mea-

sures the local temperature and wants to compute the average temperature over the sensor field. Such simplicity, however, is deceptive, as consensus lies at the heart of an array of sophisticated problems. It has seen use in load-balancing [49], in distributed optimization [50–52], and in distributed estimation and filtering [53, 54].

Consensus has been studied under various guises, including the early work of Tsitsiklis [50], who examined it in the context of distributed estimation. Recent interest in consensus was sparked by the introduction of *gossip algorithms*, and in particular the randomized gossip of [55]. In gossip, the network is modeled by a graph. Nodes iteratively pair with neighbors, exchange estimates, and average those estimates together, eventually converging on the true average. Gossip is simple, requiring minimal processing and network knowledge, and it is robust, retaining performance even with failing links and changing topology. Randomized gossip, however, has relatively slow convergence on random graphs, requiring on the order of N^2 transmissions in a network of N nodes. Since then, researchers have searched for faster consensus algorithms. In *geographic gossip* [56], nodes pair up with geographically distant nodes, exchanging estimates via multi-hop routing. The extra complexity garners faster convergence; geographic gossip requires on the order of $N^{3/2}$ transmissions. Geographic gossip was further refined by the introduction of *path averaging* [27], in which routing nodes contribute their own estimates “along the way.” Path averaging closes the gap to order optimality, requiring roughly N transmissions, which is the minimum of any consensus algorithm. Recently, *multi-scale gossip*, in which the network is hierarchically partitioned into subnetworks, was proposed in [28].

Several works have addressed wireless aspects of consensus, but to my knowledge no study offers a comprehensive analysis. The broadcast nature of wireless is considered in [57, 58]; in these works a single transmission arrives at multiple receivers simultaneously, but otherwise signals do not interact. However, in these works broadcast

does not significantly improve performance over randomized gossip. The superposition nature of wireless is addressed—and in fact exploited—in [59], in which lattice codes are used to compute sums of estimates “over the air.” In [60] it is observed that network topology can be adjusted via power control, and the optimum power allocation is derived for a few specific networks. In a somewhat similar work [61], the optimum graphical structure for consensus is derived. Finally, the impact of noisy links has been studied. In [62], continuous-valued estimates are corrupted by zero-mean additive noise, and optimal linear consensus strategies are derived. For a similar model, the *bias-variance dilemma* is identified in [63]: running consensus for longer reduces the bias of the resulting estimates, but it increases the variance. Algorithms that resolve the dilemma are presented, but they suffer from slow convergence. In [64,65] *quantized consensus* algorithms are presented that achieve consensus while passing finite-alphabet estimates. In [66] traditional gossip algorithms are augmented with *dithered* quantization and are shown to achieve consensus on the true average in expectation. In [67] the increasing correlation among estimates is exploited to construct a consensus algorithm employing Wyner-Ziv style coding with side information.

1.2.3 Lattice Codes

Since the original work by Forney [68,69], who used them for constructing trellis codes, lattices have been shown to achieve capacity on AWGN channels. De Buda and others [70] showed that a capacity-achieving codebook could be constructed by intersecting a lattice with a “thin” spherical shell. Urbanke and Rimoldi [71] showed that capacity can be achieved by the intersection of a lattice with a spherical region. These approaches assume maximum-likelihood decoding at the receiver which, while simpler than typical-sequence decoding, is still rather complex.

Poltyrev [72] studied lattice coding for the AWGN channel without power constraints. He found that *lattice decoding*, in which the decoder simply bins the received signal according to the lattice's Voronoi regions, is asymptotically efficient. Lattice decoding, while much simpler than ML decoding, is suboptimal in terms of absolute probability of error for the ordinary power-constrained AWGN channel, and for some time it was not clear whether lattice decoding was sufficient to achieve capacity. Loeliger showed the existence of lattice codes that achieve $\frac{1}{2} \log_2(\text{SNR})$ under lattice decoding [73], which he conjectured was the highest rate possible. Finally, Erez and Zamir [74] showed, by means of the modulo-lattice transform and random dithers, that lattice codes achieve capacity with lattice decoding at the receiver.

In addition to offering structure, achieving capacity, and reducing complexity, lattice codes are desirable because they are the Euclidean-space analogue to linear codes. Inspired by the existence of capacity-achieving lattice codes, low-density *lattice* codes were proposed in [75]. Much as LDPC codes approach the capacity of the binary symmetric channel, these codes approach the capacity of the AWGN channel. In addition to wireless network coding problems, lattice codes have seen use in a variety of information-theoretic problems, including source coding [76–78], physical-layer security [79–81], and relay networks [82–85].

1.3 Contributions

1.3.1 Network Coding

On the topic of network coding, I offer two main contributions. In Chapter 3, I study a single-layer wireless network of L transmitters and M receivers, in which the L transmitters may overhear each other's signals and cooperate and in which the L receivers intend to compute linearly independent functions of the transmitters'

finite-field messages. This model is based on the observation that, if transmitters were able to encode their messages jointly, compute-and-forward would reduce to a multiple-antenna broadcast channel, the capacity of which is known [86]. While *perfect* cooperation is infeasible, users can cooperate *partially* by exploiting another consequence of the broadcast nature: transmitters can overhear each other's signals and jointly encode portions of their messages.

For this scenario, I develop a cooperative strategy for compute-and-forward. I construct a lattice-coding instantiation of decode-and-forward block Markov encoding by decomposing a codebook of nested lattices into two linearly independent, lower-rate constituent codes. Transmitters broadcast lattice codewords, after which they decode the codewords of other transmitters. They then cooperatively transmit “resolution” information corresponding to the linear combinations desired at the receivers. Receivers employ a variant of sliding-window decoding tailored to the proposed lattice decomposition. They decode the resolution information and subtract it from the original signal; they then need only to decode the remaining low-rate component of the desired sum. This strategy allows an improvement in computation rate due to two factors. First, since cooperating transmitters decode others' messages, they can jointly encode portions of the linear combinations directly, relaxing the need for receivers to recover the messages from separately-encoded signals. Second, the jointly encoded signals combine coherently at receivers, resulting in a beamforming gain.

In addition to proving that cooperation improves the achievable rate, I conduct a diversity-multiplexing tradeoff (DMT) analysis. For the special case of a two-transmitter, single receiver system, the proposed scheme achieves the optimum DMT, which is identical to a single-user 2×1 multiple-antenna, single-output (MISO) channel. For the more general $L \times 1$ system, cooperation affords a DMT gain over non-cooperation, but it does not achieve the associated MISO outer bound. Unfortunately,

beyond a single receiver it is difficult to say much about the DMT performance of the proposed scheme.

In Chapter 4, I study a two-transmitter, two-receiver network in which a dedicated relay assists transmitters in communicating finite-field combinations to the receiver. The relay operates according to one of two relay modalities. In the standard modality, the relay transmits a function of signals received in previous symbol times. For this scenario I construct a lattice-based compress-and-forward scheme. It achieves higher rates than non-cooperative techniques, but it appears to give no improvement in the high-SNR regime. In the *instantaneous* modality, the relay's signal may be a function of signals received in the *current* symbol time, meaning that there is no delay between reception and transmission. For this scenario I construct a simple lattice-based amplify-and-forward scheme. Not only does amplify-and-forward achieve the optimal degrees-of-freedom (or multiplexing gain), but it also achieves rates within a constant gap of capacity for a non-trivial set of channel gains. Thus, by contrast with the non-cooperative case, relay cooperation permits near-optimal performance at both moderate SNR and in the limit of high SNR.

1.3.2 Averaging Consensus

In Chapter 5 I present a comprehensive analysis of the resource demands of consensus over wireless networks. First I define a realistic but tractable framework in which to study the resource demands of consensus. It consists of a path-loss dominated propagation model in which connectivity is determined by a signal-to-noise ratio (SNR) threshold. Initially I suppose connected links are perfect and have infinite capacity. For this case I propose three resource metrics appropriate to the wireless medium: the total energy expended in order to achieve consensus, the total time elapsed, and the time-bandwidth product consumed. Under this model, I derive

lower bounds on the required resources and characterize the resource requirements of several existing consensus strategies. Path averaging, which is optimal over graphical networks in the number of required transmissions, turns out to be nearly order optimal in the energy expended. However, it remains strictly suboptimal in elapsed time and consumed time-bandwidth product.

Next, I propose a new consensus algorithm, termed *hierarchical averaging*, designed specifically for wireless networks. Instead of communicating with neighbors over a graph, nodes broadcast estimates to geographically-defined clusters of nodes. These clusters expand as consensus proceeds, which is enabled by adjusting nodes' transmit power. Much like the hierarchical cooperation of [87] and the multiscale gossip of [28], small clusters cooperatively broadcast information to larger clusters, continuing until consensus is achieved. Depending on the particulars of the channel model, hierarchical averaging is nearly order optimal in all three metrics simultaneously. When channel phases are fixed and identical, hierarchical averaging is order optimal, up to an arbitrarily small gap in the exponents, for path-loss exponents $2 \leq \alpha < 4$. In the more realistic case in which phases are random and independent, however, hierarchical averaging is no longer order optimal in transmit energy when $\alpha > 2$, although it remains order optimal with respect to the other two metrics.

Finally, in Section 5.5 I incorporate quantization into the proposed model. Since practical wireless links suffer from noise, achievable rates are finite and estimates must be quantized prior to transmission. This introduces a tradeoff: Expending more energy increases the rate of the links, thereby reducing the quantization error inherent to each transmission and therefore the estimation error accrued during consensus. Therefore, in addition to the resource metrics of energy, time, and bandwidth, I consider a fourth performance metric: mean-square error of the consensus estimates. Again I characterize existing consensus techniques. I also apply quantization to hier-

archical averaging, showing that it permits an efficient tradeoff between energy and estimation error.

1.4 Notation

Let \mathbb{R} and \mathbb{Z} be the real and integer fields, respectively. Let \mathbb{F}_p be the finite (Galois) field of prime characteristic p , and let \oplus and \odot denote addition and (matrix) multiplication, respectively, modulo p ; however, I will occasionally treat the *result* of modular arithmetic as a member of the reals according to context. Bold uppercase letters (e.g. \mathbf{A}) refer to matrices and bold lowercase letters (e.g. \mathbf{x}) to refer to column vectors. For $n \times m$ matrix \mathbf{A} , \mathbf{a}_i refers to the i th column of \mathbf{A} , i.e. $\mathbf{A} = [\mathbf{a}_1 \cdots \mathbf{a}_m]$. I denote subvectors of a vector using $\mathbf{x}[a : b] = (x_a, x_{a+1}, \dots, x_b)^T$, where $(\cdot)^T$ denotes the usual transpose. I use $\|\cdot\|$ for the Euclidean norm. Let \circ denote the element-wise or Hadamard product. Let \mathbb{F}_p denote the finite field of prime characteristic p . Let $[x]^+ = \max\{x, 0\}$ denote the positive part of x . Let

$$C_{\text{mac}}(\mathbf{h}, P, \sigma^2) = \min_{\mathcal{B} \subset \{1, \dots, I\}} \frac{1}{2|\mathcal{B}|} \log \left(1 + \frac{P \sum_{i \in \mathcal{B}} h_i^2}{\sigma^2} \right)$$

denote the symmetric-rate capacity of the I -user Gaussian multiple-access channel having channel gains \mathbf{h} and noise variance σ^2 . In discussing asymptotics, I use the Landau notation: $f(n) = O(g(n))$ implies $f(n) \leq kg(n)$, $f(n) = \Omega(g(n))$ implies $f(n) \geq kg(n)$, and $f(n) = \Theta(g(n))$ implies $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$, all for arbitrary constant k and sufficiently large n .

Lattices and Lattice Codes

Lattices are mathematical structures with a simple definition but far-reaching consequences. In addition to their inherent mathematical significance, lattices have connections to cryptography, crystallography, solid-state physics, and, perhaps surprisingly, information and coding theory. In this chapter I review a few basic facts about lattices and their use as channel codes. I also present a decomposition of a nested lattice codebook into independent subspaces for the purpose of block Markov encoding. As a simple demonstration of the idea, I show how the decomposition can be used to achieve the decode-and-forward rate of the three-terminal AWGN relay channel proven in [15].

2.1 Lattices

Formally, a lattice Λ is a discrete additive subgroup of \mathbb{R}^n , which implies that for any $\lambda_1, \lambda_2 \in \Lambda$, both $\lambda_1 + \lambda_2 \in \Lambda$ and $\lambda_1 - \lambda_2 \in \Lambda$. Equivalently, a lattice Λ is the set of all integer combinations of a set of basis vectors, which need not be unique. Collecting such a basis into a matrix, form the *generator matrix* of Λ , denoted by $\mathbf{G} \in \mathbb{R}^{n \times n}$.

One can therefore express Λ as:

$$\Lambda = \mathbf{G}\mathbb{Z}^n. \quad (2.1)$$

Define Q_Λ to be the *lattice quantizer*, which maps any point $\mathbf{x} \in \mathbb{R}^n$ to the nearest point in Λ :

$$Q_\Lambda(\mathbf{x}) = \arg \min_{\lambda \in \Lambda} \|\mathbf{x} - \lambda\|. \quad (2.2)$$

The lattice Λ induces a partition of \mathbb{R}^n into the *Voronoi regions* $\mathcal{V}(\lambda)$ of each lattice point $\lambda \in \Lambda$:

$$\mathcal{V}(\lambda) = \{\mathbf{x} \in \mathbb{R}^n : Q_\Lambda(\mathbf{x}) = \lambda\}, \quad (2.3)$$

where ties are broken arbitrarily. In other words, the Voronoi region of $\lambda \in \Lambda$ is the set of points that are closer to λ than to any other lattice point.

Let $\mathcal{V} = \mathcal{V}(0)$ be the *fundamental Voronoi region* of Λ . The mod operation with respect to Λ returns the quantization error

$$\mathbf{x} \bmod \Lambda = \mathbf{x} - Q_\Lambda(\mathbf{x}), \quad (2.4)$$

which is always a member of \mathcal{V} . The mod operation allows one to draw an analogy with modulo arithmetic over a finite field. Just as modulo arithmetic ensures that the result remains a member of the finite field, performing arithmetic modulo Λ “wraps” the result within \mathcal{V} . It is straightforward to prove that the mod operation obeys the associativity property:

$$[[\mathbf{x}] \bmod \Lambda + \mathbf{y}] \bmod \Lambda = [\mathbf{x} + \mathbf{y}] \bmod \Lambda. \quad (2.5)$$

The *second moment* $\sigma^2(\Lambda)$ quantifies the average power of a random variable

uniformly distributed across \mathcal{V} :

$$\sigma^2(\Lambda) = \frac{1}{n\text{Vol}(\mathcal{V})} \int_{\mathcal{V}} \|\mathbf{x}\|^2 d\mathbf{x}, \quad (2.6)$$

where $\text{Vol}(A)$ is the volume of a set $A \subset \mathbb{R}^n$. The *normalized second moment* is defined as:

$$G(\Lambda) = \frac{\sigma^2(\Lambda)}{\text{Vol}(\mathcal{V})^{\frac{2}{n}}}. \quad (2.7)$$

The normalized second moment provides a measure of the efficiency of Λ as a shaping region. The normalized second moment of a sphere in \mathbb{R}^n is $1/2\pi e$, which is the minimum value for any lattice. The closer \mathcal{V} is to being spherical, the smaller, and closer to $1/2\pi e$, $G(\Lambda)$ is.

The *covering radius* $r_{\text{cov}}(\Lambda)$ is the radius of the smallest sphere that covers \mathcal{V} :

$$r_{\text{cov}}(\Lambda) = \inf_r \{r > 0 | \mathcal{V} \subset r\mathcal{B}_n\}, \quad (2.8)$$

where \mathcal{B}_n is the unit sphere in \mathbb{R}^n . The *effective radius* $r_{\text{eff}}(\Lambda)$ be the radius of a sphere with the same volume as \mathcal{V} :

$$r_{\text{eff}}(\Lambda) = \left(\frac{\text{Vol}(\mathcal{V})}{\text{Vol}(\mathcal{B}_n)} \right)^{\frac{1}{n}}. \quad (2.9)$$

Note that $r_{\text{cov}}(\Lambda) \geq r_{\text{eff}}(\Lambda)$.

In order to construct lattice codebooks suitable for proving information-theoretic results, *sequences* of lattices that asymptotically satisfy several desirable properties are required. For example, a sequence of lattices $\{\Lambda^{(n)}\}$, $\Lambda^{(n)} \in \mathbb{R}^n$, is said to be *good for covering* or *Rogers good* [88] provided the covering radius approaches the effective radius:

$$\lim_{n \rightarrow \infty} \frac{r_{\text{cov}}(\Lambda^{(n)})}{r_{\text{eff}}(\Lambda^{(n)})} = 1.$$

Similarly, a sequence of lattices is said to be *good for quantization* provided

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

Finally, let $\mathbf{z} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ be a Gaussian random vector. Define the *volume-to-noise ratio* $\mu(\Lambda, P_e)$ as

$$\mu(\Lambda, P_e) = \frac{(\text{Vol}(\mathcal{V}))^{\frac{2}{n}}}{\sigma^2},$$

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

$$\lim_{n \rightarrow \infty} \mu(\Lambda^{(n)}, P_e) = 2\pi e, \text{ for } 0 < P_e < 1,$$

and if, for fixed $\mu(\Lambda^{(n)}, P_e)$ greater than $2\pi e$, P_e goes to zero exponentially in n . The existence of such sequences was proven by Poltyrev in [72]. Furthermore, Erez et al. proved that there exist sequences of lattices that are simultaneously good for covering, quantization, and AWGN coding [89].

2.2 Lattice Codes

In [74], Erez and Zamir showed that codes constructed from nested lattices can achieve the capacity of the AWGN channel. Since it constitutes the backbone of techniques used throughout this thesis, I briefly review their construction.

Let $\Lambda_s^{(n)}$ be a sequence of *shaping lattices* that are good for covering and AWGN coding and satisfy $\sigma^2(\Lambda_s^{(n)}) = 1$, and let $\mathbf{G}_s^{(n)}$ denote a generator matrix of each lattice in the sequence. Then, following [90], adapt Construction A [73] to construct a sequence of coding lattices $\Lambda_c^{(n)} \supset \Lambda_s^{(n)}$. The construction goes as follows:

1. For each n , choose an integer k and a prime p . Draw a $n \times k$ matrix $\mathbf{F}_c^{(n)} \in \mathbb{F}_p^{n \times k}$

randomly and uniformly.

2. Construct the linear codebook over \mathbb{F}_p defined by $\mathbf{F}_c^{(n)}$:

$$\hat{\mathcal{C}}^{(n)} = \mathbf{F}_c^{(n)} \mathbb{F}_p^k$$

3. “Lift” the codebook $\hat{\mathcal{C}}^{(n)}$ to \mathbb{R}^n by defining the lattice

$$\hat{\Lambda}_c^{(n)} = p^{-1} \hat{\mathcal{C}}^{(n)} + \mathbb{Z}^n.$$

4. Finally, rotate $\hat{\Lambda}_c^{(n)}$ so that it is nested inside $\Lambda_s^{(n)}$:

$$\Lambda_c^{(n)} = \mathbf{G}_s^{(n)} \hat{\Lambda}_c^{(n)}.$$

The lattice codebook is the intersection of the coding lattice and the fundamental Voronoi region of the shaping lattice:

$$\mathcal{C}^{(n)} = \Lambda_c^{(n)} \cap \mathcal{V}_s^{(n)}.$$

The rate of this codebook is

$$R = \frac{1}{n} \log_2 |\mathcal{C}^{(n)}| = \frac{k \log_2(p)}{n}.$$

It is shown in [36] that choosing p such that $n/p \rightarrow 0$ as $n \rightarrow \infty$ guarantees that the sequence of coding lattices $\Lambda_c^{(n)}$ is good for AWGN coding. For any desired rate $R > 0$, one can construct an appropriate sequence of codebooks by choosing $p = n \log_2(n)$ and $k = \lfloor \frac{nR}{\log_2(p)} \rfloor$.

Intuitively, the preceding codebook construction allows one to take a random linear

block code over \mathbb{F}_p and create a corresponding linear code over Euclidean space. If the underlying linear code achieves capacity, as does the ensemble of random linear codes, so too does the resulting lattice codebook. One can use any linear code in place of the random code chosen above, including low-density parity-check codes or other low-complexity linear block codes. The cost in achievable rate is only the gap to capacity of the linear code chosen.

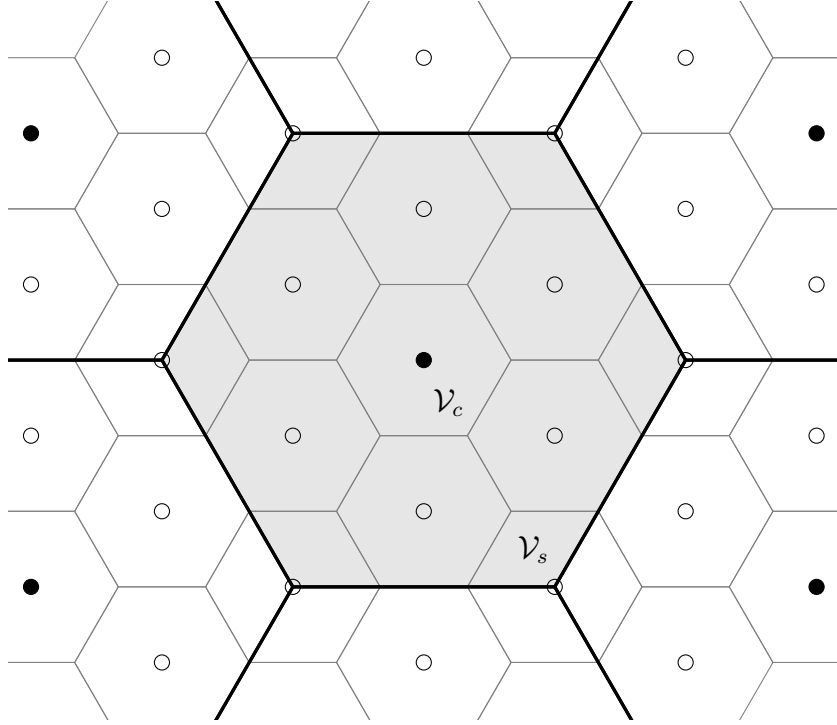


Figure 2.1: Nested lattice codes. White dots are elements of the coding lattice, and black dots are elements of the shaping lattice. Each lattice point inside the shaded Voronoi region \mathcal{V}_s is a member of the codebook.

2.3 The AWGN Channel

Next I review the lattice and decoding scheme proposed in [74] to achieve the capacity of the AWGN channel. Along the way I give a bit of intuition—but no proofs!—as to why the scheme achieves capacity.

The point-to-point AWGN channel is described by the following input-output relationship

$$\mathbf{y} = \mathbf{x} + \mathbf{z},$$

where $\mathbf{x} \in \mathbb{R}^n$ is the transmitted signal, $\mathbf{y} \in \mathbb{R}^n$ the received signal, and $\mathbf{z} \in \mathbb{R}^n$ is additive white Gaussian noise having variance N . The transmitted signal is constrained to have average power P :

$$\frac{1}{n} \|\mathbf{x}\|^2 \leq P. \quad (2.10)$$

The transmitter employs the nested lattice codebook $\mathcal{C}^{(n)}$ as described above. To send the codeword $\lambda \in \mathcal{C}^{(n)}$, it transmits the signal

$$\mathbf{x} = \sqrt{P}[\lambda + \mathbf{t}] \bmod \Lambda_s^{(n)},$$

where \mathbf{t} is a random dither¹ uniformly distributed across $\mathcal{V}_s^{(n)}$ and known to both source and destination. Since \mathbf{t} is uniformly distributed over $\mathcal{V}_s^{(n)}$, and since $\Lambda_s^{(n)}$ has unit normalized second moment, the dithered codeword has unit average power. Thus \mathbf{x} has average power P .

The destination receives $\mathbf{y} = \mathbf{x} + \mathbf{z}$. To decode λ , the destination first scales the received signal, subtracts the dither, and takes the result modulo the shaping lattice to form

$$\begin{aligned} \mathbf{y}' &= [\beta \mathbf{y} - \mathbf{t}] \bmod \Lambda_s^{(n)} \\ &= [\mathbf{x} - \mathbf{t} + \beta \mathbf{z} + (\beta - 1)\mathbf{x}] \bmod \Lambda_s^{(n)} \\ &= [\lambda + \beta \mathbf{z} + (\beta - 1)\mathbf{x}] \bmod \Lambda_s^{(n)}. \end{aligned}$$

The factor β allows the receiver to tradeoff between the power in the noise term $\beta \mathbf{z}$

¹For further discussion of the need for dithers, see [91].

and the power in the “self-noise” term $(\beta - 1)\mathbf{x}$. A simple choice is $\beta = 1$, which is equivalent to having signaled using no dither at all. In this case, the receiver can correctly decode λ so long as the volume-to-noise ratio is smaller than $2\pi e$, which corresponds to rates smaller than $\frac{1}{2} \log_2(P/N)$. However, one can do better. Choosing β as the MMSE coefficient

$$\beta = \frac{P}{P + N}$$

optimizes the effective SNR at the receiver. The receiver estimates λ by lattice quantization

$$\hat{\lambda} = Q_{\Lambda_c^{(n)}}(\mathbf{y}').$$

It is shown in [74, Theorem 3] that, averaging over the dither \mathbf{t} , reliable decoding is possible if $R < \frac{1}{2} \log_2(1 + \frac{P}{N})$.

2.4 Lattice subspaces

For the lattice compute-and-forward proposed in [36], an important fact is that there exists a mapping from finite-field messages to lattice codewords that preserves linearity. That is, the mapping sends finite-field linear combinations of messages to integer sums of lattice points modulo the shaping lattice. Formally, this implies that there is an isomorphism between the additive group of field elements and the group of lattice codewords modulo the shaping lattice, as expressed in the following lemma.

Lemma 2.1. *There exists an isomorphism $\phi : \mathbb{F}_p^k \rightarrow \mathcal{C}^{(n)}$, namely*

$$\phi(\mathbf{w}) = [\mathbf{G}_s^{(n)} p^{-1} \mathbf{F}_c^{(n)} \mathbf{w}] \bmod \Lambda_s^{(n)}. \quad (2.11)$$

Proof. To establish the claim, it is necessary to show that ϕ is a bijection and that it respects the group operation; that is, $\phi(\mathbf{w}_1 \oplus \mathbf{w}_2) = [\phi(\mathbf{w}_1) + \phi(\mathbf{w}_2)] \bmod \Lambda_s^{(n)}$ for

any $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{F}_p^k$. That ϕ is a bijection was shown in [36, Lemma 5]. To see that ϕ respects the group operation, direct computation suffices:

$$\phi(\mathbf{w}_1 \oplus \mathbf{w}_2) = [\mathbf{G}_s^{(n)} p^{-1} \mathbf{F}_c^{(n)}(\mathbf{w}_1 \oplus \mathbf{w}_2)] \bmod \Lambda_s^{(n)} \quad (2.12)$$

$$= [\mathbf{G}_s^{(n)} p^{-1} (\mathbf{F}_c^{(n)}(\mathbf{w}_1 + \mathbf{w}_2) + p\mathbf{i})] \bmod \Lambda_s^{(n)}, \quad (2.13)$$

where $\mathbf{i} \in \mathbb{Z}^n$ is a vector of integers corresponding to the discrepancy between real-valued and modulo- p arithmetic. Further manipulations yield

$$\phi(\mathbf{w}_1 \oplus \mathbf{w}_2) = [\mathbf{G}_s^{(n)} p^{-1} \mathbf{F}_c^{(n)}(\mathbf{w}_1 + \mathbf{w}_2) + \mathbf{G}_s^{(n)} \mathbf{i}] \bmod \Lambda_s^{(n)} \quad (2.14)$$

$$= [\mathbf{G}_s^{(n)} p^{-1} \mathbf{F}_c^{(n)}(\mathbf{w}_1 + \mathbf{w}_2)] \bmod \Lambda_s^{(n)} \quad (2.15)$$

$$= [\phi(\mathbf{w}_1) + \phi(\mathbf{w}_2)] \bmod \Lambda_s^{(n)}. \quad (2.16)$$

where the last equality is due to the fact that $\mathbf{G}_s^{(n)} \mathbf{i} \in \Lambda_s^{(n)}$ and that adding a member of $\Lambda_s^{(n)}$ does not change the result of the arithmetic modulo $\Lambda_s^{(n)}$. \square

In the cooperative strategy proposed in Chapter 3, I tailor block Markov encoding to lattice codes. A key ingredient of this approach is the decomposition of the lattice codebook into subspaces. Let $k_r \leq k$, and let $\mathbf{F}_r^{(n)} \in \mathbb{F}_p^{n \times k_r}$ denote the matrix composed of the first k_r columns of $\mathbf{F}_c^{(n)}$. Similarly, let $k_v = k - k_r$, and let $\mathbf{F}_v^{(n)} \in \mathbb{F}_p^{n \times k_v}$ denote the matrix of the remaining k_v columns. Then define the *resolution lattice* Λ_r and the *vestigial*¹ *lattice* Λ_v as

$$\Lambda_r^{(n)} = \mathbf{G}_s^{(n)} (p^{-1} \mathbf{F}_r^{(n)} \mathbb{F}_p^{k_r} + \mathbb{Z}^n)$$

$$\Lambda_v^{(n)} = \mathbf{G}_s^{(n)} (p^{-1} \mathbf{F}_v^{(n)} \mathbb{F}_p^{k_v} + \mathbb{Z}^n).$$

¹This terminology is intended to convey the fact that this lattice component encodes the “residual” or “leftover” information bits. I use the less-common synonym in order to minimize notational confusion.

Since these sequences of lattices are special cases of the lattice construction from the previous subsection, each sequence is individually good for AWGN coding. By construction $\Lambda_c^{(n)} = \Lambda_r^{(n)} + \Lambda_v^{(n)}$ and $\Lambda_s^{(n)} \subset \Lambda_r^{(n)}, \Lambda_v^{(n)} \subset \Lambda_c^{(n)}$. Define the resolution and vestigial codebooks

$$\begin{aligned}\mathcal{C}_r^{(n)} &= \Lambda_r^{(n)} \cap \mathcal{V}_{\Lambda_s^{(n)}} \\ \mathcal{C}_v^{(n)} &= \Lambda_v^{(n)} \cap \mathcal{V}_{\Lambda_s^{(n)}},\end{aligned}$$

having rates

$$\begin{aligned}R_r &= \frac{k_r}{n} \log_2 p \\ R_v &= \frac{k_v}{n} \log_2 p.\end{aligned}$$

By construction $R_r + R_v = R_c$. Furthermore, for any $0 \leq R_r \leq R$, one can choose $k_r = \lfloor \frac{nR_r}{\log_2(p)} \rfloor$ to achieve the desired resolution codebook rate. For any message $\mathbf{w} \in \mathbb{F}_p^k$, define the *projection* onto the resolution and vestigial codebook as follows:

$$\begin{aligned}\phi_r(\mathbf{w}) &= [\mathbf{G}_s p^{-1} \mathbf{F}_r \mathbf{w} [1 : k_r]] \bmod \Lambda_s \\ \phi_v(\mathbf{w}) &= [\mathbf{G}_s p^{-1} \mathbf{F}_v \mathbf{w} [k_r + 1 : k]] \bmod \Lambda_s.\end{aligned}$$

Using these projections, the lattice codebook can be decomposed linearly, as depicted in Figure 2.2 and expressed in the following lemma.

Lemma 2.2. *For any $\mathbf{w} \in \mathbb{F}_p^k$,*

$$\phi(\mathbf{w}) = [\phi_r(\mathbf{w}) + \phi_v(\mathbf{w})] \bmod \Lambda_s^{(n)}, \quad (2.17)$$

Proof. The result follows from Lemma 2.1. By definition,

$$\mathbf{w} = (\mathbf{w}^T[1 : k_r] \mathbf{0}_{k_v}^T)^T \oplus (\mathbf{0}_{k_r}^T \mathbf{w}^T[k_r + 1 : k])^T,$$

so

$$\begin{aligned} \phi(\mathbf{w}) &= \phi((\mathbf{w}^T[1 : k_r], \mathbf{0}_{k_v}^T)^T \oplus (\mathbf{0}_{k_r}^T, \mathbf{w}^T[k_r + 1 : k])^T) \\ &= [\phi((\mathbf{w}^T[1 : k_r], \mathbf{0}_{k_v}^T)^T) + \phi((\mathbf{0}_{k_r}^T, \mathbf{w}^T[k_r + 1 : k])^T)] \bmod \Lambda_s^{(n)} \\ &= [\phi_r(\mathbf{w}) + \phi_v(\mathbf{w})] \bmod \Lambda_s^{(n)}, \end{aligned}$$

where the last equality follows from the definition of $\mathbf{F}_r^{(n)}$ and $\mathbf{F}_v^{(n)}$; zeroing out the unwanted portions of \mathbf{w} is equivalent to discarding the associated columns of $\mathbf{F}^{(n)}$. \square

The codeword $\phi(\mathbf{w}) \in \mathcal{C}^{(n)}$ is therefore the sum of two linearly independent lattice points: $\phi_r(\mathbf{w})$, termed the *resolution information* and which encodes the first $k_r \log_2 p$ bits of the message, and $\phi_v(\mathbf{w})$, termed the *vestigial information* and which encodes the remaining $k_v \log_2 p$ bits. Furthermore, the decomposition is linear in the sense that the decomposition of sums of lattice points is the same as the sum of decompositions, as shown in the following lemma.

Lemma 2.3. *Let \mathbf{w}_1 and \mathbf{w}_2 be messages in \mathbb{F}_p^k , and let $\mathbf{w} = \mathbf{w}_1 \oplus \mathbf{w}_2$. Then*

$$\phi_r(\mathbf{w}) = [\phi_r(\mathbf{w}_1) + \phi_r(\mathbf{w}_2)] \bmod \Lambda_s^{(n)}, \quad (2.18)$$

and

$$\phi_v(\mathbf{w}) = [\phi_v(\mathbf{w}_1) + \phi_v(\mathbf{w}_2)] \bmod \Lambda_s^{(n)}. \quad (2.19)$$

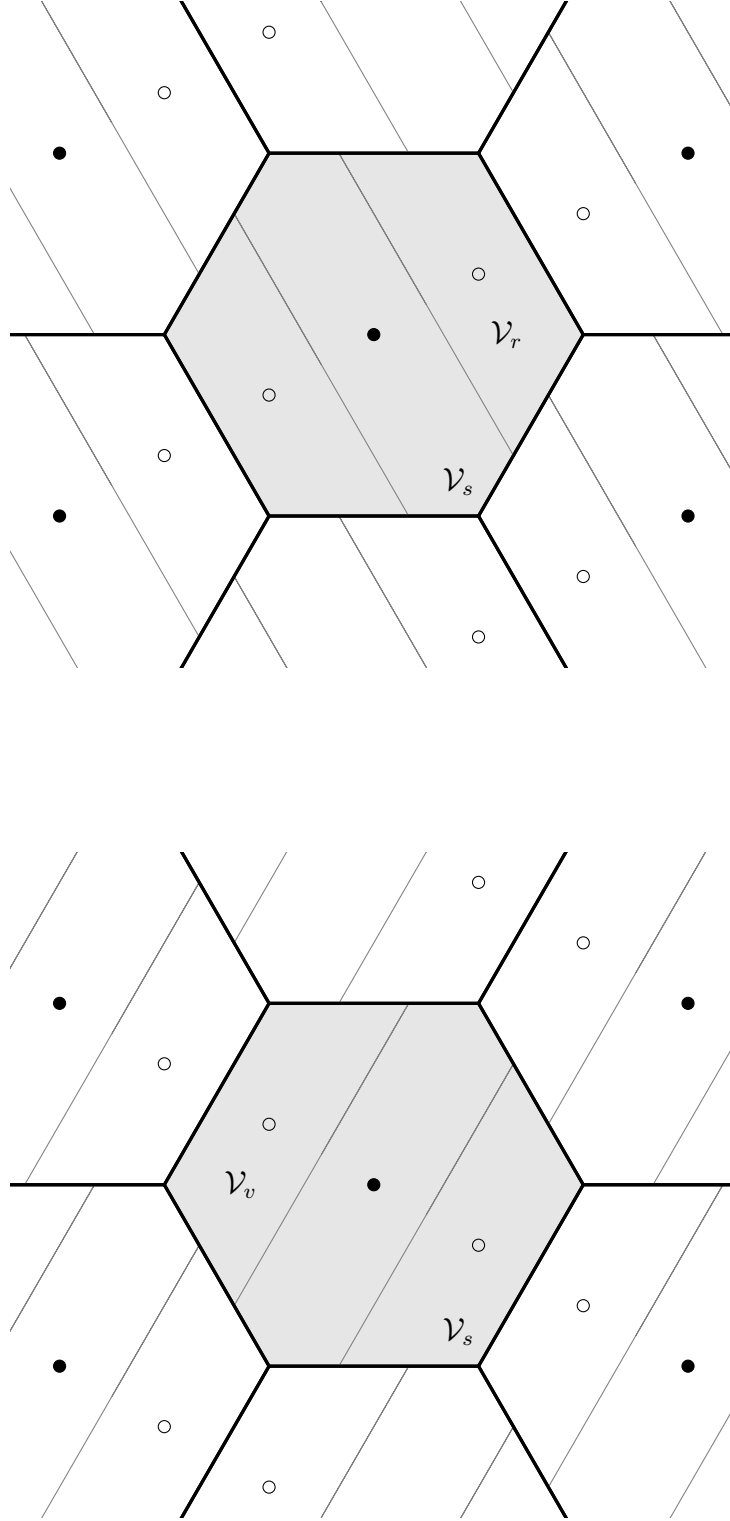


Figure 2.2: Lattice subspace decomposition. Each lattice codeword in $\mathcal{C}^{(n)}$ is the sum of a point in $\mathcal{C}_r^{(n)}$ (upper) and a point in $\mathcal{C}_v^{(n)}$ (lower). The shaded region \mathcal{V}_s defines the codebook, whereas the strip-shaped Voronoi regions \mathcal{V}_r and \mathcal{V}_v define the decoding regions of the resolution and vestigial codebooks, respectively.

Proof. This follows directly from the fact that ϕ is an isomorphism:

$$\phi_r(\mathbf{w}) = \phi_r(\mathbf{w}_1 \oplus \mathbf{w}_2) \quad (2.20)$$

$$= \phi(\mathbf{w}_1[1 : k_r] \oplus \mathbf{w}_2[1 : k_r]) \quad (2.21)$$

$$= [\phi(\mathbf{w}_1[1 : k_r]) + \phi(\mathbf{w}_2[1 : k_r])] \bmod \Lambda_s^{(n)} \quad (2.22)$$

$$= [\phi_r(\mathbf{w}_1) + \phi_r(\mathbf{w}_2)] \bmod \Lambda_s^{(n)}. \quad (2.23)$$

A similar argument holds for ϕ_v . □

The preceding decomposition permits a lattice-coding instantiation of block Markov encoding, as illustrated in the next section.

2.5 A Diversion: Lattice Coding over the Relay Channel

To illustrate how the preceding lattice subspaces can be used for block Markov encoding, I briefly examine the three-terminal relay channel. I present a lattice-based encoding scheme that achieves the decode-and-forward rate derived by Cover and El Gamal in [15].

The relay channel, depicted in Figure 2.3, is a fundamental unit of cooperative communications. Three terminals comprise the relay channel: a source with a message to transmit, a destination intending to receive the source's message, and a relay willing to facilitate communication with the destination. In the AWGN relay channel, the source and relay transmit signals \mathbf{x}_s and \mathbf{x}_r , and the relay and destination receive

signals \mathbf{y}_r and \mathbf{y}_d , as follows:

$$\mathbf{y}_r = \mathbf{x}_s + \mathbf{z}_r$$

$$\mathbf{y}_d = \mathbf{x}_s + \mathbf{x}_r + \mathbf{z}_d,$$

where \mathbf{z}_r and \mathbf{z}_d are AWGN having average power N_r and N_d , respectively. As in the point-to-point AWGN channel, the transmit signals must obey power constraints:

$$\frac{1}{n} \|\mathbf{x}_s\|^2 \leq P_s, \quad \frac{1}{n} \|\mathbf{x}_r\|^2 \leq P_r.$$

The relay is capable of *full-duplex* operation, meaning that it can transmit and receive simultaneously. In general, the capacity of the relay channel is unknown, but a popular lower bound is the achievable rate of the decode-and-forward scheme, in which the relay decodes the source's message entirely and retransmits (perhaps a portion of) it to the destination. Decode-and-forward achieves rates satisfying

$$R < \max_{0 \leq \beta \leq 1} \min \left\{ \frac{1}{2} \log_2 \left(1 + \frac{\bar{\beta} P_s}{N_r} \right), \frac{1}{2} \log_2 \left(1 + \frac{P_s + P_r + 2\sqrt{\beta P_s P_r}}{N_d} \right) \right\}, \quad (2.24)$$

where β denotes the split of the source's power between sending fresh information and resolution information to the destination, and $\bar{\beta} = 1 - \beta$. The decode-and-forward rates are achieved using block Markov encoding, in which the source sends a message during one block and in the next block collaborates with the relay to send resolution

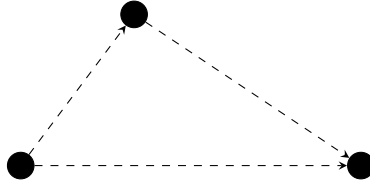


Figure 2.3: The relay channel.

information to the destination. Here I adapt block Markov encoding to lattice codes.

Encoding: To achieve (2.24), the terminals employ main, resolution, and vestigial codebooks \mathcal{C} , \mathcal{C}_r , and \mathcal{C}_v , respectively,¹ constructed from lattices as described in the previous section.

The encoding procedure, shown in Table 4.1, is based on block Markov encoding. As in [15], the source transmit B messages over $B + 1$ blocks of n channel uses each. Let $\mathbf{w}[b] \in \mathbb{F}_p^k$ be the message transmitted by the source in block b . Then, let $\lambda[b] = \phi(\mathbf{w}[b])$ be the associated lattice point, and let $\lambda_r[b] = \phi_r(\mathbf{w}[b])$ and $\lambda_v[b] = \phi_v(\mathbf{w}[b])$ be the resolution and vestigial components of $\lambda[b]$, respectively.

Fix $0 \leq \beta \leq 1$. At block 1, the source transmits

$$\mathbf{x}_s[1] = \sqrt{\beta P_s}([\lambda[1] + \mathbf{t}[1]] \bmod \Lambda_s),$$

where $\mathbf{t}[1]$ is a random dither, uniformly distributed over \mathcal{V}_s and known to source, relay, and destination. Suppose the relay, which is silent during block 1, correctly decodes $\lambda(1)$.

Table 2.1: Lattice block Markov encoding

	$b = 1$	$b = 2$	\cdots	$b = B + 1$
Transmitter	$\lambda[1]$	$\lambda[2] + \lambda_r[1]$	\cdots	$\lambda_r[B]$
Relay	-	$\lambda_r[1]$	\cdots	$\lambda_r[B]$

During block 2 the source and relay cooperatively send resolution information—in the form of $\lambda_r[1]$ —to the destination. The relay transmits

$$\mathbf{x}_r[2] = \sqrt{P_r}([\lambda_r[1]] + \mathbf{s}[1]) \bmod \Lambda_s),$$

where $\mathbf{s}[1]$ is also a random dither, uniformly distributed over \mathcal{V}_s and known to source,

¹From here on out, I omit the superscripts on lattices and lattice codebooks.

relay, and destination. In addition to sending the resolution information, the source needs to transmit the codeword for block 2, so it sends

$$\mathbf{x}_s[2] = \sqrt{\bar{\beta}P_s}([\lambda[2] + \mathbf{t}[2]] \bmod \Lambda_s) + \sqrt{\beta P_s}([\lambda_r[1] + \mathbf{s}[1]] \bmod \Lambda_s).$$

Encoding continues in this manner: at block b , the source sends

$$\mathbf{x}_s[b] = \sqrt{\bar{\beta}P_s}([\lambda[b] + \mathbf{t}[b]] \bmod \Lambda_s) + \sqrt{\beta P_s}([\lambda_r[b-1] + \mathbf{s}[b-1]] \bmod \Lambda_s),$$

while the relay sends

$$\mathbf{x}_r[b] = \sqrt{P_r}([\lambda_r[b-1] + \mathbf{s}[b-1]] \bmod \Lambda_s),$$

until block $B + 1$. At the final block, there is no fresh information for the source to send, so it simply transmits

$$\mathbf{x}_s[B+1] = \sqrt{\beta P_s}([\lambda_r[B] + \mathbf{s}[B]] \bmod \Lambda_s).$$

The source has transmitted nRB bits over $n(B+1)$ channel uses. The overall rate can therefore be made as close as possible to R by choosing large enough B .

Decoding: Decoding proceeds in three stages. First, the relay decodes the entire message $\lambda[b]$; then, the destination decodes the resolution information $\lambda_r[b]$; finally, it decodes vestigial component $\lambda_v[b]$. In each case, the receiver takes the incoming signal $\mathbf{y}[b]$, subtracts any known interference, and scales down by the transmit power to form $\mathbf{y}'[b]$. Then, as in the single-user case described in Section 2.3, the receiver applies MMSE scaling and subtracts the random dither to form $\mathbf{y}''[b]$. Finally, the receiver estimates the desired codeword by quantizing $\mathbf{y}''[b]$.

After each block b , the relay decodes $\lambda[b]$ from $\mathbf{y}_r[b]$. It receives the signal

$$\mathbf{y}_r[b] = \sqrt{\bar{\beta}P_s}([\lambda[b] + \mathbf{t}[b]] \bmod \Lambda_s) + \sqrt{\beta P_s}([\lambda_r[b-1] + \mathbf{s}[b-1]] \bmod \Lambda_s) + \mathbf{z}_r[b].$$

The relay first subtracts the $\lambda_r[b-1]$ component—which, supposing decoding in the previous block was successful, it already knows—from $\mathbf{y}_r[b]$ and scales the signal down by $(\bar{\beta}P_s)^{-\frac{1}{2}}$:

$$\mathbf{y}'_r[b] = [\lambda[b] + \mathbf{t}[b]] \bmod \Lambda_s + \mathbf{z}'_r[b],$$

where $\mathbf{z}'_r[b] = \mathbf{z}_r[b]/\sqrt{\bar{\beta}P_s}$. The relay then applies MMSE scaling and subtracts the dither $\mathbf{t}[b]$:

$$\begin{aligned} \mathbf{y}''_r[b] &= [\gamma_r \mathbf{y}'_r[b] - \mathbf{t}[b]] \bmod \Lambda_s \\ &= [\gamma_r(\lambda[b] + \mathbf{z}'_r) + (\gamma_r - 1)\mathbf{t}[b]] \bmod \Lambda_s, \end{aligned}$$

where $\gamma_r = \frac{\bar{\beta}P_s}{\bar{\beta}P_s + N_r}$ is the appropriate MMSE coefficient. The relay then forms the estimate

$$\hat{\lambda}^r(i) = Q_{\Lambda_c}(\mathbf{y}''_r[b]). \quad (2.25)$$

The destination first decodes $\lambda_r[b]$ from $\mathbf{y}_d[b+1]$, which arrives as

$$\begin{aligned} \mathbf{y}_d[b+1] &= (\sqrt{\beta P_s} + \sqrt{P_r})([\lambda_r[b] + \mathbf{s}[b]] \bmod \Lambda_s) \\ &\quad + \sqrt{\bar{\beta}P_s}([\lambda[b+1] + \mathbf{t}] \bmod \Lambda_s) + \mathbf{z}_d[b+1]. \end{aligned}$$

Treating the portion encoding $\lambda[b+1]$ as noise, the destination scales down $\mathbf{y}_d[b+1]$ by $(\sqrt{\beta P_s} + \sqrt{P_r})^{-1}$, yielding

$$\mathbf{y}'_d[b+1] = [\lambda_r[b] + \mathbf{s}[b]] \bmod \Lambda_r + \mathbf{z}'_d[b+1],$$

where $\mathbf{z}'_d[b+1]$ is a scaled version of the interference-plus-noise term. The destination then forms $\mathbf{y}''_d[b+1]$ by applying MMSE scaling and subtracting the dither $\mathbf{s}[b]$:

$$\begin{aligned}\mathbf{y}''_d(i+i) &= [\gamma_r \mathbf{y}'_d(i+1) - \mathbf{s}[b]] \bmod \Lambda_s \\ &= [\gamma_r(\lambda_r[b] + \mathbf{z}'_d[b+1]) - (\gamma_r - 1)\mathbf{s}[b]] \bmod \Lambda_s\end{aligned}$$

where $\gamma_r = \frac{\beta P_s + P_r + 2\sqrt{\beta P_s P_r}}{P_s + P_r + 2\sqrt{\beta P_s P_r} + N_d}$ is the MMSE coefficient. The destination then forms the estimate

$$\hat{\lambda}_r[b] = Q_{\Lambda_r}(\mathbf{y}''_d[b+1]). \quad (2.26)$$

Now that the destination knows $\lambda_r[b]$, it decodes $\lambda_v[b]$ from $\mathbf{y}_d[b]$. First, it scales down the incoming signal by $(\bar{\beta}P_s)^{-\frac{1}{2}}$ and subtracts (modulo Λ_s) $\lambda_r[b]$:

$$\mathbf{y}'_d[b] = [\lambda_v[b] + \mathbf{t}[b] + \mathbf{z}'_d[b]] \bmod \Lambda_s,$$

where, as before, $\mathbf{z}'_d[b]$ is a scaled version of the noise. The destination forms $\mathbf{y}''_d[b]$ by applying MMSE scaling and subtracting the dither \mathbf{t} :

$$\begin{aligned}\mathbf{y}''_d[b] &= [\gamma_v \mathbf{y}'_d[b] - \mathbf{t}] \bmod \Lambda_s^{(n)} \\ &= [\gamma_v(\lambda_v[b] + \mathbf{z}'_d[b]) - (\gamma_v - 1)\mathbf{t}] \bmod \Lambda_s^{(n)},\end{aligned}$$

where $\gamma_v = \frac{\bar{\beta}P_s}{\bar{\beta}P_s + N_d}$ is the appropriate MMSE coefficient. The destination estimates $\lambda_v[b]$ using the lattice quantizer

$$\hat{\lambda}_v[b] = Q_{\Lambda_v}(\mathbf{y}''_d[b]). \quad (2.27)$$

With estimates of $\lambda_r[b]$ and $\lambda_v[b]$, the destination can estimate the desired message

$$\hat{\lambda}^d[b] = [\hat{\lambda}_r[b] + \hat{\lambda}_v[b]] \bmod \Lambda_s. \quad (2.28)$$

Error Probability: Let P_e be the probability of decoding any $\lambda[b]$ incorrectly at the destination, which is governed by the error probabilities of the lattice decoders of (2.25), (2.26) and (2.27):

$$\begin{aligned} P_r &= \Pr \left\{ \bigcup_{b=1}^B \hat{\lambda}^r[b] \neq \lambda[b] \right\} \\ P_0 &= \Pr \left\{ \bigcup_{b=1}^B \hat{\lambda}_r[b] \neq \lambda_r[b] \right\} \\ P_1 &= \Pr \left\{ \bigcup_{b=1}^B \hat{\lambda}_v[b] \neq \lambda_v[b] \right\}. \end{aligned}$$

Provided all three lattice decoders decode correctly, the destination can successfully recover the desired message. By the union bound, P_e satisfies

$$P_e \leq P_r + P_0 + P_1.$$

Since Λ_s is good for covering and Λ_s and Λ_c are good for coding, [74, Theorem 3] guarantees that, averaging over $\mathbf{t}[b]$, P_r can be made small as long as

$$R < \frac{1}{2} \log_2 \left(1 + \frac{\bar{\beta} P_s}{N_r} \right). \quad (2.29)$$

Similarly, since Λ_c is good for AWGN coding and Λ_m is good for coding and for covering, P_0 , averaging over $\mathbf{s}[b]$, can be made small provided

$$R_0 < \frac{1}{2} \log_2 \left(1 + \frac{\beta P_s + P_r + 2\sqrt{\beta P_s P_r}}{\bar{\beta} P_s + N_d} \right). \quad (2.30)$$

Finally, since Λ_m is good for coding and Λ_s is good for coding and covering, P_1 can be made small when

$$R_1 < \log_2 \left(1 + \frac{\bar{\beta} P_s}{N_d} \right). \quad (2.31)$$

The probabilities P_r , P_0 , and P_1 are bounded above by exponential terms which are functions of the Poltyrev exponent [72]. The asymptotic error probability is dominated by the term with the smallest error exponent, corresponding to the rate component closest to the bounds given in (2.29), (2.30), and (2.31).

Combining (2.29), (2.30), and (2.31), it is immediate that P_e can be made small for any rate satisfying (2.24).

User Cooperation

In this chapter, I explore the benefits of user cooperation to physical-layer network coding. I examine a single-layer network with L transmitters and M receivers, in which the receivers intend to recover finite-field linear combinations of the transmitters' message. The transmitters overhear each other's signals and can cooperatively encode portions of their messages. For this topology, I present a decode-and-forward style cooperative scheme, based on lattice block Markov encoding as described in Chapter 2. The proposed scheme provides significant improvement in the achievable computation rate. Furthermore, it improves the diversity-multiplexing tradeoff. For the special case of two users and a single receiver, the proposed scheme is DMT optimal. In the more general case of N transmitters and a single receiver, cooperation affords a DMT improvement; however, it does not achieve the optimum DMT.

3.1 Preliminaries

3.1.1 System model

Here I study the *cooperative computation channel*, depicted in Figure 4.1. In the cooperative computation channel, L transmitters communicate with $M \leq L$ receivers over the wireless medium. Each of the L users has B messages $\mathbf{w}_l[b] \in \mathbb{F}_p^k$, for $1 \leq t \leq T$. Structurally, this channel resembles the compound multiple-access channel or, when $M = L$, the interference channel. However, unlike those more traditional channels, here each receiver intends to decode a finite-field linear combination¹ of the transmitters' messages:

$$\mathbf{f}_m[b] = \bigoplus_{l=1}^L a_{lm} \odot \mathbf{w}_l[b], \quad (3.1)$$

for $a_{lm} \in \mathbb{Z}$. Let the matrix $\mathbf{A} = [a_{lm}] \in \mathbb{Z}^{L \times M}$ describe the functions computed by the receivers.

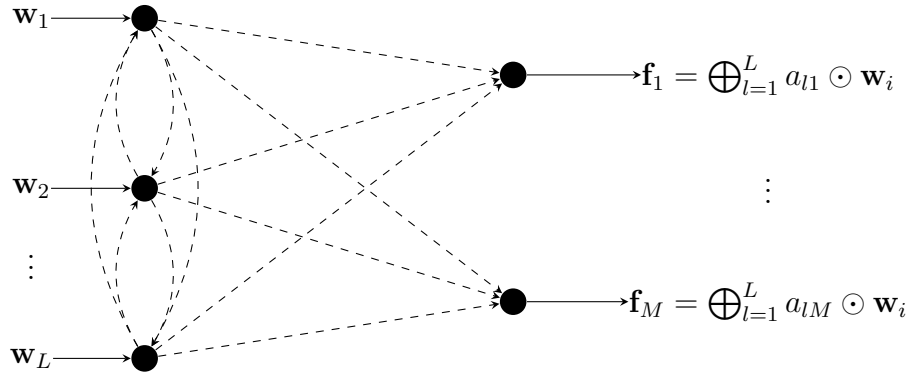


Figure 3.1: The cooperative computation channel. L users cooperatively transmit to M receivers, which decode the desired linear functions.

In order to accommodate block Markov encoding, transmissions are divided into $B+1$ blocks of n channel uses each. At block b , each transmitter l broadcasts a signal

¹Very precisely, receivers compute any of a *sequence* of linear combinations since, as will be shown later, $k, p \rightarrow \infty$ as the codeword length becomes large.

$\mathbf{x}_l[b] \in \mathbb{R}^n$, subject to an average power constraint:

$$\frac{1}{n} \|\mathbf{x}_l[b]\|^2 \leq P,$$

for some $P > 0$. The superposition of the transmitters' signals, scaled by channel coefficients and corrupted by noise, arrives at each receiver:

$$\mathbf{y}_m[b] = \sum_{l=1}^L h_{lm} \mathbf{x}_l[b] + \mathbf{n}[b], \quad (3.2)$$

where $h_{lm} \in \mathbb{R}$ is the channel coefficient from transmitter l to receiver m , and $\mathbf{n}[b]$ is a white, unit-variance Gaussian random vector. For convenience, the channel coefficients are gathered into the matrix $\mathbf{H} = [h_{lm}]$.

Each transmitter l also obtains the noisy superposition of the other transmitters' signals:

$$\mathbf{z}_l[b] = \sum_{\substack{l'=1 \\ l' \neq l}}^L g_{l'l} \mathbf{x}_{l'}[b] + \mathbf{n}_l[b], \quad (3.3)$$

where $g_{l'l} \in \mathbb{R}$ is the channel coefficient from transmitter l' to transmitter l , and $\mathbf{n}_l[b]$ is again white, unit-variance Gaussian. Again define the matrix $\mathbf{G} = [g_{l'l}]$ with diagonal elements equal to zero. The choice of zero for the diagonal elements implies that a transmitters' signal does not self-interfere; in other words, each transmitter is capable of *full-duplex* operation and can transmit and receive simultaneously. Channel matrices \mathbf{H} and \mathbf{G} are taken to be fixed and known globally among the transmitters and receivers.

Also define the *non-cooperative computation channel*, which is identical to the cooperative network except that the transmitters have no access to each other's transmissions. Equivalently, \mathbf{G} is the all-zero matrix.

3.1.2 Computation capacity

The primary figure of merit is the *computation capacity* of the channel. Since the receivers recover functions of incoming messages, rather than the messages themselves, the computation capacity is defined somewhat differently than the usual Shannon capacity. Each transmitter has an encoder $E_l : \mathbb{F}_p^{k \times B} \times \mathbb{R}^{n \times B} \rightarrow \mathbb{R}^{n \times (B+1)}$. That is, the encoder E_l takes as its input the messages $\mathbf{w}_l[b]$ and the received signals $\mathbf{z}_l[b]$ and generates as its output the codewords $\mathbf{x}_l[b]$. The encoder must be causal: The output codeword $\mathbf{x}_l[b]$ may depend on received signals $\mathbf{z}_l[s]$ only for $s < b$. As usual, the encoding rate is defined as the logarithm of the cardinality of the message set divided by the number of channel realizations over which the messages are encoded:

$$R = \frac{B \log_2(|\mathbb{F}_p^k|)}{n(B+1)} = \frac{Bk \log_2(p)}{n(B+1)} \approx \frac{k \log_2(p)}{n}, \quad (3.4)$$

where the approximation holds for large B . Note that this is the *symmetric* rate among all transmitters.

Each receiver has a decoder $D_m : \mathbb{R}^{n \times (B+1)} \rightarrow \mathbb{F}_p^{k \times B}$, taking as inputs the received signals $\mathbf{y}_m[b]$ and generating as outputs the estimates $\hat{\mathbf{f}}_m[b]$. Let the absolute probability of error be the probability that any receiver makes an incorrect estimate of any of the desired functions:

$$P_e = \Pr\{\hat{\mathbf{f}}_m[b] \neq \mathbf{f}_m[b], \text{ for any } 1 \leq m \leq M, 1 \leq b \leq B\}. \quad (3.5)$$

A computation rate R is said to be *achievable* if for any $\epsilon > 0$ there exists a sequence of encoders with encoding rate greater than $R - \epsilon$ and decoders such that $P_e \rightarrow 0$ as $n \rightarrow \infty$. For fixed channel gains \mathbf{H}, \mathbf{G} , function coefficients \mathbf{A} , and transmit power P , let $R(\mathbf{H}, \mathbf{G}, \mathbf{A}, P)$ denote the supremum over all achievable computation rates.

In order to define the computation capacity, there must be limitations on the

permissible function coefficients \mathbf{A} . Otherwise one could choose a trivial coefficient matrix, such as the all-zero matrix, for which the achievable computation rate is unbounded. Therefore, \mathbf{A} must be a member of the following set:

$$\mathcal{A} = \{\mathbf{A} \in \mathbb{Z}^{L \times M} : \text{rank}(\mathbf{A}) = M, \forall m \exists l \text{ such that } a_{ml} \neq 0\}. \quad (3.6)$$

The first condition ensures that the recovered functions retain as much information as possible about the individual transmitters' messages; for $L = M$ it implies that one can recover the individual messages from the recovered functions. The second condition, which is redundant for $L = M$, ensures that each transmitter is represented in the recovered messages; the receivers cannot simply ignore a transmitter in order to achieve a higher computation rate.

Finally, define the *computation capacity* as the supremum of achievable rates over the set of permissible coefficient matrices:

$$C(\mathbf{H}, \mathbf{G}, P) = \sup_{\mathbf{A} \in \mathcal{A}} R(\mathbf{H}, \mathbf{G}, \mathbf{A}, P). \quad (3.7)$$

In their seminal work, Nazer and Gastpar developed a computation strategy based on nested lattice codes [36]. It achieves the following computation rate:

$$R_{\text{nc}}(\mathbf{H}, P) = \max_{\mathbf{A} \in \mathcal{A}} \min_{1 \leq m \leq M} \left[\frac{1}{2} \log_2(1 + P \|\mathbf{h}_m\|^2) - \frac{1}{2} \log_2(\|\mathbf{a}_m\|^2 + P(\|\mathbf{a}_m\|^2 \|\mathbf{h}_m\|^2 - |\mathbf{a}_m^T \mathbf{h}_m|^2)) \right]^+. \quad (3.8)$$

The first term in (3.8) corresponds to the power in the received signal, whereas the second term is a penalty determined by the gap in the Cauchy-Schwarz inequality between \mathbf{h}_m and \mathbf{a}_m . The closer \mathbf{h}_m and \mathbf{a}_m are to being co-linear, the smaller is the rate penalty. Since the Nazer-Gastpar scheme was designed for a non-cooperative

network, the rate does not depend on \mathbf{G} ; nevertheless, it serves as a lower bound on the cooperative computation capacity $C(\mathbf{H}, \mathbf{G})$.

3.1.3 Diversity-multiplexing tradeoff

A major advantage of user cooperation is that cooperating transmitters can achieve performance similar to that of a multiple-antenna transmitter. Multiple antennas can improve performance on two fronts: increased reliability in the presence of channel fading, and increased throughput. In the high-SNR regime, the *diversity-multiplexing tradeoff* quantifies this improvement [92]. Let the elements of \mathbf{H} and \mathbf{G} be identically and independently distributed according to a Rayleigh distribution. Next, suppose there is a scheme that achieves the computation rate $R_{\text{scheme}}(\mathbf{H}, \mathbf{G}, P)$. Then, the *diversity order at multiplexing gain r* is defined as

$$d(r) = \lim_{P \rightarrow \infty} \frac{\log \Pr\{R_{\text{scheme}}(\mathbf{H}, \mathbf{G}, P) < \frac{r}{2} \log(P)\}}{\log P}. \quad (3.9)$$

In other words, $d(r)$ is the exponent of the outage probability, with the rate taken to have multiplexing gain r , as the SNR goes to infinity. The diversity-multiplexing tradeoff of the system, denoted by $d^*(r)$, is the supremum of $d(r)$ over all possible schemes.

The multiplexing gain for compute-and-forward is studied in [44]. There it is shown that, using the Nazer-Gastpar approach, the multiplexing gain can be no higher than $\frac{2}{L+1}$. In other words, $d(r) = 0$ for $r > \frac{2}{L+1}$ for this scheme.

3.2 Computation Rates

3.2.1 Upper bounds

First I present two upper bounds on the cooperative computation rate. I obtain the first bound by supposing that the transmitters are capable of perfect cooperation, which is equivalent to having a genie supply all messages to each transmitter. The problem then reduces to a multiple-input, single-output (MISO) broadcast channel, the capacity of which is known [86]. In Section 3.3 I use this result to bound the diversity-multiplexing tradeoff.

Theorem 3.1. *Denote the capacity region of a Gaussian MISO broadcast channel by*

$$\mathcal{C}_{\text{miso}}(\mathbf{H}, P) = \text{conv} \left\{ \bigcup_{\pi \in \Pi} \left\{ \mathbf{r} : r_m \leq \frac{1}{2} \log_2 \left(1 + \frac{\mathbf{h}_{\pi(m)} \mathbf{V}_{\pi(m)} \mathbf{h}_{\pi(m)}^T}{\mathbf{h}_{\pi(m)} \sum_{i=1}^{m-1} \mathbf{V}_{\pi(i)} \mathbf{h}_{\pi(m)}^T + 1} \right) \right\} \right\}, \quad (3.10)$$

where $\text{conv}\{\cdot\}$ is the convex hull, Π is the set of permutations from $\{1, \dots, L\}$ to itself, and \mathbf{V}_m is a collection of positive semi-definite matrices such that $\sum_{m=1}^M \text{tr}(\mathbf{V}_m) \leq NP$. Then the computation capacity of the cooperative computation channel is bounded above by

$$C(\mathbf{H}, \mathbf{G}, P) \leq R_{\text{miso}}^+(\mathbf{H}, P), \quad (3.11)$$

where

$$R_{\text{miso}}^+(\mathbf{H}, P) = \sup\{r : r\mathbf{1} \in \mathcal{C}_{\text{miso}}(\mathbf{H}, P)\} \quad (3.12)$$

is the symmetric-rate capacity of the Gaussian MISO broadcast channel.

To prove Theorem 3.1, a quick lemma is necessary.

Lemma 3.1. *Let $\mathbf{w}_1, \dots, \mathbf{w}_L \in \mathbb{F}_p^k$ be independently and uniformly distributed messages. Then, the functions $\mathbf{f}_1, \dots, \mathbf{f}_M$ are also independent and uniformly distributed across \mathbb{F}_p^k .*

Proof. Since the finite-field linear combinations in \mathbf{f}_i are taken element-wise, it is sufficient to show the result for an arbitrary element of both messages and functions. Therefore, let $\mathbf{w} = (w_{11}, \dots, w_{L1})^T$ and $\mathbf{f} = (f_{11}, \dots, f_{M1})^T = \mathbf{A}\mathbf{w}$. It remains to be shown that the elements of \mathbf{f} are independent and uniformly distributed.

Since \mathbf{w} is uniformly distributed over \mathbb{F}_p^L , its probability mass function is

$$p(\mathbf{w}) = p^{-L}. \quad (3.13)$$

The conditional pmf of \mathbf{f} is, trivially,

$$p(\mathbf{f}|\mathbf{w}) = \delta(\mathbf{f} - \mathbf{A}\mathbf{w}), \quad (3.14)$$

where $\delta(\cdot)$ is the Kronecker delta function. Now, compute the marginal pmf for \mathbf{f} :

$$p(\mathbf{f}) = \sum_{\mathbf{w} \in \mathbb{F}_p^L} p(\mathbf{f}|\mathbf{w})p(\mathbf{w}) \quad (3.15)$$

$$= p^{-L} \sum_{\mathbf{w} \in \mathbb{F}_p^L} \delta(\mathbf{f} - \mathbf{A}\mathbf{w}) \quad (3.16)$$

$$= p^{-L} |\{\mathbf{w} | \mathbf{A}\mathbf{w} = \mathbf{f}\}| \quad (3.17)$$

$$= p^{-L} p^{L-M} = p^{-M}, \quad (3.18)$$

where (3.17) follows because \mathbf{A} is full rank. Since the pmf $p(\mathbf{f})$ does not depend on \mathbf{f} , the elements are independent and uniformly distributed. \square

With Lemma 3.1, it is straightforward to prove Theorem 3.1.

Proof of Theorem 3.1. Suppose that a genie provides the messages $\mathbf{w}_l(t)$ to each of the transmitters. Then the transmitters each can compute the functions $\mathbf{f}_m(t)$. By Lemma 3.1 these functions are independent and uniformly distributed, so the scenario is equivalent to an L -transmitter antenna having M independent messages to send to M users. In [86] the capacity region is shown to be (3.10). Since the computation capacity is defined in terms of achievable *symmetric* rate, it cannot exceed the symmetric-rate MISO capacity given in (3.12). \square

I obtain the second bound by supposing that a genie supplies to the receivers all messages except for those of a single transmitter l . Then the receivers need only to recover the messages of transmitter l in order to compute any suitable set of functions. This converts the system to a compound relay channel in which the other transmitters serve as dedicated relays, the capacity of which can be bound using cut-set arguments. This upper bound is somewhat more realistic than R_{miso}^+ , and I use it in Section 3.4 for comparisons to the achievable rates.

Theorem 3.2. *For each transmitter $1 \leq l \leq L$, let $\mathbf{S}_l = \{1, \dots, l-1, l+1, \dots, L\}$ be the set of transmitters other than transmitter l . Then the computation capacity of the cooperative computation channel is bounded above by*

$$C(\mathbf{H}, \mathbf{G}, P) \leq R_{\text{single}}^+(\mathbf{H}, \mathbf{G}, P),$$

where

$$R_{\text{single}}^+(\mathbf{H}, \mathbf{G}, P) = \min_{1 \leq l \leq L} \max_{\mathbf{A} \in \mathcal{A}} \min_{m, a_{lm} \neq 0} \max_{p(\mathbf{x})} \min_{S \in \mathbf{S}_l} I(x_l, x_S; y_m, z_{S^c} | x_{S^c}), \quad (3.19)$$

where $p(\mathbf{x})$ is any distribution over the transmitted signals $(x_1, \dots, x_L)^T$ satisfying the input power constraint.

Proof. Choose a transmitter l , and suppose that a genie supplies the messages $w_{l'}(t)$ to the receivers for every $l' \neq l$. By the crypto lemma [74], each $\mathbf{f}_m(t)$ such that $a_{lm} \neq 0$ is statistically independent of the messages $w_{l'}(t)$, so the receivers remain equivocal as to the desired functions. Thus the scenario is equivalent to a compound relay channel, with transmitter l acting as the source, the transmitters l' acting as relays, and each receiver m such that $a_{lm} \neq 0$ acting as destinations all needing the messages $w_l(t)$. The capacity of the compound relay channel can be bounded using cut-set arguments. For any cut $S \in \mathcal{S}_l$, the capacity of the compound relay channel, and thus the computation capacity of the cooperative computation channel, is bounded by

$$C(\mathbf{H}, \mathbf{G}, P) \leq \max_{p(\mathbf{x})} \min_{m, a_{lm} \neq 0} I(x_l, x_S; y_m, z_{S^c} | x_{S^c}) \quad (3.20)$$

$$\leq \min_{m, a_{lm} \neq 0} \max_{p(\mathbf{x})} I(x_l, x_S; y_m, z_{S^c} | x_{S^c}). \quad (3.21)$$

Taking the minimum over all transmitters and all cuts S yields the result. \square

3.2.2 Lower Bound

Here I detail the proposed cooperative computation strategy and derive the computations rates it achieves. The scheme is decode-and-forward in nature: at one block transmitters send out lattice codewords corresponding to their individual messages; these messages are decoded by other transmitters. At the next block transmitters cooperatively encode resolution information to assist the receivers. As with any decode-and-forward strategy, one must contend with the fact that it may be difficult for transmitters to decode each other's messages. Therefore, not every transmitter will cooperate.¹ A subset \mathcal{B} of the transmitters decodes the messages of every other

¹Other approaches are possible. For example, in an earlier work [93] transmitters are partitioned into clusters; transmitters decode only in-cluster messages. In the interest of brevity I discuss only

user, after which they cooperatively transmit resolution information to the receivers. Transmitters not in \mathcal{B} , not having decoded incoming messages, do not send any resolution information.

Theorem 3.3. *Let $\mathcal{B} \subset \{1, \dots, L\}$. In the cooperative computation channel, the following computation rate is achievable:*

$$R_c(\mathbf{H}, \mathbf{G}, P) = \max_{\mathbf{A} \in \mathcal{A}} \min \left\{ \min_{l \in \mathcal{B}} C_{\text{mac}}(\mathbf{g}_l[1 : l-1, l+1 : L] \circ \mathbf{v}_0[1 : l-1, l+1 : L], P, 1), \right. \\ \min_{1 \leq m \leq M} \left\{ \frac{1}{2} \log_2 \left(1 + \frac{P |\mathbf{h}_m^T \mathbf{v}_m|^2}{1 + I_{m,r}} \right) + \right. \\ \left. \left[\frac{1}{2} \log_2 (P \|\mathbf{h}_m \circ \mathbf{v}_0\|^2 + I_{m,v}) - \frac{1}{2} \log_2 (\|\mathbf{a}_m\|^2 (1 + I_{m,v}) + P (\|\mathbf{a}_m\|^2 \|\mathbf{h}_m \circ \mathbf{v}_0\|^2 - |\mathbf{a}_m^T (\mathbf{h}_m \circ \mathbf{v}_0)|^2)) \right]^+ \right\} \Bigg\}, \quad (3.22)$$

where

$$I_{m,r} = P \left(\|\mathbf{h}_m \circ \mathbf{v}_0\|^2 + \sum_{m' \neq m, 0} |\mathbf{h}_m^T \mathbf{v}_{m'}|^2 \right) \quad (3.23)$$

is the interference power seen at receiver m as it decodes its resolution information,

$$I_{m,v} = P \sum_{m' \neq m, 0} |\mathbf{h}_m^T \mathbf{v}_{m'}|^2 \quad (3.24)$$

is the interference seen at receiver m as it decodes the vestigial information, and for any vectors $\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_M$ such that

$$\sum_{m=0}^M |v_{lm}|^2 \leq 1, \forall l \quad (3.25)$$

and $v_{lm} = 0$ for $l \neq \mathcal{B}$ and $m > 0$.

The achievable rate (3.22) comprises three main components. First is the rate of a the approach presented in Theorem 3.3.

Gaussian multiple-access channel, which corresponds to the rate at which cooperating transmitters can decode others' messages. Second is the rate at which each receiver can decode the resolution information, which is that of a virtual MISO link between cooperating transmitters and the receiver; signals unrelated to the resolution information are treated as noise. Third is the rate at which the receivers, having already decoded the resolution information, can decode the vestigial component of the desired combination of lattice points; this is the Nazer-Gastpar rate of (3.8), with resolution information intended for other receivers treated as noise.

Each transmitter splits its power between sending its own lattice codewords and cooperatively sending resolution information. The split is defined by the steering vectors $\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_M$. Each element v_{l0} dictates the fraction of power transmitter l expends on its own lattice codewords. For cooperating transmitter l , each element v_{lm} dictates the fraction of power expended on resolution information for receiver m . The steering vectors introduce two separate notions of alignment. First, one can choose \mathbf{v}_0 in order to minimize the Cauchy-Schwarz penalty in (3.22). Second, one can choose the remaining vectors \mathbf{v}_m to trade off between increasing the coherence gain at the intended receivers and decreasing the interference generated at other receivers. Finding the optimum steering vectors is a non-convex problem; for further results and in simulations I rely on several heuristics for selecting them.

Before proving Theorem 3.3, I examine a few of its corollaries. One can obtain a simpler expression for the achievable rate by choosing $\mathcal{B} = \{1, \dots, L\}$ and taking the steering vectors $\mathbf{v}_1, \dots, \mathbf{v}_M$ to be zero-forcing beamformers. Thus the cooperative signals do not interfere at other receivers.

Corollary 3.1. *The following computation rate is achievable for the cooperative com-*

putation channel:

$$R_{\mathbf{zf}}(\mathbf{H}, \mathbf{G}, P) = \max_{\mathbf{A} \in \mathcal{A}} \min \left\{ \min_{1 \leq l \leq L} C_{\text{mac}}(\mathbf{g}_l[1 : l-1, l+1 : L] \circ \mathbf{v}_0[1 : l-1, l+1 : L], P, 1), \right. \\ \left. \min_{1 \leq m \leq M} \left[\frac{1}{2} \log_2(1 + P(\|\mathbf{h}_m \circ \mathbf{v}_0\|^2 + |\mathbf{h}_m^T \mathbf{v}_m|^2)) - \right. \right. \\ \left. \left. \frac{1}{2} \log_2(\|\mathbf{a}_m\|^2 + P(\|\mathbf{a}_m\|^2 \|\mathbf{h}_m \circ \mathbf{v}_0\|^2 - |\mathbf{a}_m^T(\mathbf{h}_m \circ \mathbf{v}_0)|^2)) \right]^+ \right\}, \quad (3.26)$$

for any vectors $\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_M$ satisfying

$$\sum_{m=0}^M |v_{lm}|^2 \leq 1 \quad (3.27)$$

and

$$\mathbf{v}_m^T \mathbf{h}_{m'} = 0, \forall m \neq m'. \quad (3.28)$$

Since $L \geq M$, it is possible to choose non-trivial zero-forcing beamforming vectors for almost every \mathbf{H} .

Finally, choosing $\mathcal{B} = \emptyset$, one obtains an achievable rate for both the cooperative and non-cooperative computation channel. This yields a rate similar to (3.8), except that each transmitter can adjust its transmit power in order to tune the effective channels to match the desired linear functions. In fact this rate is a special case of the “superposition” compute-and-forward presented in [36, Theorem 13].

Corollary 3.2. *In both the non-cooperative computation channel and the cooperative computation channel, the following rate is achievable:*

$$R(\mathbf{H}, \mathbf{G}, P) = \max_{\mathbf{A} \in \mathcal{A}} \min_{1 \leq m \leq M} \left[\frac{1}{2} \log_2(1 + P(\|\mathbf{h}_m \circ \mathbf{v}_0\|^2)) - \right. \\ \left. \frac{1}{2} \log_2(\|\mathbf{a}_m\|^2 + P(\|\mathbf{a}_m\|^2 \|\mathbf{h}_m \circ \mathbf{v}_0\|^2 - |\mathbf{a}_m^T(\mathbf{h}_m \circ \mathbf{v}_0)|^2)) \right]^+, \quad (3.29)$$

for any \mathbf{v}_0 satisfying

$$|v_{l0}|^2 \leq 1, \forall 1 \leq l \leq L. \quad (3.30)$$

Proof of Theorem 3.3. The proof goes in three parts: a description of the encoding scheme, a description of the decoding scheme, and an analysis of the probability of error.

Encoding: Each transmitter employs identical lattice codebooks \mathcal{C} having rate R_c . As shown in Chapter 2, the codebook \mathcal{C} decomposes into resolution and vestigial codebooks \mathcal{C}_r and \mathcal{C}_v which have respective rates R_r and R_v . Recall that $R_c = R_r + R_v$.

Transmitters encode their B messages over $B + 1$ blocks as depicted in Table 3.1. At block b , each transmitter l has a message $\mathbf{w}_l[b]$, which it encodes by mapping it to the corresponding codeword in \mathcal{C} :

$$\lambda_l[b] = \phi(\mathbf{w}_l[b]). \quad (3.31)$$

By Lemma 2.2, each lattice codeword can be decomposed by projecting onto the resolution and vestigial codebooks:

$$\lambda_{r,l}[b] = \phi_r(\mathbf{w}_l[b])$$

$$\lambda_{v,l}[b] = \phi_v(\mathbf{w}_l[b]).$$

Each transmitter dithers the lattice codewords over the shaping region. Therefore, define the effective codeword

$$\mathbf{c}_l[b] = [\lambda_l[b] + \mathbf{t}_l[b]] \bmod \Lambda_s, \quad (3.32)$$

where $\mathbf{t}_l[b]$ is a dither drawn randomly and uniformly over \mathcal{V}_s , independent for each $1 \leq l \leq L$ and $1 \leq t \leq T$. Each receiver m intends to recover the finite-field linear

combination $\mathbf{f}_m[b] = \bigoplus_{l=1}^L a_{lm} \mathbf{w}_m[b]$, which corresponds to the lattice point

$$\lambda_m[b] = \phi(\mathbf{f}[b]) = \left[\sum_{l=1}^L a_{lm} \lambda_l[b] \right] \bmod \Lambda_s. \quad (3.33)$$

As with the individual codewords, one can decompose $\lambda_m[b]$ into resolution and vestigial components:

$$\lambda_{r,m}[b] = \phi_r(\mathbf{f}_m[b]) \quad (3.34)$$

$$\lambda_{v,m}[b] = \phi_v(\mathbf{f}_m[b]). \quad (3.35)$$

The transmitters in \mathcal{B} will cooperatively transmit $\lambda_{r,m}[b]$ to each receiver, again dithering the lattice point over \mathcal{V}_s . The effective codeword is

$$\mathbf{c}_{r,m} = [\lambda_{r,m}[b] + \mathbf{s}_m[b]] \bmod \Lambda_s, \quad (3.36)$$

where, similar to before, $\mathbf{s}_m[b]$ is a dither drawn uniformly over \mathcal{V}_s and independent for each $1 \leq m \leq M$, and $1 \leq t \leq B$.

At block $t = 1$, each transmitter simply sends its own lattice codeword:

$$\mathbf{x}_l(1) = \sqrt{P} v_{l0} \mathbf{c}_l[b]. \quad (3.37)$$

For subsequent blocks $2 \leq t \leq B$, each transmitter in \mathcal{B} sends a combination of “fresh” information corresponding to its own message $\mathbf{w}_l[b]$ and resolution information corresponding to the messages sent in the previous time slot. Suppose that each transmitter in \mathcal{B} has successfully decoded $\lambda_{l'}[b-1]$ for each $l' \neq l$. Then each transmitter in \mathcal{B} can construct every $\lambda_m[b]$ and, by extension, every $\lambda_{r,m}[b]$. Every transmitter sends its own lattice codeword, and transmitters in \mathcal{B} send the resolution

components for each receiver:

$$\mathbf{x}_l[b] = \begin{cases} \sqrt{P} \left(v_{l0} \mathbf{c}_l[b] + \sum_{m=1}^M v_{lm} \mathbf{c}_{r,m}[b-1] \right), & \text{for } l \in \mathcal{B} \\ \sqrt{P} v_{l0} \mathbf{c}_l[b], & \text{otherwise} \end{cases}. \quad (3.38)$$

Finally, at block $b = B + 1$ there is no new fresh information for the transmitters to send. Each transmitter in \mathcal{B} sends only the resolution information corresponding to block B , and the other transmitters send nothing:

$$\mathbf{x}_l[B+1] = \begin{cases} \sqrt{P} \sum_{m=1}^M v_{lm} \mathbf{c}_{r,b(l)m}[B], & \text{for } l \in \mathcal{B} \\ 0 & \text{otherwise} \end{cases}. \quad (3.39)$$

Table 3.1: Superposition Block Markov encoding for Theorem 3.3

	$b = 1$	$b = 2$	\dots	$b = B + 1$
$\mathbf{x}_1[b], 1 \in \mathcal{B}$	$v_{10} \mathbf{c}_1[1]$	$v_{10} \mathbf{c}_1[2] + \sum_{m=1}^M v_{1m} \mathbf{c}_{r,m}[1]$	\dots	$\sum_{m=1}^M v_{1m} \mathbf{c}_{r,m}[B]$
$\mathbf{x}_2[b], 2 \notin \mathcal{B}$	$v_{20} \mathbf{c}_2[1]$	$v_{20} \mathbf{c}_2[2]$	\dots	0
\vdots	\vdots	\vdots	\vdots	\vdots
$\mathbf{x}_L[b], L \in \mathcal{B}$	$v_{L0} \mathbf{c}_L[1]$	$v_{L0} \mathbf{c}_L[2] + \sum_{m=1}^M v_{Lm} \mathbf{c}_{r,m}[1]$	\dots	$\sum_{m=1}^M v_{Lm} \mathbf{c}_{r,m}[B]$

Note that, since Λ_s has normalized second moment equal to unity, and since the dithers are independently and uniformly drawn from \mathcal{V}_s , with high probability

$$\frac{1}{n} \|\mathbf{x}_l[b]\|^2 \rightarrow P \sum_{m=0}^M v_{lm}^2 \leq P. \quad (3.40)$$

Thus the transmit signals obey the average power constraint.

Decoding: Decoding proceeds in three stages. Each transmitter decodes the messages of every other transmitter, the receivers decode the resolution information send cooperatively by the clusters, and finally the receivers decode the vestigial information. Having decoded both components of the desired lattice point, the receiver

can recover the desired linear function.

At block $t = 1$ each transmitter receives the superposition of all the other transmitters' signals, scaled by channel gains and corrupted by noise:

$$\mathbf{z}_l[1] = \sqrt{P} \sum_{l' \neq l} v_{l'0} g_{l'l} \mathbf{c}_{l'}[1] + \mathbf{n}_l[1]. \quad (3.41)$$

Each transmitter forms estimates $\hat{\mathbf{w}}_{l'l}[1]$ for every $l' \neq l$ via typical sequence decoding: if there is a unique collection of messages jointly typical with the received signal, that collection is taken as the estimate; otherwise an error is declared. Note that here the transmitters do *not* employ lattice decoding.

For blocks $2 \leq t \leq B$ the situation is similar. Each transmitter receives the superposition of other transmitters' signals, but here the received signals also contain resolution information:

$$\mathbf{z}_l[b] = \sqrt{P} \left(\sum_{l' \neq l} g_{l'l} v_{l'0} \mathbf{c}_{l'}[b] + \sum_{l' \in \mathcal{B}} \sum_{m=1}^M g_{l'l} v_{l'm} \mathbf{c}_{r,m}(t-1) \right) + \mathbf{n}_l[b]. \quad (3.42)$$

Supposing that each transmitter has successfully decoded the messages from block $t-1$, it also knows the resolution information. It therefore can subtract this component out, resulting in the effective signal

$$\mathbf{z}'_l[b] = \mathbf{z}_l[b] - \sqrt{P} \sum_{l' \in \mathcal{B}} \sum_{m=1}^M g_{l'l} v_{l'm} \mathbf{c}_{r,m}(t-1) \quad (3.43)$$

$$= \sqrt{P} \sum_{l' \neq l} g_{l'l} v_{l'0} \mathbf{c}_{l'}[b] + \mathbf{n}_l[b] \quad (3.44)$$

Now, just as for $b = 1$, each transmitter can form estimates $\hat{\mathbf{w}}_{l'l}[b]$ of the other transmitters' messages via typical sequence decoding.

Next, consider the receivers. To decode the function $\mathbf{f}_m[b]$, each receiver first

decodes the resolution information from the signal received in block $t + 1$:

$$\begin{aligned} \mathbf{y}_m[b + 1] = \sqrt{P} \sum_{l=1}^L h_{lm} v_{l0} \mathbf{c}_l[b + 1] + \sqrt{P} \sum_{m' \neq m}^M \sum_{l \in \mathcal{B}} h_{lm} v_{lm'} \mathbf{c}_{l,m'}[b] + \\ \sqrt{P} \sum_{l \in \mathcal{B}} h_{lm} v_{lm} \mathbf{c}_{r,m}[b] + \mathbf{n}_m[b + 1]. \end{aligned} \quad (3.45)$$

Each receiver decodes the resolution information treating the interference—the fresh information from each transmitter and the resolution information intended for other receivers—as noise. Each estimate $\hat{\lambda}_{r,m}[b]$ is formed via lattice decoding as outlined in Chapter 2. The receivers first apply MMSE scaling to the incoming signal and subtract off the dither. Let

$$\mathbf{n}'_m[b + 1] = \sqrt{P} \sum_{l=1}^L h_{lm} v_{l0} \mathbf{c}_l[b + 1] + \sqrt{P} \sum_{m' \neq m}^M \sum_{l \in \mathcal{B}} h_{lm} v_{lm'} \mathbf{c}_{l,m'}[b] + \mathbf{n}_m[b + 1] \quad (3.46)$$

be the sum of the interference and noise at receiver m . Then the scaled signal is

$$\mathbf{y}'_m[b + 1] = [\gamma_m[b + 1] \mathbf{y}_m[b + 1] - \mathbf{s}_m[b + 1]] \bmod \Lambda_s \quad (3.47)$$

$$= \left[\gamma_m[b + 1] \sqrt{P} \sum_{l \in \mathcal{B}} h_{lm} v_{lm} \mathbf{c}_{r,m}[b] + \gamma_m \mathbf{n}'_m[b + 1] - \mathbf{s}_m[b + 1] \right] \bmod \Lambda_s \quad (3.48)$$

$$= \left[\lambda_{r,m}[b + 1] + \left(\gamma_m[b + 1] \sqrt{P} \sum_{l \in \mathcal{B}} h_{lm} v_{lm} - 1 \right) \mathbf{c}_{r,m}[b] + \right. \\ \left. \gamma_m[b + 1] \mathbf{n}'_m[b + 1] \right] \bmod \Lambda_s \quad (3.49)$$

$$= [\lambda_{r,m}[b + 1] + \mathbf{n}''_m[b + 1]] \bmod \Lambda_s, \quad (3.50)$$

where

$$\mathbf{n}_m''[b+1] = \left(\gamma_m[b+1] \sqrt{P} \sum_{l \in \mathcal{B}} h_{lm} v_{lm} - 1 \right) \mathbf{c}_{r,m}[b] + \gamma_m[b+1] \mathbf{n}_m'[b+1] \quad (3.51)$$

is the effective noise, including thermal noise, interference, and self-noise associated with MMSE scaling. Then, the estimate is formed by lattice quantization:

$$\hat{\lambda}_{r,m}[b] = Q_{\Lambda_r}(\mathbf{y}_m'[b+1]). \quad (3.52)$$

After decoding the resolution information, each receiver turns to $\mathbf{y}_m[b]$ to decode the vestigial component $\lambda_{v,m}[b]$. First, note that, supposing that each receiver has successfully decoded the resolution information from the previous block, it can subtract away that portion of the interference, yielding:

$$\mathbf{y}_m'[b] = \mathbf{y}_m[b] - \sqrt{P} \sum_{l \in \mathcal{B}} h_{lm} v_{lm} \mathbf{c}_{r,m}[b-1] \quad (3.53)$$

$$= \sqrt{P} \sum_{l=1}^L h_{lm} v_{l0} \mathbf{c}_l[b] + \sqrt{P} \sum_{m' \neq m} \sum_{l \in \mathcal{B}} h_{lm} v_{lm'} \mathbf{c}_{r,m'}[b-1] + \mathbf{n}_m[b]. \quad (3.54)$$

Furthermore, supposing that the resolution information was decoded successfully, each receiver can subtract $\lambda_{r,m}[b]$ from the received signal modulo the shaping lattice. Finally, in preparation for lattice decoding, receivers apply MMSE scaling to the signal and subtract the dithers as in [36, 74]. Let

$$\mathbf{n}_m'[b] = \sqrt{P} \sum_{m' \neq m} \sum_{l \in \mathcal{B}} h_{lm} v_{lm'} \mathbf{c}_{r,m'}[b-1] + \mathbf{n}_m[b] \quad (3.55)$$

be the sum of the interference and noise in $\mathbf{y}_m[b]$. The resulting signal is then

$$\mathbf{y}_m''[b] = \left[\alpha_m[b] \mathbf{y}_m'[b] - \lambda_{v,m}[b] - \sum_{l=1}^L a_{lm} \mathbf{t}_l[b] \right] \bmod \Lambda_s \quad (3.56)$$

$$= \left[\sum_{l=1}^L (\alpha_m[b] \sqrt{P} h_{lm} v_{l0} \mathbf{c}_l[b] - a_{lm} \mathbf{t}_l[b]) - \lambda_{r,m}[b] + \alpha_m[b] \mathbf{n}_m'[b] \right] \bmod \Lambda_s \quad (3.57)$$

$$= \left[\sum_{l=1}^L a_{lm} (\mathbf{c}_l[b] - \mathbf{t}_l[b]) - \lambda_{r,m}[b] + \sum_{l=1}^L (\alpha_m[b] \sqrt{P} h_{lm} v_{lm} - a_{lm}) \mathbf{c}_l[b] + \alpha_m[b] \mathbf{n}_m'[b] \right] \bmod \Lambda_s \quad (3.58)$$

$$= \left[\lambda_m[b] - \lambda_{r,m}[b] + \sum_{l=1}^L (\alpha_m[b] \sqrt{P} h_{lm} v_{lm} - a_{lm}) \mathbf{c}_l[b] + \alpha_m[b] \mathbf{n}_m'[b] \right] \bmod \Lambda_s \quad (3.59)$$

$$= \left[\lambda_{v,m}[b] + \sum_{l=1}^L (\alpha_m[b] \sqrt{P} h_{lm} v_{lm} - a_{lm}) \mathbf{c}_l[b] + \alpha_m[b] \mathbf{n}_m'[b] \right] \bmod \Lambda_s \quad (3.60)$$

$$= [\lambda_{v,m}[b] + \mathbf{n}_m''[b]] \bmod \Lambda_s, \quad (3.61)$$

where

$$\mathbf{n}''[b] = \sum_{l=1}^L (\alpha_m[b] \sqrt{P} h_{lm} v_{lm} - a_{lm}) \mathbf{c}_l[b] + \alpha_m[b] \mathbf{n}_m'[b] \quad (3.62)$$

is the effective noise, including thermal noise, interference from other transmitters and clusters, and self-noise associated with MMSE scaling. Each receiver decodes the estimate $\hat{\lambda}_{v,m}[b]$ by quantizing to the nearest point in Λ_v :

$$\hat{\lambda}_{v,m}[b] = Q_{\Lambda_v}(\mathbf{y}_m''[b]). \quad (3.63)$$

Finally, having recovered both the resolution and vestigial components, each receiver constructs its estimate of the desired lattice codeword, from which it can recover

the desired finite-field message:

$$\hat{\mathbf{f}}_m[b] = \phi^{-1}(\hat{\lambda}_m[b]) = \phi^{-1} \left(\left[\hat{\lambda}_{r,m}[b] + \hat{\lambda}_{v,m}[b] \right] \bmod \Lambda_s \right). \quad (3.64)$$

Probability of error: An error occurs when (a) any of the transmitters in \mathcal{B} fails to decode the other transmitters' messages, (b) any of the receivers fails to decode correctly the incoming resolution information, or (c) when any of the receivers fails to decode correctly the vestigial information associated with the desired lattice point. By the union bound, the probability of error follows

$$P_e \leq \sum_{b=1}^B \sum_{m=1}^M \Pr\{\hat{\mathbf{f}}_m[b] \neq \mathbf{f}_m[b]\} \quad (3.65)$$

$$\begin{aligned} &\leq \sum_{b=1}^B \sum_{l \in \mathcal{B}} \sum_{l' \neq l} \Pr\{\hat{\mathbf{w}}_{l'l}[b] \neq \mathbf{w}_{l'}[b]\} + \sum_{b=1}^B \sum_{m=1}^M \Pr\{\hat{\lambda}_{r,m}[b] \neq \lambda_{r,m}[b]\} + \\ &\quad \sum_{b=1}^B \sum_{m=1}^M \Pr\{\hat{\lambda}_{v,m}[b] \neq \lambda_{v,m}[b]\}. \end{aligned} \quad (3.66)$$

It remains to show that as long as the rates satisfy (3.22), each error term in (3.66) goes to zero exponentially. I start with the first summation. Each transmitter decodes the messages within its cluster via typical sequence decoding while treating all out-of-cluster interference as noise. By Lemma A.1 the joint mutual information between the transmit codewords $\mathbf{c}_l[b]$ and the receive signal $\mathbf{z}'[b]$ approaches that of a Gaussian multiple-access channel with channel coefficients $g_{l'l}v_{l'0}$, transmit power P , and unit noise power. Therefore, so long as

$$R < \min_{l \in \mathcal{B}} C_{\text{mac}}(\mathbf{g}_l[1 : l-1, l+1 : L] \circ \mathbf{v}_0[1 : l-1, l+1 : L], P, 1), \quad (3.67)$$

then $\Pr\{\hat{\mathbf{w}}_{l'l}[b] \neq \mathbf{w}_{l'}[b]\} \rightarrow 0$ exponentially for each l and $l' \neq l$.

Next, consider the resolution information. Here each receiver decodes $\lambda_{r,m}[b]$ via

lattice decoding on $\mathbf{y}_m(t+1)$. In [74] it is shown that lattice decoding is sufficient to achieve the capacity of the Gaussian channel. From (3.46) it follows that the interference power in $\mathbf{n}'_m(t+1)$ is

$$I_{m,r} = \frac{1}{n} E[\|\mathbf{n}'_m(t+1)\|^2] = P \left(\|\mathbf{h}_m \circ \mathbf{v}_0\|^2 + \sum_{m' \neq m, 0} |\mathbf{h}_m^T \mathbf{v}'_{m'}|^2 \right). \quad (3.68)$$

Similarly, the power of the resolution information in $\mathbf{y}_m(t+1)$ is $P|\mathbf{h}^T \mathbf{v}_m|^2$. Putting these together, if

$$R_r < \min_{1 \leq m \leq M} \frac{1}{2} \log_2 \left(1 + \frac{P|\mathbf{h}_m^T \mathbf{v}_m|^2}{1 + P(\|\mathbf{h}_m \circ \mathbf{v}_0\|^2 + \sum_{m' \neq m} |\mathbf{h}_m^T \mathbf{v}'_{m'}|^2)} \right), \quad (3.69)$$

then $\Pr\{\hat{\lambda}_{r,m}[b] \neq \lambda_{r,m}[b]\} \rightarrow 0$ exponentially for each m .

Finally, consider the vestigial information. Here each receiver decodes $\lambda_{v,m}[b]$ by lattice decoding the sum of multiple incoming lattice points, so the main result from [36] applies directly. The interference power in (3.55) is

$$I_{m,v} = P \sum_{m' \neq m} |\mathbf{h}_m^T \mathbf{v}_{m'}|^2, \quad (3.70)$$

and the effective channel gains in (3.61) are $\mathbf{h}_m \circ \mathbf{v}_0$. Applying these to the rate in (3.8), if

$$R_v < \left[\frac{1}{2} \log_2 (\|P\mathbf{h}_m \circ \mathbf{v}_0\|^2 + I_{m,v}) - \frac{1}{2} \log_2 (\|\mathbf{a}_m\|^2 (1 + I_{m,v}) + P(\|\mathbf{a}_m\|^2 \|\mathbf{h}_m \circ \mathbf{v}_0\|^2 - |\mathbf{a}_m^T (\mathbf{h}_m \circ \mathbf{v}_0)|^2)) \right]^+, \quad (3.71)$$

then $\Pr\{\hat{\lambda}_{v,m}[b] \neq \lambda_{v,m}[b]\} \rightarrow 0$ exponentially.

Recall that $R_c = R_r + R_v$ and $R = \frac{BR_c}{B+1}$. Choosing B arbitrarily large, the desired

result obtains. □

3.3 Diversity-multiplexing Tradeoff

In this section I present inner and outer bounds on the diversity-multiplexing tradeoff, which coincide in several cases.

3.3.1 Non-cooperative Computation Channel

First, I consider the DMT of the non-cooperative computation channel, which is bounded above by that of a scalar Gaussian channel. In the case of a single receiver, one can achieve this upper bound with lattice codes and signal alignment. With the steering vector \mathbf{v}_0 chosen such that the equivalent channel vector is a constant, the achievable rate—and therefore the error probability—is approximately that of a single SISO link.

Theorem 3.4. *For the non-cooperative computation channel, the diversity-multiplexing tradeoff for any scheme is upper-bounded as follows:*

$$d^*(r) \leq d_{\text{nc}}^+(r) = 1 - r. \quad (3.72)$$

For the case of $M = 1$, $d^*(r) = d_{\text{nc}}^+(r)$.

Proof. First I prove the upper bound. For the non-cooperative case, Nazer and Gastpar proved in [36, Theorem 13] that the computation capacity is upper-bounded by

$$C(\mathbf{H}, P) \leq \max_{\mathbf{A} \in \mathcal{A}} \min_{\substack{l, m \\ a_{lm} \neq 0}} \frac{1}{2} \log_2(1 + Ph_{lm}^2) \quad (3.73)$$

$$\leq \frac{1}{2} \log_2(1 + Ph_{lm}^2), \quad (3.74)$$

where l and m can be chosen arbitrarily such that $a_{lm} \neq 0$. Then the computation capacity is bounded by the Shannon capacity of a single SISO link, which is proven in [92] to have diversity-multiplexing tradeoff $d^*(r) = 1 - r$. The computation channel therefore has DMT bounded by

$$d^*(r) \leq d_{\text{nc}}^+(r) = 1 - r. \quad (3.75)$$

To prove the lower bound for $M = 1$, I apply the non-cooperative rate of Corollary 3.2, choosing \mathbf{v}_0 to align with the channels. For multiplexing gain r , choose $\mathbf{a} = \mathbf{1}$ and $v_l^2 = P^{r-1}/h_l^2$, resulting in the achievable computation rate

$$R(\mathbf{H}, P) = \frac{1}{2} \log_2(1 + LP^r) - \frac{1}{2} \log_2(L) \quad (3.76)$$

$$= \frac{1}{2} \log_2 \left(\frac{1 + LP^r}{L} \right) \quad (3.77)$$

$$\geq \frac{1}{2} \log_2(P^r). \quad (3.78)$$

Outage occurs only when the power constraint precludes $v_l^2 = P^{r-1}/h_l^2$. Since $v_l^2 \leq 1$, this occurs when $h_l^2 \leq P^{r-1}$. The probability of outage is therefore

$$P_o \leq \Pr \left\{ \bigcup_{l=1}^L h_l \leq P^{r-1} \right\} \leq \sum_{l=1}^L \Pr \{h_l^2 \leq P^{r-1}\} \approx LP^{r-1}. \quad (3.79)$$

Therefore, the scheme achieves a diversity order at multiplexing gain r of

$$d_{\text{nc}}^-(r) = \lim_{P \rightarrow \infty} -\frac{\log(P_o)}{\log(P)} \quad (3.80)$$

$$\geq \lim_{P \rightarrow \infty} \frac{(1 - r) \log(P) - \log(L)}{\log(P)} \quad (3.81)$$

$$= 1 - r. \quad (3.82)$$

Since this matches the upper bound, the DMT is established. \square

3.3.2 Cooperative Computation Channel

Next, I examine the DMT of the cooperative computation channel, which is upper bounded by that of a single MISO link.

Theorem 3.5. *For the cooperative computation channel, the diversity-multiplexing tradeoff is upper-bounded as*

$$d^*(r) \leq d_c^+(r) = L(1 - r). \quad (3.83)$$

Proof of Theorem 3.5. The result follows directly from the MISO outer bound on the computation capacity in Theorem 3.1. The symmetric-rate capacity of the MISO broadcast channel is trivially upper bounded by the capacity of the single-user MISO link between the source and any destination. Thus the DMT is upper-bounded by that of a single L -antenna MISO link, which is shown in [92] to be $d^*(r) = L(1 - r)$. Therefore,

$$d^*(r) \leq d^+(r) = L(1 - r). \quad (3.84)$$

\square

In the case of $L = 2$ and $M = 1$, the upper bound is tight, and can be achieved by the proposed block Markov strategy.

Theorem 3.6. *For the cooperative computation channel with $L = 2$ transmitters and $M = 1$ receiver, the diversity-multiplexing tradeoff is*

$$d^*(r) = 2(1 - r). \quad (3.85)$$

Proof. That $d^*(r) \leq 2(1-r)$ follows directly from Theorem 3.5. To show that the upper bound is achievable, I invoke the decode-and-forward rate of Theorem 3.3. Choose $\mathbf{a} = (1, 1)^T$, and choose \mathbf{v}_0 in order to align the equivalent channels to \mathbf{a} :

$$v_l = \frac{\min\{h_1, h_2\}}{h_l}. \quad (3.86)$$

From (3.22), it is immediate that each rate satisfies

$$R_{\mathcal{B}}(\mathbf{H}, \mathbf{G}, P) \geq \min \left\{ \min_{l \in \mathcal{B}} \frac{1}{2} \log_2 \left(P \left(\frac{\min\{h_1, h_2\}}{h_{l'}} \right)^2 g_{l',l}^2 \right), \frac{1}{2} \log_2 \left(P \sum_{l \in \mathcal{B}} h_l^2 \right) - \frac{1}{2} \right\}. \quad (3.87)$$

For each \mathcal{B} , define the outage event

$$\mathcal{O}_{\mathcal{B}} = \left\{ R_{\mathcal{B}}(\mathbf{H}, \mathbf{G}, P) < \frac{r}{2} \log(P) \right\}. \quad (3.88)$$

Outage occurs when each choice of \mathcal{B} fails, or

$$\mathcal{O} = \bigcap_{\mathcal{B}} \mathcal{O}_{\mathcal{B}} \subset \mathcal{O}_{\{1\}} \cap \mathcal{O}_{\{2\}}. \quad (3.89)$$

As suggested by the latter inclusion, it is sufficient to consider only $\mathcal{B} = \{1\}$ and $\mathcal{B} = \{2\}$. The achievable rate for each case obeys

$$R_{\{l\}}(\mathbf{H}, \mathbf{G}, P) \geq \min \left\{ \frac{1}{2} \log_2 \left(P \left(\frac{\min\{h_1, h_2\}}{h_{l'}} \right)^2 g_{l',l}^2 \right), \frac{1}{2} \log_2 (P h_l^2) \right\} \quad (3.90)$$

The rate has two terms, the failure of either of which results in the failure of the cooperation modality. Therefore, define the events \mathcal{C}_l , in which transmitter l fails to decode the message from receiver l' , and \mathcal{N}_l , in which, even if transmitter l successfully

decodes, the receiver fails to decode the desired linear combination. The event \mathcal{C}_l can be written as

$$\mathcal{C}_l = \left\{ \frac{1}{2} \log_2 \left(P \left(\frac{\min\{h_1, h_2\}}{h_{l'}} \right)^2 g_{l',l}^2 \right) < \frac{r}{2} \log_2(P) \right\} \quad (3.91)$$

$$= \left\{ \left(\frac{\min\{h_1, h_2\}}{h_{l'}} \right)^2 g_{l',l}^2 < P^{r-1} \right\}. \quad (3.92)$$

Similarly, \mathcal{N}_l can be written as

$$\mathcal{N}_l = \{h_l^2 < P^{r-1}\}. \quad (3.93)$$

By the union bound, the outage probability therefore obeys

$$\Pr(\mathcal{O}) \leq \Pr(\mathcal{N}_1 \cap \mathcal{N}_2) + \Pr(\mathcal{N}_1 \cap \mathcal{C}_2) + \Pr(\mathcal{N}_2 \cap \mathcal{C}_1) + \Pr(\mathcal{C}_1 \cap \mathcal{C}_2), \quad (3.94)$$

which can be expanded as

$$\begin{aligned} \Pr(\mathcal{O}) &\leq \Pr(\{h_1^2 < P^{r-1}\}) \Pr(\{h_2^2 < P^{r-1}\}) + \\ &\quad \Pr\left(\{h_1^2 < P^{r-1}\} \cap \left\{\left(\frac{\min\{h_1, h_2\}}{h_1}\right)^2 g_{1,2}^2 < P^{r-1}\right\}\right) + \\ &\quad \Pr\left(\{h_2^2 < P^{r-1}\} \cap \left\{\left(\frac{\min\{h_1, h_2\}}{h_2}\right)^2 g_{2,1}^2 < P^{r-1}\right\}\right) + \\ &\quad \Pr\left(\left\{\left(\frac{\min\{h_1, h_2\}}{h_2}\right)^2 g_{2,1}^2 < P^{r-1}\right\} \cap \left\{\left(\frac{\min\{h_1, h_2\}}{h_1}\right)^2 g_{1,2}^2 < P^{r-1}\right\}\right) \end{aligned} \quad (3.95)$$

Each term in (3.95) is approximately equal to $P^{2(r-1)}$. The first term entails that both h_1^2 and h_2^2 are small, each of which occurs with approximate probability P^{r-1} . The second term can be estimated similarly: h_1^2 must be small, which implies with high probability that $h_1 < h_2$; therefore $g_{2,1}^2$ must be small. Again, each occurs

with approximate probability P^{r-1} . The third term can be estimated by the same argument. For the final term, suppose without loss of generality that $h_1 < h_2$. Then $g_{1,2}^2$ must be small alongside either h_1^2 or $g_{2,1}^2$ (or both). Again, each term is small with probability P^{r-1} , and since each term involves independently-distributed channels, the total probability is approximately $P^{2(r-1)}$.

Putting the result into the definition of the DMT yields

$$d^*(r) = \lim_{P \rightarrow \infty} -\frac{\Pr(\mathcal{O})}{\log(P)} \quad (3.96)$$

$$\gtrsim -\frac{\log(P^{2(r-1)})}{\log(P)} \quad (3.97)$$

$$= 2(1-r), \quad (3.98)$$

as was to be shown. □

In the case of a single receiver and an arbitrary number of transmitters, deriving the DMT is a more challenging task. The main difficulty is the Cauchy-Schwarz penalty inherent to lattice coding. In the 2×1 case, the optimum DMT is achieved by choosing \mathbf{v}_0 to align the equivalent channels with a suitable function. In the case of multiple transmitters, however, doing so precludes cooperation with high probability. Therefore I present two approaches which, while DMT-suboptimal, garner an improvement over noncooperation. The first is derived using a rather simple strategy employing time sharing and Gaussian codes. It achieves the full diversity gain, but it has somewhat poor multiplexing performance.

Theorem 3.7. *For the cooperative computation channel, the following diversity-multiplexing tradeoff is achievable:*

$$d_{\text{random}}^-(r) = L \min\{1 - 2r, (L-1)(1 - 2(L-1)r)\}. \quad (3.99)$$

In particular, $d_{\text{random}}^-(0) = L$.

The proof of Theorem 3.5, depends on an achievable rate using Gaussian codes and time sharing, which is established in the following lemma.

Lemma 3.2. *Let $\mathcal{B} \subset \{1, \dots, L\}$. In the cooperative computation channel with $M = 1$ receiver, the following computation rate is achievable:*

$$R_{\text{random}}(\mathbf{H}, \mathbf{G}, P) = \min \left\{ \min_{l \in \mathcal{B}} \frac{1}{4} C_{\text{mac}}(\mathbf{g}_l[1 : l-1, l+1 : L], P, 1), \frac{1}{4} \log_2(1 + P(\mathbf{h}_{\mathcal{B}}^T \mathbf{1})^2) \right\}. \quad (3.100)$$

Proof. The encoding scheme is simple, so I only sketch the proof. Divide transmission into two equal time blocks. At the first block, each transmitter encodes and broadcasts its message using a random Gaussian codebook of power P . The transmitters in \mathcal{B} decode the incoming messages using typical sequence decoding. This is nothing more than a Gaussian multiple-access channel, so decoding is successful as long as the rate is below the first term in (3.100). The multiple-access rate is cut in half due to time sharing.

At the second block, the transmitters in \mathcal{B} directly encode and broadcast the linear combination desired at the receiver, again using a random Gaussian codebook of power P . The receiver decodes the desired function from the signal received in the second block only. This is equivalent to a MISO channel with equal beamformer weights, so decoding is successful as long as the rate is below the second term in (3.100). As before the MISO rate is cut in half due to time sharing. \square

Proof of Theorem 3.7. Achievability is based on the strategy from Lemma 3.2. The achievable rate depends on \mathcal{B} , which may vary according to the channel realizations,

so the overall rate is

$$R(\mathbf{H}, \mathbf{G}, P) = \max_{\mathcal{B}} \min \left\{ \min_{l \in \mathcal{B}} \frac{1}{2} C_{\text{mac}}(\mathbf{g}_l[1 : l-1, l+1 : L], P, 1), \frac{1}{4} \log_2(1 + P(\mathbf{h}_{\mathcal{B}}^T \mathbf{1})^2) \right\} \quad (3.101)$$

$$\geq \max_{\mathcal{B}} \min \left\{ \min_{l \in \mathcal{B}} \frac{1}{2} C_{\text{mac}}(\mathbf{g}_l[1 : l-1, l+1 : L], P, 1), \frac{1}{4} \log_2(1 + P(\|\mathbf{h}_{\mathcal{B}}\|^2)) \right\}. \quad (3.102)$$

Let each rate term in (3.102) be denoted by $R_{\mathcal{B}}(\mathbf{H}, \mathbf{G}, P)$. Then define the event in which a particular cooperation modality fails:

$$\mathcal{O}_{\mathcal{B}} = \left\{ R_{\mathcal{B}}(\mathbf{H}, \mathbf{G}, P) < \frac{r}{2} \log(P) \right\}. \quad (3.103)$$

Outage occurs when each cooperation modality fails simultaneously:

$$\mathcal{O} = \bigcap_{\mathcal{B}} \mathcal{O}_{\mathcal{B}} \quad (3.104)$$

$$\subset \bigcap_{l=1}^L \mathcal{O}_{\{l\}}. \quad (3.105)$$

As suggested by the latter inclusion, it is sufficient to consider only the events in which a single transmitter decodes the messages. Each term in (3.102) has two components, the failure of either of which results in the failure of the cooperation modality. Therefore, define two events: \mathcal{C}_l , the event that transmitter l fails to decode the other transmitters' messages, and \mathcal{N}_l , the event that, even if transmitter l decodes successfully, the receiver fails to decode the linear function. The first event

can be expressed as

$$\mathcal{C}_l = \left\{ \frac{1}{2} C_{\text{mac}}(\mathbf{g}_l, P, 1) < \frac{r}{2} \log(P) \right\} \quad (3.106)$$

$$= \bigcup_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{l\}} \left\{ \frac{1}{4|\mathcal{L}|} \log_2 \left(1 + P \sum_{l' \in \mathcal{L}} g_{l'l}^2 \right) < \frac{r}{2} \log(P) \right\} \quad (3.107)$$

$$\approx \bigcup_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{l\}} \left\{ \sum_{l' \in \mathcal{L}} g_{l'l}^2 < P^{2|\mathcal{L}|r-1} \right\} \quad (3.108)$$

$$\subset \bigcup_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{l\}} \left\{ \bigcap_{l' \in \mathcal{L}} \{g_{l'l}^2 < P^{2|\mathcal{L}|r-1}\} \right\}. \quad (3.109)$$

The second event can be expressed as

$$\mathcal{N}_l = \left\{ \frac{1}{4} \log_2 (1 + P h_l^2) < \frac{r}{2} \log_2(P) \right\} \quad (3.110)$$

$$\approx \{h_l^2 < P^{2r-1}\}. \quad (3.111)$$

Since each cooperation modality involves a different set of channel coefficients, the failure events \mathcal{O}_l are independent. Therefore the outage probability is bounded by

$$\Pr(\mathcal{O}) \leq \prod_{l=1}^L \Pr(\mathcal{C}_l \cup \mathcal{N}_l) \quad (3.112)$$

$$\lesssim \prod_{l=1}^L \left(\sum_{\mathcal{L}} \prod_{l' \in \mathcal{L}} \Pr(g_{l'l}^2 < P^{2|\mathcal{L}|r-1}) + \Pr(h_l^2 < P^{2r-1}) \right) \quad (3.113)$$

$$\approx \left(\sum_{|\mathcal{L}|=1}^{L-1} (P^{2|\mathcal{L}|r-1})^{|\mathcal{L}|} + P^{2r-1} \right)^L \quad (3.114)$$

$$\approx \left(\max_{1 \leq |\mathcal{L}| \leq L-1} P^{|\mathcal{L}|(2|\mathcal{L}|r-1)} + P^{2r-1} \right)^L. \quad (3.115)$$

To find the terms with the largest error exponent, one must maximize the quadratics in (3.115) over $|\mathcal{L}|$. Since the quadratics in question are positive, the maximizer is

either $|\mathcal{L}| = 1$ or $|\mathcal{L}| = L - 1$. This yields

$$\Pr(\mathcal{O}) \lesssim (\max \{P^{(L-1)(2(L-1)r-1)}, P^{2r-1}\} + P^{2r-1})^L \quad (3.116)$$

$$\approx (\max \{P^{(L-1)(2(L-1)r-1)}, P^{2r-1}\})^L. \quad (3.117)$$

Finally, plugging (3.117) into the definition of the DMT yields

$$d^*(r) = \lim_{P \rightarrow \infty} \frac{\log(\Pr(\mathcal{O}))}{\log(P)} \quad (3.118)$$

$$\geq L \min\{1 - 2r, (L - 1)(1 - 2(L - 1)r)\}. \quad (3.119)$$

□

The second bound is derived using the cooperative computation strategy of Theorem 3.3. Since aligning equivalent channels to the desired functions is infeasible, \mathbf{v}_0 is taken to be a constant. Transmitters balance transmit power between sending fresh information, which helps transmitters decode others' messages, and sending resolution information, which helps the receiver decode the desired linear combination. Choosing the balance properly, the benefits of cooperation outweigh the Cauchy-Schwarz penalty, but only enough to obtain a diversity gain of approximately 1/2 per transmitter. Nevertheless, for higher multiplexing gains lattice coding outperforms the strategy of Theorem 3.7.

Theorem 3.8. *For the cooperative computation channel, the following diversity-multiplexing tradeoff is achievable:*

$$d_{\text{lattice}}^-(r) = 1 - r + \min\{[1 - 2r]^+, [(L - 1)(1 - rL)]^+\} + \max_{0 \leq x \leq 1} (L - 2) \min\{[1 - x - r]^+, [(L - 1)(1 - (L - 1)r - x)]^+, [x - r]^+\}. \quad (3.120)$$

Here, $d_{\text{lattice}}^-(0) = 2 + \frac{L-2}{2}$.

Proof. The proof follows a similar outline to that of Theorem 3.7, except that it depends on the rates proved in Theorem 3.3 using lattice block Markov encoding. Again the subset of cooperating users \mathcal{B} varies according to the channel realizations, and the receiver chooses $\mathbf{a} = \mathbf{1}$, resulting in the following achievable rate

$$R(\mathbf{H}, \mathbf{G}, P) = \max_{\mathcal{B}} \min \left\{ \min_{l \in \mathcal{B}} C_{\text{mac}}(\mathbf{g}_l[1 : l-1, l+1 : L] \circ \mathbf{v}_0[1 : l-1, l+1 : L], P, 1), \left[\frac{1}{2} \log_2(1 + P(\|\mathbf{h} \circ \mathbf{v}_0\|^2 + |\mathbf{h}^T \mathbf{v}_1|^2)) - \frac{1}{2} \log_2(L + P(L\|\mathbf{h} \circ \mathbf{v}_0\|^2 - |\mathbf{1}^T(\mathbf{h} \circ \mathbf{v}_0)|^2)) \right]^+ \right\}. \quad (3.121)$$

Similar to before, let each term in (3.121) be denoted by $R_{\mathcal{B}}(\mathbf{H}, \mathbf{G}, P)$ and define the events corresponding to the failure of each cooperation modality:

$$\mathcal{O}_{\mathcal{B}} = \left\{ R_{\mathcal{B}}(\mathbf{H}, \mathbf{G}, P) < \frac{r}{2} \log(P) \right\}. \quad (3.122)$$

Outage occurs when each cooperation modality fails simultaneously:

$$\mathcal{O} = \bigcap_{\mathcal{B}} \mathcal{O}_{\mathcal{B}} \quad (3.123)$$

$$\subset \mathcal{O}_{\{1, \dots, L\}} \cap \bigcap_{l=1}^L \mathcal{O}_{\{l\}} \cap \mathcal{O}_{\emptyset}. \quad (3.124)$$

It is sufficient to consider the events in which *all* transmitters cooperate, in which $L - 2$ individual transmitters cooperate, and in which no one cooperates. When $\mathcal{B} = \emptyset$, transmitters use the strategy outlined in the proof of Theorem 3.4, choosing $v_l^2 = P^{r-1}/h_l^2$. Following that line of analysis, the non-cooperative modality fails only

when any channel gain is too low:

$$\mathcal{O}_\emptyset \subset \left\{ \bigcup_{l=1}^L h_l < P^{r-1} \right\}. \quad (3.125)$$

For $\mathcal{B} \neq \emptyset$, transmitters choose $v_{l0} = P^{-x_{\mathcal{B}}/2}$ for every l , and $v_{l1} = \sqrt{1 - P^{-x_{\mathcal{B}}}}$ for every $l \in \mathcal{B}$; otherwise $v_{1l} = 0$. Using this, the rate is bounded as follows:

$$R_{\mathcal{B}}(\mathbf{H}, \mathbf{G}, P) \geq \min \left\{ \min_{l \in \mathcal{B}} C_{\text{mac}}(\mathbf{g}_l[1 : l-1, l+1 : L], P^{1-x_{\mathcal{B}}}, 1), \right. \\ \left. \left[\frac{1}{2} \log_2 \left(1 + P \sum_{l \in \mathcal{B}} |h_l|^2 \right) - \frac{1}{2} \log_2 (L + P^{1-x_{\mathcal{B}}} (L \|\mathbf{h}\|^2 - \|\mathbf{h}\|^2)) \right]^+ \right\}. \quad (3.126)$$

For large P , this becomes

$$R_{\mathcal{B}}(\mathbf{H}, \mathbf{G}, P) \geq \min \left\{ \min_{l \in \mathcal{B}} C_{\text{mac}}(\mathbf{g}_l[1 : l-1, l+1 : L], P^{1-x_{\mathcal{B}}}, 1), \right. \\ \left. \frac{1}{2} \log_2 \left(\frac{P^{x_{\mathcal{B}}} \sum_{l \in \mathcal{B}} |h_l|^2}{(L-1) \|\mathbf{h}\|^2} \right) \right\}. \quad (3.127)$$

As before, define events corresponding to the failure of either term in (3.127): $\mathcal{C}_{\mathcal{B}}$, the event that the transmitters in \mathcal{B} fail to decode the other transmitters' messages, and $\mathcal{N}_{\mathcal{B}}$, the event that, even if the transmitters decode each other properly, the receiver fails to decode its linear function at the required rate. The first event can be expressed

as

$$\mathcal{C}_{\mathcal{B}} = \bigcup_{l \in \mathcal{B}} \left\{ C_{\text{mac}}(\mathbf{g}_l[1 : l-1, l+1 : L], P^{1-x_{\mathcal{B}}}, 1) < \frac{r}{2} \log(P) \right\} \quad (3.128)$$

$$= \bigcup_{l \in \mathcal{B}} \bigcup_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{l\}} \left\{ \frac{1}{2|\mathcal{L}|} \log_2 \left(1 + P^{1-x_{\mathcal{B}}} \sum_{l' \in \mathcal{L}} g_{l'l}^2 \right) < \frac{r}{2} \log(P) \right\} \quad (3.129)$$

$$\approx \bigcup_{l \in \mathcal{B}} \bigcup_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{l\}} \left\{ \sum_{l' \in \mathcal{L}} g_{l'l}^2 < P^{|\mathcal{L}|r+x_{\mathcal{B}}-1} \right\} \quad (3.130)$$

$$\subset \bigcup_{l \in \mathcal{B}} \bigcup_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{l\}} \left\{ \bigcap_{l' \in \mathcal{L}} \{g_{l'l}^2 < P^{|\mathcal{L}|r+x_{\mathcal{B}}-1}\} \right\}. \quad (3.131)$$

For $\mathcal{B} = \{1, \dots, L\}$, the second event can be expressed as

$$\mathcal{N}_{\{1, \dots, L\}} = \left\{ \frac{1}{2} \log_2 \left(\frac{P^{x_{\mathcal{B}}} \|\mathbf{h}\|^2}{(L-1) \|\mathbf{h}\|^2} \right) < \frac{r}{2} \log_2(P) \right\} \quad (3.132)$$

$$= \{P^{x_{\{1, \dots, L\}}} < (L-1)P^r\}. \quad (3.133)$$

Based on (3.133), choose $x_{\{1, \dots, L\}} = r - \epsilon$ for any $\epsilon > 0$. As $P \rightarrow \infty$, this forces

$\mathcal{N}_{\{1, \dots, L\}} \rightarrow \emptyset$ deterministically. For $\mathcal{B} = \{l\}$, second event can be written as

$$\mathcal{N}_{\{l\}} = \left\{ \frac{1}{2} \log_2 \left(\frac{P^{x_{\{l\}}} h_l^2}{(L-1) \|\mathbf{h}\|^2} \right) < \frac{r}{2} \log_2(P) \right\} \quad (3.134)$$

$$= \left\{ \frac{h_l^2}{(L-1) \|\mathbf{h}\|^2} < P^{r-x_{\{l\}}} \right\} \quad (3.135)$$

$$\subset \{h_l^2 < P^{r-x_{\{l\}}-\epsilon}\} \cup \left\{ \|\mathbf{h}\|^2 \geq \frac{P^\epsilon}{L-1} \right\} \quad (3.136)$$

$$\subset \bigcap_{l \in \mathcal{B}} \{h_l^2 < P^{r-x_{\{l\}}-\epsilon}\} \cup \left\{ \|\mathbf{h}\|^2 \geq \frac{P^\epsilon}{L-1} \right\}. \quad (3.137)$$

Combining the above with (3.124) yields

$$\begin{aligned} \mathcal{O} \subset & \left[\left(\bigcup_{l \in \{1, \dots, L\}} \bigcup_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{l\}} \bigcap_{l' \in \mathcal{L}} \{g_{l'l}^2 < P^{|\mathcal{L}|r+x+\epsilon-1}\} \right) \right] \cap \\ & \bigcap_{l \in \{1, \dots, L\}} \left[\left(\bigcup_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{l\}} \bigcap_{l' \in \mathcal{L}} \{g_{l'l}^2 < P^{|\mathcal{L}|r+x_{\{l\}}-1}\} \right) \cup (\{h_l^2 < P^{r-x_{\{l\}}-\epsilon}\}) \right] \cap \\ & \left\{ \bigcup_{l=1}^L h_l^2 < P^{r-1} \right\} \cup \left\{ \|\mathbf{h}\|^2 \geq \frac{P^\epsilon}{L-1} \right\}. \quad (3.138) \end{aligned}$$

Equation (3.138) contains too many terms to enumerate in full. Since only asymptotic behavior affects the DMT, it suffices to examine only the term with the highest exponent. This term contains one channel failure in $\mathcal{C}_{\{1, \dots, L\}}$, $L-2$ failures in $\mathcal{C}_{\{l\}} \cap \mathcal{N}_{\{l\}}$, and one failure in \mathcal{N}_\emptyset . The final error event, in which $\|\mathbf{h}\|^2$ is too large, has negligible contribution to the error probability. Combining these yields

$$\begin{aligned} \Pr(\mathcal{O}) \lesssim & \Pr \left(\bigcup_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{1\}} \bigcap_{l' \in \mathcal{L}} \{g_{l'l}^2 < P^{|\mathcal{L}|r+x+\epsilon-1}\} \right) \Pr(\{h_1^2 < P^{r-1}\}) \times \\ & \prod_{l=3}^L \Pr \left(\bigcup_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{l\}} \bigcap_{l' \in \mathcal{L}} \{g_{l'l}^2 < P^{|\mathcal{L}|r+x_{\{l\}}-1}\} \cup \{|h_l|^2 < P^{r-x_{\{l\}}-\epsilon}\} \right). \quad (3.139) \end{aligned}$$

Since each term in (3.139) is independent, the probabilities evaluate separately, yielding

$$\Pr(\mathcal{O}) \lesssim \left(\sum_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{1\}} (P^{|\mathcal{L}|r+x+\epsilon-1})^{|\mathcal{L}|} \right) (P^{r-1}) \left(\sum_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{3\}} (P^{|\mathcal{L}|r-x-1})^{|\mathcal{L}|} + P^{r-x-\epsilon} \right)^{L-2} \quad (3.140)$$

$$= P^{r-1} \left(\sum_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{1\}} (P^{|\mathcal{L}|(|\mathcal{L}|r+x+\epsilon-1)}) \right) \left(\sum_{\mathcal{L} \subset \{1, \dots, L\} \setminus \{3\}} (P^{|\mathcal{L}|(|\mathcal{L}|r-x-1)}) + P^{r-x-\epsilon} \right)^{L-2}, \quad (3.141)$$

where $x_{\{l\}} = x$ for every l . Similar to the proof Theorem 3.7, the maximizer of the quadratics in (3.141) is either $|\mathcal{L}| = 1$ or $|\mathcal{L}| = L - 1$. This yields

$$\Pr(\mathcal{O}) \lesssim P^{r-1} \left(\max \left\{ P^{2r-1+\epsilon}, P^{(1-L)(1-rL)+(L-1)\epsilon} \right\} \right) \times \left(\max_x \min \left\{ P^{r+x-1}, P^{(1-L)(1-(L-1)r-x)}, P^{r-x-\epsilon} \right\} \right)^{L-2}. \quad (3.142)$$

Finally, plugging (3.142) into the definition of the DMT, taking the supremum over all $\epsilon > 0$, and taking the maximum over all x yields

$$\begin{aligned} d_c^-(r) &= \lim_{P \rightarrow \infty} \frac{\log(\Pr(\mathcal{O}))}{\log(P)} \\ &\geq 1 - r + \min\{[1 - 2r]^+, [(L - 1)(1 - rL)]^+\} + \\ &\quad \max_{0 \leq x \leq 1} (L - 2) \min\{[1 - x - r]^+, [(L - 1)(1 - (L - 1)r - x)]^+, [x - r]^+\}. \end{aligned} \quad (3.143)$$

$$(3.144)$$

□

Figure 3.2 shows the DMT regions. For $L = 2$ lattice coding is sufficient to achieve the full DMT. For $L > 2$, lattice coding achieves better performance only for sufficiently high multiplexing gain. Random coding fails altogether at multiplexing gains higher than $(L - 1)/2$ due to the need for transmitters to decode $L - 1$ separate messages and the need for time-sharing. Lattice coding, on the other hand, maintains non-zero diversity for every $0 \leq r < 1$. Between the two strategies, the corner points of the DMT region are achieved.

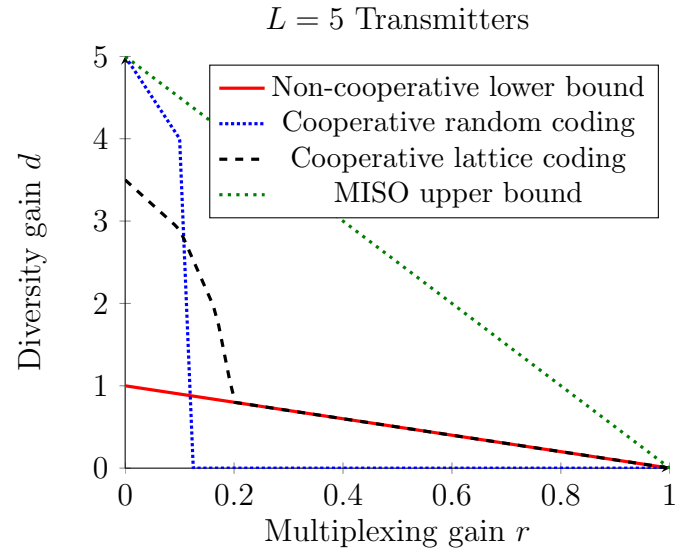
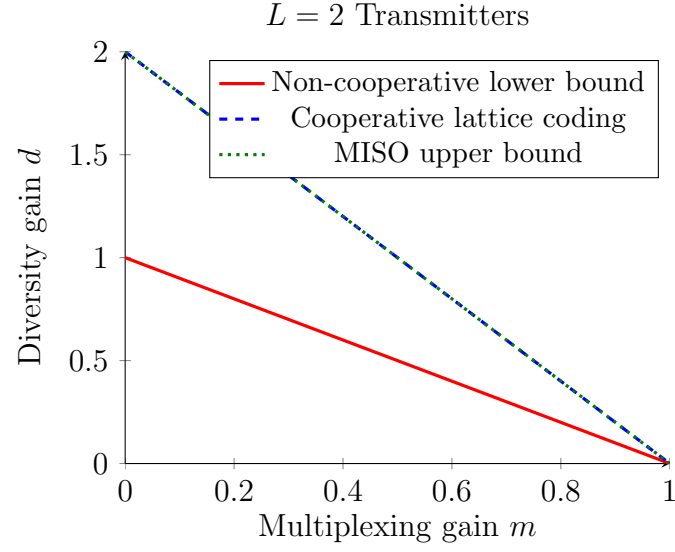


Figure 3.2: Diversity-multiplexing tradeoff for $L = 2$, $L = 5$ transmitters and a single receiver.

3.4 Numerical Examples

In this section I examine a few example scenarios in which to demonstrate the benefits of user cooperation. The first example, depicted in Figure 3.3, comprises $L = 2$ transmitters and a single receiver. The channels are symmetric, with the forward coefficients constant $h_1 = h_2 = 1$ and the inter-transmitter coefficients a variable $g_{12} = g_{21} = g$, which vary such that the gain g^2 ranges between -10dB and 30dB . I set the transmit SNR at $P = 10\text{dB}$. Since the channel gains are symmetric, either both transmitters can decode the other's message or neither of them can; therefore I choose either $\mathcal{B} = \{1, 2\}$ or $\mathcal{B} = \emptyset$ for cooperative computation. Similarly, by symmetry it is easy to see that the optimal choice for the linear function is $\mathbf{a} = (1, 1)^T$ and that the optimal steering vectors \mathbf{v}_0 and \mathbf{v}_1 are constant. I find the optimal tradeoff between \mathbf{v}_0 and \mathbf{v}_1 numerically.

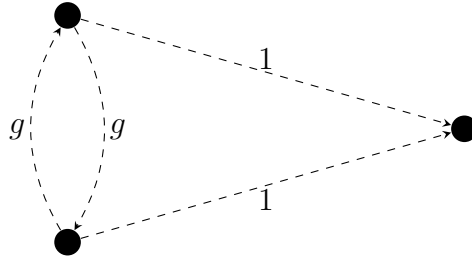


Figure 3.3: A two-by-one computation network with symmetric channel gains.

Figure 3.4 shows the achievable rate of the cooperative scheme against the upper bound of Theorem 3.2, using the Nazer-Gastpar rate of (3.8) as a baseline. The trends are easy to appreciate. When the channels between transmitters are weak, decoding each other's messages is too difficult, and the cooperative rate collapses to (3.8). As the inter-transmitter gains become stronger it becomes easier for the transmitters to decode, and cooperation can improve the computation rate and eventually approaches the upper bound. Note the “dimple” in the cooperative rate as g^2 becomes large. For sufficiently large g^2 , the optimal strategy is to turn the steering vector \mathbf{v}_0

down far enough that the Nazer-Gastpar component of the cooperative rate is zero, meaning that only the jointly-encoded resolution information carries information to the receiver. At this value of g^2 there arises a dimple, after which the rate quickly converges on the upper bound.

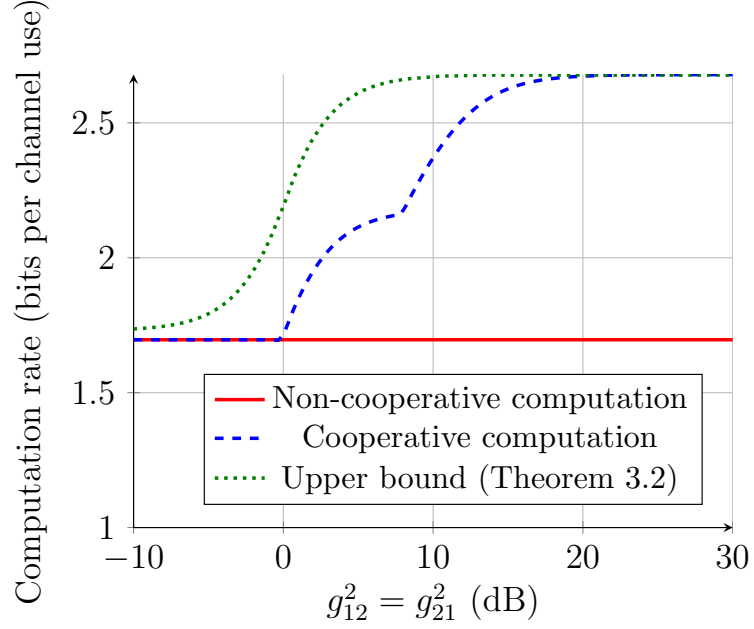


Figure 3.4: Achievable rates as a function of inter-transmitter channel gains

Next I examine a scenario in which channel gains are chosen randomly, as depicted in Figure 3.5. I place a single receiver at the origin and place $L = 3$ transmitters randomly and uniformly on a segment of the circle having specified arclength. From the geometric configuration of the network, I compute channel magnitudes according to a path-loss model:

$$g_{ij} = \sqrt{\frac{1}{d(i,j)^\alpha}}, \quad h_i = \sqrt{\frac{1}{d(i,0)^\alpha}},$$

where $d(i,j)$ is the Euclidean distance between users i and j and. I choose $P = 10\text{dB}$ and a path-loss exponent of $\alpha = 4$.

For each realization I calculate the cooperative computation rate. Since the gains

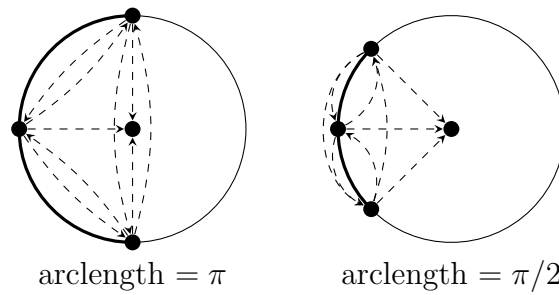


Figure 3.5: Three users are placed along a segment of the unit circle, while the receiver is placed at the origin.

from transmitters to receiver are equal, $\mathbf{a} = (1, 1, 1)^T$ is the optimal choice. The steering vectors and the clusters are optimized numerically. I run 500 simulations each for arclengths varying from 0 to π , and plot the average computation rates in Figure 3.6. Again the trends are easy to appreciate. Cooperation offers the greatest improvement when transmitters are close together. Even as transmitters spread further apart, on average enough transmitters can cooperate that cooperation garners a noticeable improvement.

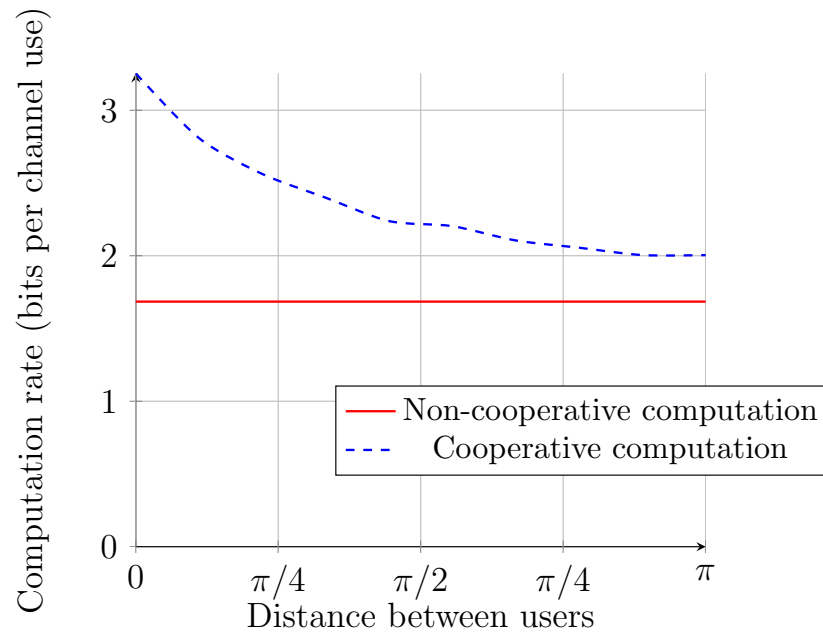


Figure 3.6: Average computation rate vs. angle between transmitters.

In the next example I examine the variation in cooperative computation rate with respect to the channel gain between transmitters and receivers. As depicted in Figure 3.7, I again take $L = 2$ and $M = 1$, but now every channel gain is unity except for h_{12} . Since the channels between transmitters and receiver are not symmetric, I cannot take $\mathbf{a} = (1, 1)^T$ or \mathbf{v}_0 and \mathbf{v}_1 to be constant. Instead, I iterate manually through possible choices of \mathbf{a} and numerically optimize over the set \mathcal{B} of cooperating transmitters and the steering vectors \mathbf{v}_0 and \mathbf{v}_1 .

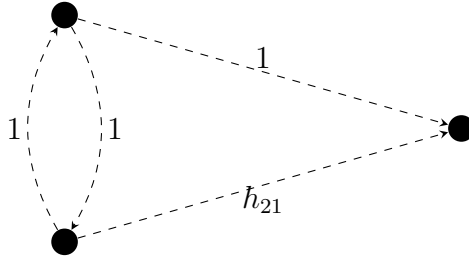


Figure 3.7: A two-by-one computation network with asymmetric channel gains.

In Figure 3.8 I plot the cooperative rate alongside (3.8) for a variety of transmit signal-to-noise ratios P . I make a few observations. First, the non-cooperative rate is low for h_{21} near to zero. Since functions must contain elements from both transmitters' messages, it becomes difficult for the receiver to decode such a function. In the cooperative case, however, the rates do not fall, since transmitter 1 can decode \mathbf{w}_2 and transmit the desired function to the receiver. This result hints at the diversity gains inherent to the cooperative approach; even when one link fails, successful computation is possible.

Furthermore, cooperation achieves the full multiplexing gain as the SNR becomes large. Non-cooperative computation results in “peaks” which correspond to rational channel gains with low denominator. The further h_{21} is from a low-denominator rational, the harder it is to align the function with the channels and the higher the Cauchy-Schwarz penalty in (3.8). However, one can always choose \mathbf{v}_0 such that

the equivalent channel vector is rational, allowing receivers to eliminate completely the Cauchy-Schwarz penalty. Note that this is not explicitly due to the cooperative nature of the approach; as shown in 3.4 non-cooperative transmitters can get the full multiplexing gain using lattice codes. However, cooperation *does* permit the transmitters to use the remaining power to secure rate and diversity gains.

Finally, I examine the system depicted in Figure 3.9. Here $L = M = 2$, and again all channel gains are unity except for h_{21} . Again asymmetry prevents an easy choice of \mathbf{a} and the steering vectors. I iterate manually over the possible choices for \mathbf{a} , choose zero-forcing beamformers for \mathbf{v}_1 and \mathbf{v}_2 , and numerically optimize over \mathbf{v}_0 . In order for zero-forcing to succeed, I choose $\mathcal{B} = \{1, 2\}$.

Figure 3.10 shows the cooperative rate alongside (3.8), again for a variety of signal-to-noise ratios. In contrast to the previous scenario, here the rate drops when $h_{12} \approx 1$; this is because the channel matrix becomes increasingly ill-conditioned. Similar to before, in the cooperative case the rate remains non-zero, but here it occurs because the transmitters can cooperatively send a full-rank set of equations even though the channel matrix is nearly singular. However, in this example cooperation does not obtain the full multiplexing gain. The freedom to choose \mathbf{v}_0 allows the receivers to mitigate the peakiness of the achievable rate, but they cannot eliminate the Cauchy-Schwarz penalty at both receivers simultaneously. Even for high SNR, however, there remains considerable robustness to channel variation.

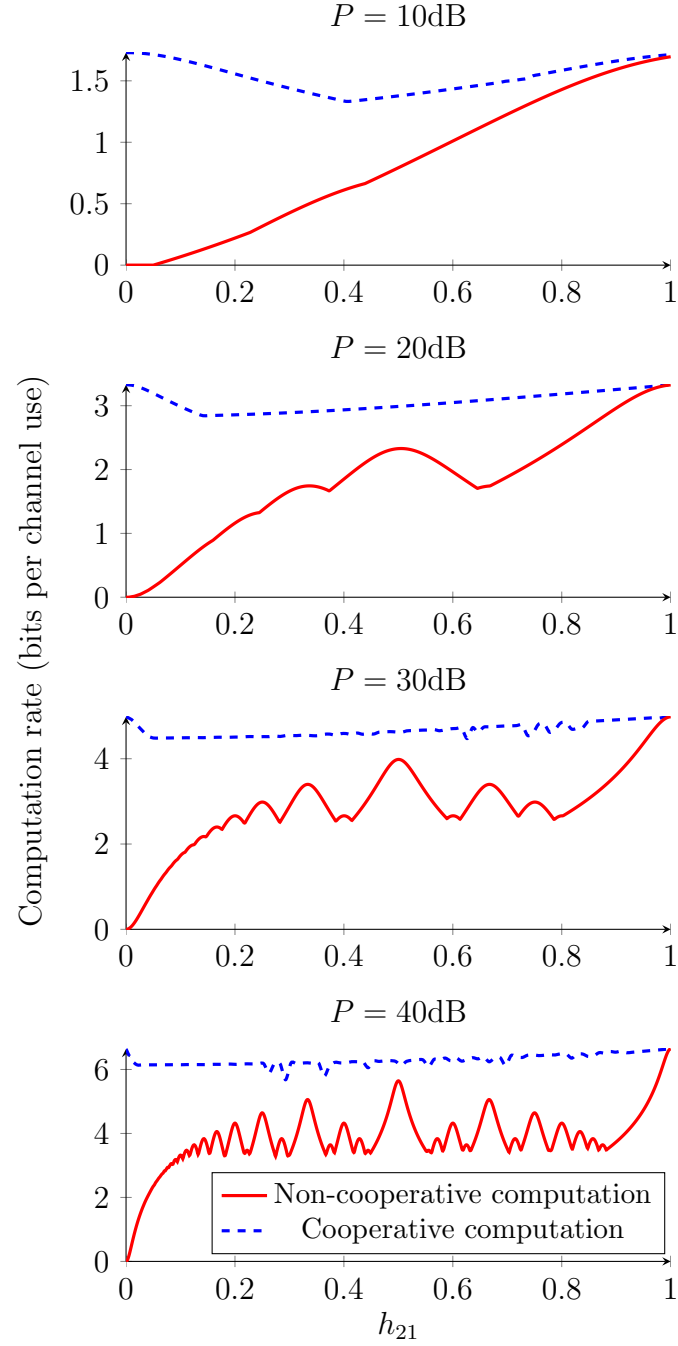


Figure 3.8: Achievable rates as a function of h_{21} and P .

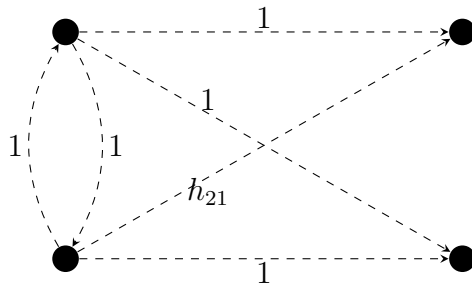


Figure 3.9: A two-by-two computation network with asymmetric channel gains.

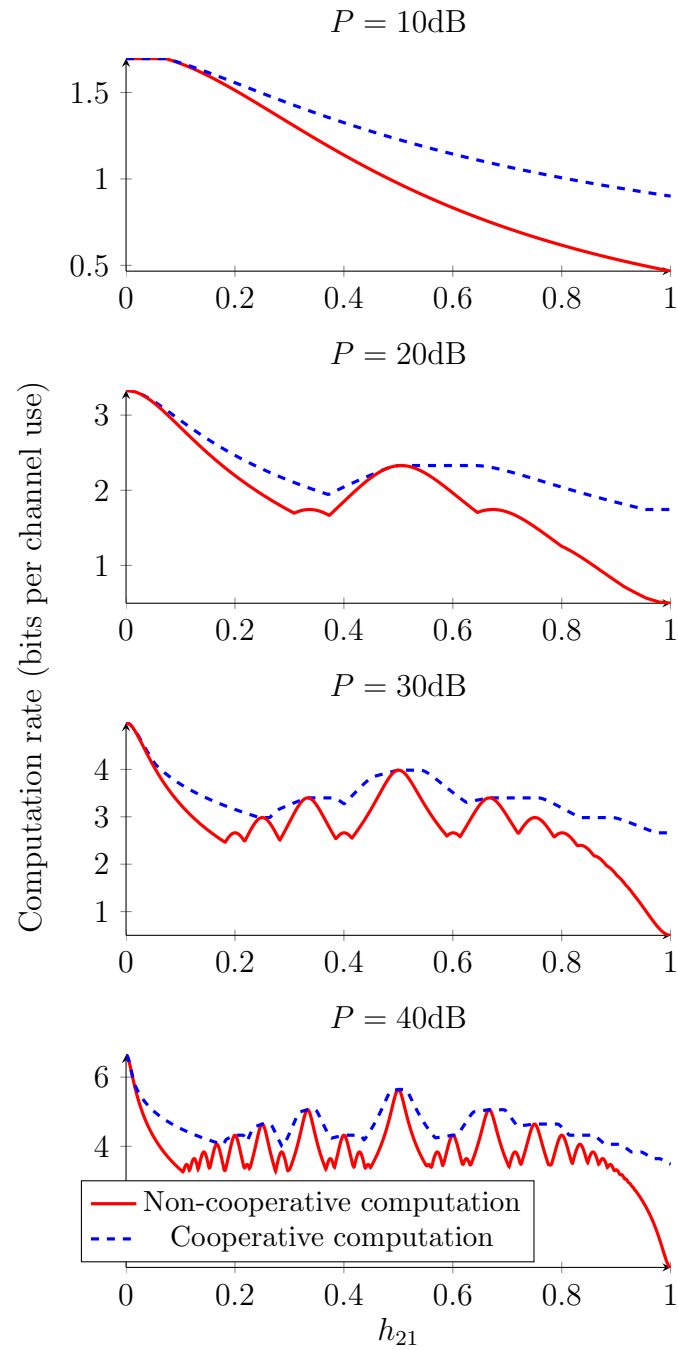


Figure 3.10: Achievable rates as a function of h_{21} and P .

Relay Cooperation

In this chapter, I explore the benefits of relay cooperation to physical-layer network coding. I examine a two-transmitter, two-receiver system aided by a dedicated relay. I derive results for two relay modalities: the *standard* relay, in which the relay's transmission depends only on signals received in previous time slots, and the *instantaneous* relay, or “relay-without-delay,” in which the relay's transmission may depend on signals received in the current time slot. For the standard relay, a compress-and-forward strategy improves the achievable computation rate, but it comes with no guarantees of even approximate optimality. However, for the instantaneous relay, an amplify-and-forward strategy is optimal in the degrees-of-freedom, and for symmetric channel gains it achieves computation rates within a constant gap of capacity.

4.1 Preliminaries

Here I study the *relay computation channel*, depicted in Figure 4.1. It consists of two sources, two destinations, and a single dedicated relay. The relay operates under one of two relay modalities: the *standard* modality, in which the relay's transmission is a function only of signals received in previous symbol times, and the *instantaneous*

modality, in which the relays' transmission may be a function of signals received during the current symbol time. While the distinction may seem small, the resulting system models are sufficiently different that I describe them separately.

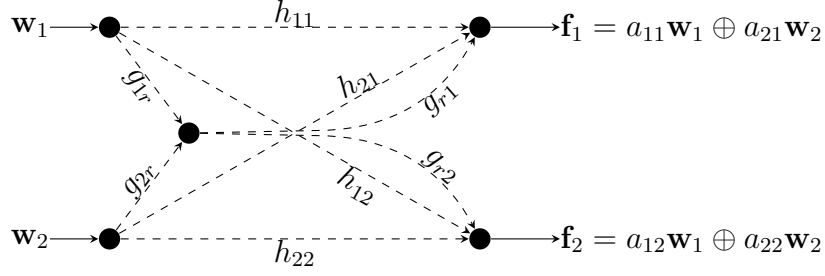


Figure 4.1: The relay computation channel.

4.1.1 System Model: Standard Relay

For the standard relay, as in the previous chapter, I will develop a block Markov strategy. Therefore I again divide transmission into $B + 1$ blocks of n channel uses each. Each source $l \in \{1, 2\}$ has B messages $\mathbf{w}_l[b] \in \mathbb{F}_p^k$. Structurally, this channel resembles the two-user relay-interference channel. But, as before, each receiver intends to decode a finite-field linear combination of the transmitters' messages instead of the messages themselves:

$$\mathbf{f}_m[b] = a_{1m} \odot \mathbf{w}_1[b] \oplus a_{2m} \odot \mathbf{w}_2[b], \quad (4.1)$$

for $a_{lm} \in \mathbb{Z}$. Again the function coefficients are represented by the 2×2 matrix $\mathbf{A} = [a_{lm}]$. The matrix \mathbf{A} must be full rank.

At block b , each transmitter l sends a signal $\mathbf{x}_l[b] \in \mathbb{R}^n$, subject to the average power constraint

$$\frac{1}{n} \|\mathbf{x}_l[b]\|^2 \leq P. \quad (4.2)$$

The relay receives the signal

$$\mathbf{y}_r[b] = g_{1r}\mathbf{x}_1[b] + g_{2r}\mathbf{x}_2[b] + \mathbf{z}_r[b], \quad (4.3)$$

where g_{1r} and g_{2r} are fixed and globally-known channel gains, and $\mathbf{z}_r[b]$ is additive white Gaussian noise having unit variance.

Regardless of relay modality, the relay is assumed to be capable of full-duplex transmission. The relay sends a signal $\mathbf{x}_r[b]$, subject to the same power constraint P . Each receiver $m \in \{1, 2\}$ obtains the signal

$$\mathbf{y}_m[b] = h_{1m}\mathbf{x}_1[b] + h_{2m}\mathbf{x}_2[b] + g_{rm}\mathbf{x}_r[b] + \mathbf{z}_m[b], \quad (4.4)$$

where again \mathbf{z}_j is unit-variance AWGN, and again channel coefficients are known globally.

The *computation capacity* is defined in a manner similar to that of Chapter 3. Each transmitter l employs an encoder $\mathcal{E}_l : \mathbb{F}_p^{k \times B} \rightarrow \mathbb{R}^{n \times B+1}$, which maps the messages $\mathbf{w}_l[b]$ to the signals $\mathbf{x}_l[b]$. The relay employs a decoder $\mathcal{E}_r : \mathbb{R}^{n \times B} \rightarrow \mathbb{R}^{n \times B+1}$ mapping received signals $\mathbf{y}_r[b]$ to transmit signals $\mathbf{x}_r[b]$, with the caveat that $\mathbf{x}_r[b]$ can only depend on $\mathbf{y}_r[s]$ for $s < b$.

Each receiver m employs a decoder $\mathcal{D}_m : \mathbb{R}^{n \times B+1} \rightarrow \mathbb{F}_p^{k \times B}$ mapping received signals $\mathbf{y}_m[b]$ to function estimates $\hat{\mathbf{w}}_m[b]$. The rate¹ of each encoder, in bits per channel use, is

$$R = \frac{Bk \log_2(p)}{(B+1)n}. \quad (4.5)$$

¹Note that, unlike in Chapter 3, each transmitter may encode at a different rate.

Define the probability of error

$$P_e = \Pr \left(\bigcup_{b=1}^B \{\hat{\mathbf{f}}_1[b] \neq \mathbf{f}_1[b]\} \cup \{\hat{\mathbf{f}}_2[b] \neq \mathbf{f}_2[b]\} \right) \quad (4.6)$$

A rate R is said to be *achievable* if, for any $\delta > 0$ and $\epsilon > 0$, there exist encoders \mathcal{E}_1 and \mathcal{E}_2 having rates larger than $R_1 - \delta$ and $R_2 - \delta$, respectively, along with a relaying function and decoders \mathcal{D}_1 and \mathcal{D}_2 , for which $P_e < \epsilon$. The *computation capacity* is the supremum of all achievable rates.

4.1.2 System Model: Instantaneous Relay

With an instantaneous relay, I will propose a simple amplify-and-forward strategy for which block Markov encoding is not required. Communication therefore simply occurs over n channel uses, rather than $B + 1$ blocks thereof. Each source $l \in \{1, 2\}$ has a single message $\mathbf{w}_l \in \mathbb{F}_p^k$, and each receiver intends to decode a single finite-field linear combination thereof:

$$\mathbf{f}_m = a_{1m} \odot \mathbf{w}_1 \oplus a_{2m} \odot \mathbf{w}_2, \quad (4.7)$$

for $a_{lm} \in \mathbb{Z}$, where again the matrix of coefficient \mathbf{A} must be full rank.

Each transmitter l sends a signal \mathbf{x}_l , which must conform to the average power constraint P . The relay obtains the signal

$$\mathbf{y}_r = g_{1r}\mathbf{x}_1 + g_{2r}\mathbf{x}_2 + \mathbf{z}_r, \quad (4.8)$$

where again g_{lr} are fixed channel gains and \mathbf{z}_r is unit-variance AWGN. The relay transmits a signal \mathbf{x}_r , which also obeys the power constraint P . Each receiver m

obtains the signal

$$\mathbf{y}_m = h_{1m}\mathbf{x}_1 + h_{2m}\mathbf{x}_2 + g_{rm}\mathbf{x}_r + \mathbf{z}_m. \quad (4.9)$$

Each transmitter employs an encoder $\mathcal{E}_l : \mathbb{F}_p^k \rightarrow \mathbb{R}^n$ mapping messages \mathbf{w}_l to transmit signals \mathbf{x}_l . The relay employs a decoder $\mathcal{E}_r : \mathbb{R}^n \rightarrow \mathbb{R}^n$ from received signals \mathbf{y}_r to transmit signals \mathbf{x}_r . Since the relay is instantaneous, the i th element of \mathbf{x}_r may depend on the 1st through i th elements of \mathbf{y}_r . Finally, each receiver employs a decoder $\mathcal{D}_m : \mathbb{R}^n \rightarrow \mathbb{F}_p^k$, mapping received signals \mathbf{y}_m to function estimates $\hat{\mathbf{f}}_m$.

In this case, the rate of each encoder is

$$R = \frac{k \log_2(p)}{n}. \quad (4.10)$$

Define the probability of error

$$P_e = \Pr \left(\{\hat{\mathbf{f}}_1 \neq \mathbf{f}_1\} \cup \{\hat{\mathbf{f}}_2 \neq \mathbf{f}_2\} \right) \quad (4.11)$$

Again any rate R is said to be achievable if there exist encoders and decoders sufficiently close to the desired rates with arbitrarily low probability of error, and again define the computation capacity as the supremum over all achievable rates.

For both relay modalities, the non-cooperative approach of Nazer and Gastpar [36] can be trivially applied. Applying (3.8) to either system model, It achieves the following rates:

$$R \leq \min_l \min_{m: a_{lm} \neq 0} \left[\frac{1}{2} \log_2(1 + P \|\mathbf{h}_m\|^2) - \frac{1}{2} \log_2(\|\mathbf{a}_m\|^2 - P(\|\mathbf{h}_m\|^2 \|\mathbf{a}_m\|^2 - \langle \mathbf{h}_m, \mathbf{a}_m \rangle^2)) \right]^+, \quad (4.12)$$

where $\mathbf{h}_m = (h_{1m}, h_{2m})^T$, $\mathbf{a}_m = (a_{1m}, a_{2m})^T$, and where $\mathbf{a}_1, \mathbf{a}_2$ must be linearly inde-

pendent.

4.1.3 Degrees-of-Freedom

In Chapter 3, I studied the diversity-multiplexing tradeoff, which provides an approximate characterization of performance at high signal-to-noise ratios. Here I examine only the multiplexing gain, which when referred to singly is commonly termed the *degrees-of-freedom*. For a scheme that achieves common computation rate R_{scheme} , define the *sum degrees-of-freedom* as

$$\text{DoF} = \lim_{P \rightarrow \infty} \frac{2R_{\text{scheme}}}{\frac{1}{2} \log_2(P)}. \quad (4.13)$$

In other words, the degrees-of-freedom compares the asymptotic achievable sum rate to that of a single point-to-point AWGN channel.

It is straightforward to show that the maximum DoFs of the relay computation channel is two, regardless of relay modality. Indeed, the maximum DoFs can be achieved even without the relay, using the interference alignment scheme presented in [44]. However, schemes that also perform well at low-to-moderate SNR are desired. Therefore, for the remainder of this chapter I examine the extent to which relay cooperation can satisfy both desiderata.

4.2 Standard Relay

Consider first the *standard* relay modality, in which the relay function can depend only on signals received in previous symbol times. I begin by establishing an upper bound on the computation capacity region.

Theorem 4.1. *For the two-user standard relay computation channel, any achievable*

computation rate pair satisfies

$$R \leq \min_l \min_{m: a_{lm} \neq 0} \max_{|\alpha| \leq 1} \min \left\{ \frac{1}{2} \log_2(1 + (1 - \alpha)P(h_{lm}^2 + g_{rm}^2)), \right. \\ \left. \frac{1}{2} \log_2(1 + P(g_{lr}^2 + h_{lm}^2 + 2g_{lr}h_{lm}\sqrt{\alpha})) \right\}, \quad (4.14)$$

for some full-rank set of coefficients a_{lm} .

Proof. The result relies on a genie-aided argument. Suppose a genie supplies \mathbf{w}_2 to both destinations. Then destinations need only to recover \mathbf{w}_1 in order to recover any linear combination. This is equivalent to the relay channel formed by source 1, the relay, and each destination for which $a_{lm} \neq 0$. Bounding the capacity of the relay channel by the cut-set bound [15] yields the result for transmitter 1. Repeating the argument while supplying \mathbf{w}_2 to the destinations establishes the result for transmitter 2. \square

Theorem 4.1 clearly shows an upper bound of two DoFs, which, as mentioned above, can be achieved by the alignment scheme of [44]. To obtain good finite-SNR performance, I propose a compress-and-forward scheme.

Theorem 4.2. *For the relay computation channel with an standard relay, the following rate is achievable:*

$$R_{\text{standard}} = \min_l \min_{m: a_{lm} \neq 0} \left[\frac{1}{2} \log_2 \left(1 + P'_m \|\mathbf{h}'_m\|^2 \right) - \right. \\ \left. \frac{1}{2} \log_2 \left(\|\mathbf{a}_m\|^2 - P'_m (\|\mathbf{h}'_m\|^2 \|\mathbf{a}_m\|^2 - \langle \mathbf{h}'_m, \mathbf{a}_m \rangle^2) \right) \right]^+, \quad (4.15)$$

where

$$P'_m = \frac{P}{1 + \gamma_m(1 + \sigma_r^2)}, \quad (4.16)$$

$$\mathbf{h}'_m = \mathbf{v} \circ (\mathbf{h}_m + \gamma_m \mathbf{g}_r), \quad (4.17)$$

where

$$\sigma_r^2 = \max_{m \in \{1,2\}} \frac{1/P + \|\mathbf{v} \circ \mathbf{h}_m\|^2 + \|\mathbf{g}_r\|^2}{g_{rm}^2} + \frac{P(\|\mathbf{v} \circ \mathbf{h}_m\|^2 \|\mathbf{g}_r\|^2 - \langle \mathbf{v} \circ \mathbf{h}_m, \mathbf{g}_r \rangle^2)}{g_{rm}^2}, \quad (4.18)$$

for any $\gamma_1, \gamma_2 \in \mathbb{R}$ and any \mathbf{v} such that

$$|v_l| \leq 1, l \in \{1, 2\}. \quad (4.19)$$

Proof. I prove this result with an encoding scheme based on via block Markov encoding, in which transmitters communicate B combinations of messages over $B+1$ blocks of n symbol times each. However, unlike the decode-and-forward scheme presented in Chapter 3, here the relay employs compress-and-forward cooperation. During one block the relay overhears the noisy superposition of lattice codewords. It encodes the received signal using Wyner-Ziv binning and transmits the compressed version during the next block. Each destination recovers the compressed relay signal, takes a linear combination of the relay signal and its own received signal, and performs lattice quantization on the result.

Encoding: Each transmitter employs a lattice codebook \mathcal{C} , constructed from nested lattices Λ_c and Λ_s as described in Chapter 2. For each message $\mathbf{w}_l[b]$, let

$$\mathbf{c}_l[b] = [\phi(\mathbf{w}_l[b]) + \mathbf{d}_l[b]] \bmod \Lambda_s \quad (4.20)$$

be the dithered lattice codeword, where $\mathbf{d}_l[b]$ is independent and uniform over \mathcal{V}_s for

every b . Recall that ϕ is the mapping from the finite field to the lattice codebook. At each block $1 \leq b \leq B$, source l transmits a scaled version of its dithered codeword; at block $B + 1$, having no codeword to send, it transmits nothing:

$$\mathbf{x}_l[b] = \begin{cases} v_l \sqrt{P} \mathbf{c}_l[b], & \text{for } 1 \leq b \leq B \\ 0, & \text{for } b = B + 1 \end{cases}. \quad (4.21)$$

Following (4.3), the relay receives the following signal at block b :

$$\mathbf{y}_r[b] = g_{1r} v_1 \sqrt{P} \mathbf{c}_1[b] + g_{2r} v_2 \sqrt{P} \mathbf{c}_2[b] + \mathbf{z}_r[b]. \quad (4.22)$$

To facilitate recovery of linear combinations of messages at the destinations, the relay forwards a compressed version of $\mathbf{y}_r[b]$ by means of Wyner-Ziv coding with side information at the destinations [94]. The relay has a quantization codebook $\mathcal{C}_r^q \subset \mathbb{R}^n$ containing $2^{n\tilde{R}_r}$ codewords drawn randomly and independently from a zero-mean Gaussian distribution having variance $E[\mathbf{y}_r^2[b]] + \sigma_r^2 = P(g_{1r}^2 v_1^2 + g_{2r}^2 v_2^2) + 1 + \sigma_r^2$. The set of codeword indices $[1 : 2^{n\tilde{R}_r}]$ are partitioned into 2^{nR_r} equal-sized bins by the binning function

$$\psi(u) = [(u - 1)2^{n(\tilde{R}_r - R_r)} : m2^{n(\tilde{R}_r - R_r)}], \quad (4.23)$$

for bin index $u \in [1 : 2^{nR_r}]$. The relay also has a standard codebook $\mathcal{C}_r \subset \mathbb{R}^n$, with 2^{nR_r} codewords drawn i.i.d. from a zero-mean Gaussian distribution with variance P . Obviously there exists a one-to-one mapping between bin indices and codewords in \mathcal{C}_r .

To compress $\mathbf{y}_r[b]$, the relay first finds a codeword $\hat{\mathbf{y}}_r[b] \in \mathcal{C}_r^q$ jointly typical with $\mathbf{y}_r[b]$, randomly choosing between them if there is more than one such codeword and

declaring an error if there is no such codeword. Then the relay finds the bin index $u[b]$ such that the bin $\phi(u[b])$ contains the selected codeword. Finally, the relay selects the unique codeword $\mathbf{c}_r[b] \in \mathcal{C}_r$ corresponding to the bin index $b[u]$. The relay transmits $\mathbf{c}_r[b]$ in block $b + 1$:

$$\mathbf{x}_r[b] = \begin{cases} 0, & \text{for } b = 1 \\ \mathbf{c}_r[b - 1], & \text{for } 2 \leq b \leq B + 1 \end{cases}. \quad (4.24)$$

The encoding process is summarized in Table 4.1.

Table 4.1: Block Markov encoding for compress-and-forward

	$b = 1$	$b = 2$	\dots	$b = B + 1$
$\mathbf{x}_1[b]$	$v_1 \mathbf{c}_1[1]$	$v_1 \mathbf{c}_1[2]$	\dots	—
$\mathbf{x}_2[b]$	$v_2 \mathbf{c}_2[1]$	$v_2 \mathbf{c}_2[2]$	\dots	—
$\mathbf{x}_r[b]$	—	$\mathbf{c}_r[1]$	\dots	$\mathbf{c}_r[B - 1]$

Decoding: To decode $\mathbf{f}_m[b]$, destination m first recovers $\hat{y}_r[b]$ from its received signal in block $b + 1$, which by (4.4) is

$$\mathbf{y}_m[b + 1] = h_1 v_1 \sqrt{P} \mathbf{c}_1[b + 1] + h_{2m} v_2 \sqrt{P} \mathbf{c}_2[b + 1] + g_{rm} \mathbf{c}_r[b] + \mathbf{z}_m[b + 1]. \quad (4.25)$$

Destination j decodes $\mathbf{c}_r[b]$, treating the rest of the signal as noise. It is immediate that decoding is successful provided

$$R_r \leq \frac{1}{2} \log_2 \left(1 + \frac{P g_{rm}^2}{1 + P \|\mathbf{v} \circ \mathbf{h}_m\|^2} \right). \quad (4.26)$$

From $\mathbf{c}_r[b]$, the destination can recover the bin index $u[b]$. To recover $\hat{y}_r[b]$ from $u[b]$, destination j uses $\mathbf{y}_m[b]$ as side information. First, supposing that $\mathbf{c}_r[b - 1]$ has already

been decoded, destination m subtracts it from $\mathbf{y}_m[b]$ to form

$$\tilde{\mathbf{y}}_m[b] = \mathbf{y}_m[b] - \mathbf{c}_r[b-1] \quad (4.27)$$

$$= v_1 h_{1m} \sqrt{P} \mathbf{c}_1[b] + v_2 h_{2m} \sqrt{P} \mathbf{c}_2[b] + \mathbf{z}_m[b]. \quad (4.28)$$

Finally, destination j chooses the unique $\hat{\mathbf{y}}_r[b] \in \mathcal{C}_r^q$ that is both jointly typical with $\tilde{\mathbf{y}}_m[b]$ and a member of the bin $\psi(u[b])$. The Wyner-Ziv theorem guarantees that the receiver recovers the correct $\hat{\mathbf{y}}_r[b]$ so long as

$$R_r \geq I(\mathbf{y}_r[b]; \hat{\mathbf{y}}_r[b] | \tilde{\mathbf{y}}_h[b]). \quad (4.29)$$

Substituting (4.26) into (4.29) and carrying out straightforward manipulation yields the bound on quantization error σ_r^2 given in (4.18).

To decode $\mathbf{f}_m[b]$, destination m combines $\tilde{\mathbf{y}}_m[b]$ with $\hat{\mathbf{y}}_r[b]$ to form

$$\mathbf{y}'_r[b] = \tilde{\mathbf{y}}_m[b] + \gamma_m \hat{\mathbf{y}}_r[b] \quad (4.30)$$

$$= h'_{1m} \mathbf{c}_1[b] + h'_{2m} \mathbf{c}_2[b] + \mathbf{z}_m[b] + \mathbf{z}_r[b] + \mathbf{z}_q[b], \quad (4.31)$$

where h'_{ij} is defined as in (4.17). Now destination j can employ standard compute-and-forward decoding. Applying (4.31) to the standard compute-and-forward rate in (4.12) yields the rates claimed in (4.15), which establishes the claim. \square

The quantized relay signal $\hat{\mathbf{y}}_r[b]$ affords the destinations quite a bit of flexibility. For almost every channel realization, it is straightforward to choose γ_j such that the equivalent channels h'_{jm} are co-linear with suitable finite-field linear combinations, thus eliminating the Cauchy-Schwarz penalty. However, (4.18) shows that the distortion σ_r^2 grows linearly in P . This fact induces a tradeoff. By limiting the magnitude of γ_j , destination j can mitigate the compression noise, but only at the cost of reduced

freedom to align the equivalent channels h'_{lm} . To the best of my knowledge, while this tradeoff is sufficient for a clear improvement at finite SNR, it does not improve the degrees-of-freedom.

Figure 4.2 shows the achievable rate from Theorem 4.2 alongside the upper bound from Theorem 4.1 and the lower bound provided by standard compute-and-forward. Channel gains are set to unity except for $h_{21} = h_{12} = h$, which are taken to be the independent variable. The optimal choice of v_1, v_2 and γ_1, γ_2 is non-convex and even non-smooth. I therefore resort to numerical optimization, which introduces numerical artifacts. Note, nevertheless, the noticeable increase in computation rate. In particular, many of the “valleys” from the non-cooperative case become “peaks.”

4.3 Instantaneous relay

The *instantaneous* relay, or “relay-without-delay,” was first studied in [95] in the context of the three-terminal relay channel. Instead of relying on signals received only in previous symbol times, the relay can form its transmit signal from signals received up to the *current* symbol time. In [95] it was shown that the capacity of the instantaneous relay channel is often higher than that of the standard relay channel, albeit with no improvement to the degrees-of-freedom. Moreover, for a non-trivial set of channel gains, memoryless amplify-and-forward achieves capacity.

Instantaneous relaying has also been studied in the context of the relay-interference channel. In [96] the strong and very strong interference regimes are examined, and the capacity is derived for a range of channel coefficients. In [97] the degrees-of-freedom is studied. By contrast to the relay channel—as well as the relay-interference channel with a standard relay—they find that the introduction of an instantaneous relay improves the achievable DoFs. Using interference alignment techniques, they show that $3/2$ DoFs can be achieved. This result falls short of the interference-free maximum

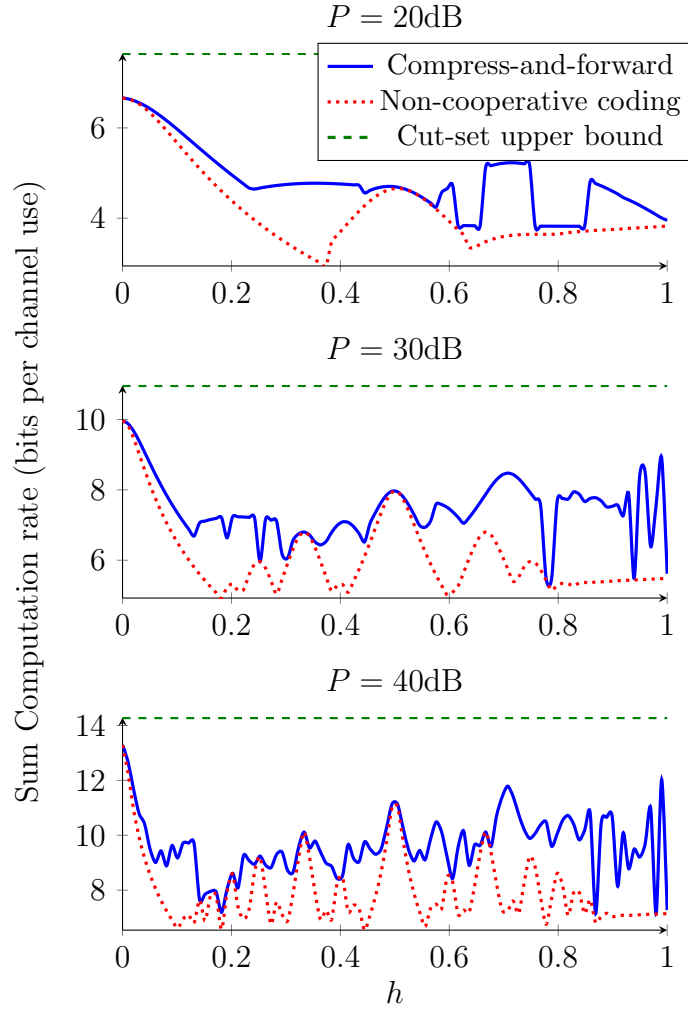


Figure 4.2: Achievable rates as a function of h and P for the relay computation channel with a standard relay.

of two, but it offers a 50% increase over the case of an standard relay (or no relay at all).

In the case of compute-and-forward, an instantaneous relay can obtain the full DoF while retaining provably good performance at finite SNR. I begin by proving an upper bound on the computation rate.

Theorem 4.3. *For the two-user relay computation channel with an standard relay,*

any achievable computation rate satisfies

$$R \leq \min_l \min_{m: a_{lm} \neq 0} \max_{|\alpha| \leq 1} \min \left\{ \frac{1}{2} \log_2(1 + P(g_{lr}^2 + h_{lm}^2 + 2g_{lr}h_{lm}\sqrt{\alpha})), \right. \\ \left. \frac{1}{2} \log_2 \left(1 + P \left(g_{rm}^2 + h_{lm}^2 - \frac{\alpha}{1 + g_{rm}h_{lm}(1 - \alpha)P} \right) \right) \right\}, \quad (4.32)$$

for some full-rank set of coefficients a_{lm} .

Proof. As in Theorem 4.1, suppose a genie supplies \mathbf{w}_2 to the destinations. Then, the computation capacity is limited by the capacity of the relay channels between source 1, the relay, and the relevant destinations. The capacity of those relay channels is bounded by the relay-without-delay cut-set bound derived in [95], which establishes the result for transmitter 1. Repeating the argument for transmitter 2 by supplying \mathbf{w}_1 to the destinations establishes the result. \square

Again the upper bound clearly implies a maximum DoF of two. For this relay modality, however, I construct an amplify-and-forward scheme that achieves this maximum. Again the relay receives a noisy superposition of source transmissions. Instead of compressing the signal and retransmitting in a subsequent block, the relay instantaneously forwards a scaled version of the received symbols. The destinations receive a noisy superposition of source transmissions that can be *tuned* by the choice of scaling coefficient at the relay. This scheme is detailed in the following theorem.

Theorem 4.4. *For the two-user relay computation channel with an instantaneous relay, the following computation rate is achievable:*

$$R_{\text{instantaneous}} = \min_l \min_{m: a_{lm} \neq 0} \left[\frac{1}{2} \log_2 \left(1 + P' \|\mathbf{h}'_m\|^2 \right) - \right. \\ \left. \frac{1}{2} \log_2 \left(\|\mathbf{a}_m\|^2 - P'(\|\mathbf{h}'_m\|^2 \|\mathbf{a}_m\|^2 - \langle \mathbf{h}'_m, \mathbf{a}_m \rangle^2) \right) \right]^+, \quad (4.33)$$

where

$$\mathbf{h}'_m = \mathbf{v} \circ \mathbf{h}_m + v_r g_{rm} \mathbf{g}_r, \quad (4.34)$$

and

$$P' = \frac{P}{1 + v_r^2}, \quad (4.35)$$

for any $-1 \leq v_1, v_2 \leq 1$ and for any

$$v_r^2 \leq \frac{P}{P \|\mathbf{v} \circ \mathbf{g}_r\|^2 + 1}. \quad (4.36)$$

Proof. As before, I present the encoding and decoding stages separately, then show that they achieve the computation rates claimed.

Encoding: Again each transmitter employs a nested lattice codebook \mathcal{C} , and, similar to before, define

$$\mathbf{c}_l = [\phi(\mathbf{w}_l) + \mathbf{d}_l] \bmod \Lambda_s, \quad (4.37)$$

where \mathbf{d}_l is uniform dither over \mathcal{V}_s and independent for each l . Transmitter l simply transmits its dithered lattice codeword \mathbf{c}_l , scaled by the coefficient v_l :

$$\mathbf{x}_l = v_l \sqrt{P} \mathbf{c}_l, \quad (4.38)$$

where $v_l^2 \leq 1$ ensures that \mathbf{x} satisfies the power constraint. By (4.3) the relay receives the signal

$$\mathbf{y}_r = g_{1r} v_1 \sqrt{P} \mathbf{c}_1 + g_{2r} v_2 \sqrt{P} \mathbf{c}_2 + \mathbf{z}_r. \quad (4.39)$$

The relay employs an amplify-and-forward strategy. It simply transmits a scaled version of its incoming signal, which is possible because it can relay instantaneously:

$$\mathbf{x}_r = v_r \mathbf{y}_r. \quad (4.40)$$

In order to satisfy the power constraint at the relay, the transmit signal must satisfy

$$\frac{1}{n} \|\mathbf{x}_r\|^2 \leq P, \quad (4.41)$$

which implies

$$v_r^2 P \|\mathbf{v} \circ \mathbf{g}_r\|^2 + 1 \leq P \quad (4.42)$$

$$\implies v_r^2 \leq \frac{P}{P \|\mathbf{v} \circ \mathbf{g}_r\|^2 + 1}, \quad (4.43)$$

which necessitates the requirement in (4.36).

Decoding: By (4.4) and (4.40), destination m receives the signal

$$\begin{aligned} \mathbf{y}_m &= (h_{1m} + v_r g_{rm} g_{1r}) v_1 \sqrt{P} \mathbf{c}_1 + \\ &\quad (h_{2m} + v_r g_{rm} g_{2r}) v_2 \sqrt{P} \mathbf{c}_2 + \mathbf{z}_m + v_r \mathbf{z}_r \\ &= h'_{1m} \sqrt{P} \mathbf{c}_1 + h'_{2m} \sqrt{P} \mathbf{c}_2 + \mathbf{z}_m + v_r \mathbf{z}_r, \end{aligned} \quad (4.44)$$

where h'_{lm} is defined as in (4.34). Now the received signal is equivalent to the non-cooperative case, where the presence of the relay noise yields an equivalent signal-to-noise ratio of $P' = P/(1 + v_r^2)$. Each destination performs ordinary compute-and-forward lattice decoding on \mathbf{y}_r , which using (4.12) establishes the desired result. \square

By appropriate choice of the parameters v_1, v_2 , and v_r , one can show that the amplify-and-forward scheme of Theorem 4.4 achieves the full DoFs.

Theorem 4.5. *For the two-user relay computation channel with an instantaneous relay, the achievability scheme of Theorem 4.4 achieves the maximum DoF of two.*

Proof. Several parameter choices achieve the full DoFs; here I point out only one option. The strategy is to choose v_r in order to perfectly cancel out the contribution

of \mathbf{c}_2 to \mathbf{y}_1 , and to choose v_1, v_2 such that the contributions of $\mathbf{c}_1(\mathbf{w}_1)$ and $\mathbf{c}_2(\mathbf{w}_2)$ to \mathbf{y}_2 have the same magnitude.

To satisfy the first desideratum, the weights must obey

$$\begin{aligned} h_{21} + v_r g_{r1} g_{2r} &= 0 \\ \implies v_r &= -\frac{h_{21}}{g_{r1} g_{2r}}. \end{aligned} \quad (4.45)$$

To satisfy the second desideratum, they must obey

$$\begin{aligned} (h_{12} + v_r g_{r2} g_{1r}) v_1 &= (h_{22} + v_r g_{r2} g_{2r}) v_2 \\ \implies \frac{v_1}{v_2} &= \frac{h_{22} + v_r g_{r2} g_{2r}}{h_{12} + v_r g_{r2} g_{1r}} \\ &= \frac{h_{22} - h_{21} g_{r2} / g_{r1}}{h_{12} - h_{21} g_{r2} g_{1r} / (g_{r1} g_{2r})}. \end{aligned} \quad (4.46)$$

If the numerator of the RHS of (4.46) has higher magnitude than the denominator, set $v_1 = 1$ and solve (4.46) for v_2 ; otherwise do the opposite. Note that (4.45) and (4.46) have solutions for almost every set of channel gains. Furthermore, for sufficiently large P the choice of v_r in (4.45) satisfies the power constraint in (4.36).

With these choices, the received signals at the destinations are

$$\mathbf{y}_1 = h'_1 \mathbf{c}_1(\mathbf{w}_1) + \mathbf{z}_1 + v_r \mathbf{z}_r \quad (4.47)$$

$$\mathbf{y}_2 = h'_2 (\mathbf{c}_1(\mathbf{w}_1) + \mathbf{c}_2(\mathbf{w}_2)) + \mathbf{z}_2 + v_r \mathbf{z}_r, \quad (4.48)$$

where h'_1, h'_2 are fixed equivalent channel gains. Choose $\mathbf{a}_1 = (1, 0)^T$ and $\mathbf{a}_2 = (1, 1)^T$,

which yields the achievable rates

$$R_1 = \frac{1}{2} \log_2 \left(1 + \frac{P}{1 + v_r^2} (h'_1)^2 \right) \approx \frac{1}{2} \log_2(P) \quad (4.49)$$

$$R_2 = \frac{1}{2} \log_2 \left(1 + \frac{4P}{1 + v_r^2} (h'_2)^2 \right) - 1 \approx \frac{1}{2} \log_2(P). \quad (4.50)$$

Substituting these into the DoFs definition of (4.13) establishes the result. \square

When channel gains are symmetric, one can do even better. Let $g_{ij} = 1$ for all i, j , $h_{ii} = 1$ for all i , and $h_{ij} = h$ for all $i \neq j$. Then, the computation capacity is bounded as follows.

Theorem 4.6. *For the symmetric two-user relay computation channel with an instantaneous relay, the computation rates are bounded as follows:*

$$\frac{1}{2} \log_2(P) - 1 - c \leq R \leq \frac{1}{2} \log_2^+(P) + \frac{3}{2}, \quad (4.51)$$

for $P \geq 1$ and $0 \leq h \leq 1$, except possibly for an outage interval $\mathcal{I} = [1 - 2^{-c}, 1]$.

Proof. First I prove the upper bound. Using the second term from the bound in Theorem 4.3 and choosing $\alpha = 0$ yields

$$R \leq \frac{1}{2} \log_2(1 + P(1 + \max\{1, h^2\})) \quad (4.52)$$

$$\leq \frac{1}{2} \log_2^+(2P) + 1 \quad (4.53)$$

$$\leq \frac{1}{2} \log_2^+(P) + \frac{3}{2}. \quad (4.54)$$

The lower bound results from an explicit choice of function coefficients and weighting coefficients. In particular, choose $v_1 = v_2 = 1$ and choose the function coefficients to be symmetric, i.e. $a_{11} = a_{22} = a_1$ $a_{12} = a_{21} = a_2$. Then, choose v_r such that the

equivalent channel are co-linear with the function coefficients, which entails

$$1 + v_r = xa_1 \quad (4.55)$$

$$h + v_r = xa_2, \quad (4.56)$$

for some $x \in \mathbb{R}$. Solving (4.55) and (4.56) for x and v_r yields

$$x = \frac{1 - h}{a_1 - a_2} \quad (4.57)$$

$$v_r = a_1 \frac{1 - h}{a_1 - a_2} - 1. \quad (4.58)$$

Note that x and v_r are uniquely determined by the choice of a_1, a_2 . Substituting these choices into the achievable rate from Theorem 4.4 yields

$$R \geq \frac{1}{2} \log_2 \left(1 + \frac{P}{1 + v_r^2} x^2 \right). \quad (4.59)$$

Therefore, receivers choose a_1 and a_2 to minimize x^2 , which entails the minimization of $(a_1 - a_2)^2$ subject to a feasible v_r . Recall from (4.36) that $v_r^2 \leq P/(2P + 1) \triangleq V$. Therefore

$$-\sqrt{V} \leq a_1 \frac{1 - h}{a_1 - a_2} - 1 \leq \sqrt{V} \quad (4.60)$$

$$\frac{-\sqrt{V} + 1}{1 - h} \leq \frac{a_1}{a_1 - a_2} \leq \frac{\sqrt{V} + 1}{1 - h}. \quad (4.61)$$

Define $b \triangleq a_1 - a_2$. Then

$$b \frac{-\sqrt{V} + 1}{1 - h} \leq a_1 \leq b \frac{\sqrt{V} + 1}{1 - h}. \quad (4.62)$$

Now, in order for a suitable $a_1 \in \mathbb{Z}$ to exist, the gap in (4.62) must be greater than

unity, which entails

$$b \geq \frac{1-h}{2\sqrt{V}}. \quad (4.63)$$

Since $b \in \mathbb{Z}$, choose

$$b = \left\lceil \frac{1-h}{2\sqrt{V}} \right\rceil, \quad (4.64)$$

and choose a_1 to be any integer satisfying (4.62). Now, going back to the definition of x , note that

$$x = \frac{1-h}{a_1 - a_2} = \frac{1-h}{b} \quad (4.65)$$

$$\geq \frac{1-h}{\frac{1-h}{2\sqrt{V}} + 1} \quad (4.66)$$

$$\geq \frac{1-h}{\frac{1}{2\sqrt{V}} + 1} \quad (4.67)$$

$$\geq \frac{(1-h)2\sqrt{V}}{3}. \quad (4.68)$$

Substituting this result back into (4.59) yields

$$R \geq \frac{1}{2} \log_2 \left(P(1-h)^2 \frac{9}{4} V \right) - \frac{1}{2} \quad (4.69)$$

$$= \frac{1}{2} \log_2 \left(P(1-h)^2 \frac{9P}{8P+4} \right) - \frac{1}{2} \quad (4.70)$$

$$\geq \frac{1}{2} \log_2 \left(P(1-h)^2 \frac{3}{4} \right) - \frac{1}{2} \quad (4.71)$$

$$\geq \frac{1}{2} \log_2(P) - 1 + \log_2(1-h). \quad (4.72)$$

Finally, note that $\log_2(1-h) \leq -c$ only when $h \geq 1 - 2^{-c}$. Therefore, the outage interval on which the claimed bound does not apply is no bigger than $\mathcal{I} = [1 - 2^{-c}, 1]$. \square

Figure 4.3 shows the sum rates achieved by the strategy presented in Theorem

4.6, alongside the upper and lower bounds proven therein and the rates achieved by non-cooperative compute-and-forward. In these plots the constant $c = 1$ is chosen, so the lower bound applies only when $h < 1/2$. As h approaches 1, the channel matrix between transmitters and receivers approaches one, and it becomes impossible to decode symmetric functions at non-zero rate. The amplify-and-forward rate, however, varies relatively smoothly in h ; the need for (approximately) rational channel gains is eliminated.

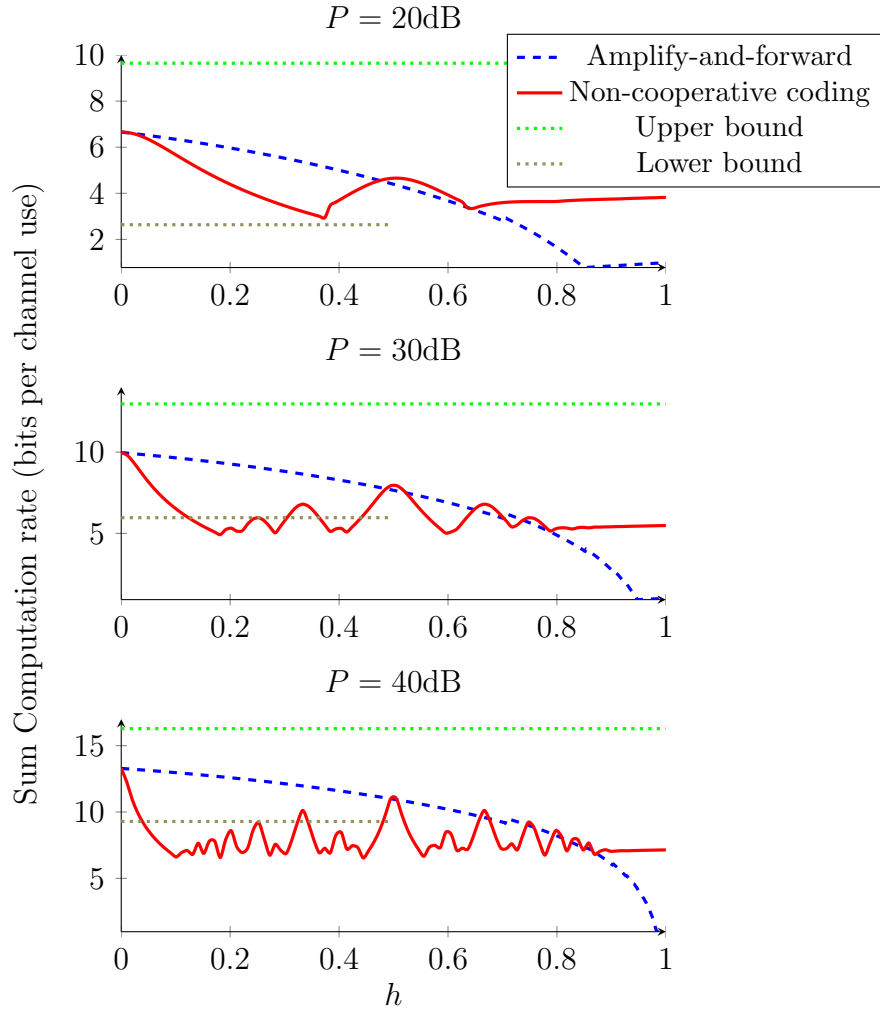


Figure 4.3: Achievable rates as a function of h and P for the relay computation channel with an instantaneous relay.

Consensus

In this chapter I study averaging consensus over wireless. By contrast to the previous two chapters, here I study large, dense networks and derive inner and outer bounds on how the resource requirements of consensus scale with the network size. I propose a simple but sufficiently realistic wireless model based on path-loss propagation and SNR-based connectivity. I define three key resource metrics: expended energy, elapsed time, and consumed time-bandwidth product. Under this model, I first examine a few existing consensus strategies, showing that they may be nearly optimal with respect to expended energy, but they remain strictly suboptimal with respect to the other two metrics. I then propose an explicitly wireless approach to consensus, termed *hierarchical averaging*. Depending on the details of the propagation environment, hierarchical averaging is order-optimal with respect to all three metrics simultaneously. Finally, I examine the effects of quantization on consensus performance, showing that hierarchical averaging obtains a near-optimal tradeoff between resource consumption and estimation error.

5.1 Preliminaries

Consider a network of N nodes, each of which possesses a scalar measurement $z_n(0) \in \mathbb{R}$. In averaging consensus, each node wishes to compute the average of these measurements

$$z_{\text{ave}} = \frac{1}{N} \sum_{n=1}^N z_n(0), \quad (5.1)$$

by means of local interactions. As discussed earlier, most previous approaches to consensus suppose a wired, graphical topology. In this chapter I examine the implications of the wireless medium on consensus, which necessitates the definition of a tractable, but realistic, wireless model.

5.1.1 System Model

The proposed wireless model strikes a balance between tractability and practicality. It entails four critical assumptions which capture the salient features of wireless while maintaining simplicity: synchronous transmission, path-loss propagation, “protocol”-model connectivity, and orthogonalized interference management. In this subsection I detail and justify these assumptions.

Although consensus algorithms are occasionally defined over synchronous models (e.g., the synchronized gossip from [55]), researchers more commonly assume communications to be asynchronous. Each node has an independent clock that “ticks” at Poisson-distributed intervals; upon each clock tick the node initiates a round of consensus, which is assumed to take place instantaneously. This model is an idealized version of ALOHA-style protocols, and it sidesteps the scheduling and interference difficulties inherent to wireless communications. Here, however, the goal is both to characterize the best possible performance under wireless and to address interference. I therefore adopt a synchronous model in which nodes transmit simultaneously in

slotted time. In practice, near-perfect synchronization can be achieved via beacons, as in the superframes of 802.15, or via GPS clocks. Formally, let $x_n(t)$ denote the signal transmitted by node n , and let $P_n(t) = |x_n(t)|^2$ denote the transmit power, during time slot t .

I suppose a path-loss propagation model. Each node n has a geographic location $\mathbf{r}_n \in [0, 1] \times [0, 1]$ taken to be independently drawn from a uniform distribution. Under the path-loss model, the channel gain between any two nodes m, n is

$$h_{mn} = e^{j\theta_{mn}} \|\mathbf{r}_m - \mathbf{r}_n\|_2^{-\alpha/2}, \quad (5.2)$$

where $\alpha \geq 2$ is the path-loss exponent, and $\theta_{mn} \in [0, 2\pi)$ is a random phase with distribution to be specified later.

I further suppose a “protocol” connectivity model. The signal $x_m(t)$ is said to *arrive* at node n provided the received power is above an arbitrary threshold γ . Define the *neighborhood* of node n as the set of nodes whose transmissions have sufficient received power:

$$\begin{aligned} \mathcal{N}_n(t) &= \{m : P_m(t) |h_{mn}|^2 \geq \gamma\} \\ &= \{m : P_m(t) \geq \gamma \|\mathbf{r}_m - \mathbf{r}_n\|_2^\alpha\}. \end{aligned} \quad (5.3)$$

For nodes $m \notin \mathcal{N}_n(t)$, I assume that node n suffers no interference from node m ’s transmission. This assumption permits a tractable, geometric analysis of connectivity.

In hierarchical averaging, presented in Section 5.4, nodes are grouped into clusters which transmit cooperatively. In this case the definition of neighborhoods must be expanded to characterize the number of unique *signals* arriving at node n . Let $\mathcal{C} \subset \{1, \dots, N\}$ denote a cluster of nodes transmitting the signal $x_{\mathcal{C}}(t)$. Define the received

power at node n as

$$R_{\mathcal{C},n}(t) = E \left[\left| \sum_{m \in \mathcal{C}} h_{mn} \sqrt{P_m(t)} \right|^2 \right],$$

where the expectation is taken over the random phases. Then, the neighborhood of n is the set of all clusters \mathcal{C} such that the received power exceeds γ :

$$\mathcal{N}_n(t) = \left\{ \mathcal{C} : E \left[\left| \sum_{m \in \mathcal{C}} h_{mn} \sqrt{P_m(t)} \right|^2 \right] \geq \gamma \right\}.$$

The connectivity of clusters depends on the distribution of the phases θ_{mn} . In the sequel I consider two choices. First, I consider the simple case in which the phases are equal and fixed. In this case, signals constructively combine at receivers, and the neighborhood of n can be written as

$$\mathcal{N}_n(t) = \left\{ \mathcal{C} : \left(\sum_{m \in \mathcal{C}} h_{mn} P_m^{\frac{1}{2}}(t) \right)^2 \geq \gamma \right\}. \quad (5.4)$$

The second, and more realistic, case I consider is that each θ_{mn} is independently and uniformly distributed across $[0, 2\pi)$. In this case signals do not combine coherently, and the neighborhood of n is

$$\mathcal{N}_n(t) = \left\{ \mathcal{C} : \sum_{m \in \mathcal{C}} h_{mn}^2 P_m(t) \geq \gamma \right\}. \quad (5.5)$$

The final assumption is a simple orthogonalized approach to interference management. For every $m \in \mathcal{N}_n(t)$, node n receives the following signal:

$$y_{mn}(t) = h_{mn}x_m(t) + w_{mn}(t), \quad (5.6)$$

where $w_{mn}(t)$ is unit-variance Gaussian noise. In other words, incoming signals arrive

independently and do not interfere. In order to avoid such interference, incoming transmissions must arrive on orthogonal sub-channels. The nature of the sub-channels is left unspecified; they may be realized in time, frequency, or code. In order for nodes to orthogonalize, there must be

$$B(t) = \max_n |\mathcal{N}_n(t)| \quad (5.7)$$

sub-channels available during time slot t . I do not worry about the specific allocation of nodes to sub-channels. In addition to distributed techniques such as asynchronous CDMA, there exist distributed graph coloring algorithms [98] that achieve an order-optimal sub-channel allocation.

5.1.2 Performance Metrics: Infinite-Rate Links

I first consider the case in which the links between neighboring nodes are perfect; that is, at time t node n decodes a real-valued scalar from each $m \in \mathcal{N}_n(t)$. This is obviously a simplification, since wireless links are necessarily rate-limited. However, most gossip algorithms are founded on the ability to exchange values with infinite precision, and I will make this assumption in the first part of this work. Later I will assume finite-rate links, which will necessitate a different set of metrics.

The first figure of merit under consideration is the ϵ -averaging time. During each time slot t , nodes exchange estimates with neighbors and update their estimates accordingly. The ϵ -averaging time, denoted T_ϵ , is the number of time slots required to achieve consensus to within a specified tolerance:

$$T_\epsilon = \sup_{\mathbf{z}(0) \in \mathbb{R}^n} \inf \left\{ t : \Pr \left(\frac{\|\mathbf{z}(t) - z_{\text{ave}} \mathbf{1}\|}{\|\mathbf{z}(0)\|} \geq \epsilon \right) \leq \epsilon \right\}, \quad (5.8)$$

where $\mathbf{z}(t)$ is the vector of estimates $z_n(t)$. The scaling law of T_ϵ is the primary focus

of study for most gossip algorithms. However, it provides only a partial measure of resource consumption in wireless networks, so I define further metrics.

I next examine energy, which is scarce in networks composed of cheap, battery-powered nodes. Define the *total transmit energy* as the energy required to achieve consensus to within the tolerance ϵ :

$$E_\epsilon = \sum_{n=1}^N \sum_{t=1}^{T_\epsilon} P_n(t). \quad (5.9)$$

Supposing each time slot to be of equal length, the transmit power $P_n(t)$ is proportional to the energy consumed by node n over slot t . Summing over nodes and time slots yields the total energy consumed.

The final figure of merit is the *time-bandwidth product*, defined as

$$B_\epsilon = \sum_{t=1}^{T_\epsilon} B(t) = \sum_{t=1}^{T_\epsilon} \max_n |\mathcal{N}_n(t)|. \quad (5.10)$$

The metric B_ϵ measure the total number of sub-channel uses required to achieve consensus to tolerance ϵ , which, as mentioned previously, may be realized in time, frequency, or code. However T_ϵ represents the *temporal component* of the time-bandwidth product. The sequential nature of consensus dictates that T_ϵ rounds occur in succession. Therefore T_ϵ characterizes a constraint on the realization of the time-bandwidth product.:All of the time-bandwidth product may be realized with temporal resources, but only a fraction of it may be realized by frequency resources.

5.1.3 Performance Metrics: Finite-Rate Links

In practice, wireless links are noisy and therefore have finite rate, which precludes the infinite-precision exchange of scalars. Instead, nodes must quantize their estimates to a finite alphabet prior to each round of consensus. To simplify the discussion, suppose

that the measurements $z_n(0)$ are drawn from the finite interval $[0, 1)$. Throughout this paper, nodes employ dithered uniform quantization described in [66]. The quantization alphabet \mathcal{Z} is defined as

$$\mathcal{Z} = \left\{ \frac{1}{L+1}, \frac{2}{L+1}, \dots, \frac{L}{L+1} \right\}, \quad (5.11)$$

for some alphabet size L . The quantizer is defined as

$$\phi(z) = \min_{q \in \mathcal{Z}} |q - (z + u)|, \quad (5.12)$$

where u is a dither, drawn uniformly and randomly from $[-\Delta/2, \Delta/2)$ each time ϕ is called. Statistically, one can write the quantized value as

$$\phi(z) = z + v,$$

where v is uniformly distributed across $[-\Delta/2, \Delta/2)$ and independent of z .

The alphabet size $L = |\mathcal{Z}|$ depends on the quality of the wireless links. Since connectivity requires signal-to-noise threshold γ , L is determined by the Shannon capacity of a wireless link at SNR γ . Supposing unit bandwidth and block duration, nodes successfully exchange $\log_2(1 + \gamma)$ bits over the wireless links [13], which results in an alphabet size of $L = \lfloor 2^{\log_2(1+\gamma)} \rfloor = \lfloor 1 + \gamma \rfloor$.

With quantization it becomes difficult to speak of convergence time. For a large class of consensus algorithms, the dynamics does not converge on the true average to within any finite tolerance, precluding defining T_ϵ as before. In fact, quantization induces a tradeoff between resource consumption and estimate quality.

For a consensus algorithm with quantization, let T be the number of rounds for which consensus runs. Then let B and E be the time-bandwidth product and total

transmit energy, defined as before but with T taking the role of T_ϵ . Finally, define the *mean squared error* as

$$\sigma^2 = \max_{\mathbf{z}(0) \in [0,1]^N} E \left[\frac{1}{N} \sum_{n=1}^N (z_n(T) - z_{\text{ave}})^2 \right], \quad (5.13)$$

where the expectation is taken over the randomness in the quantization operator as well as in the consensus algorithm. There is an inherent tradeoff between the total transmit energy E and the mean-squared error σ^2 ; one can always reduce the estimation error by injecting more transmit energy into the network and increasing the rate of the wireless links.

Finally, throughout this chapter I will rely on the following lemma, which shows that the number of nodes in a region is asymptotically proportional to its area to within an arbitrary tolerance δ .

Lemma 5.1 (Ozgur-Leveque-Tse, [87]). *Let $A \subset [0, 1] \times [0, 1]$ be a region inside the unit square having area $|A|$, and let $\mathcal{C} = \{n : \mathbf{r}_n \in A\}$ be the nodes lying in A . Then, for any $\delta > 0$,*

$$(1 - \delta)|A|N \leq |\mathcal{C}| \leq (1 + \delta)|A|N, \quad (5.14)$$

with probability greater than $1 - 1/|A|e^{-\Gamma(\delta)|A|N}$, where $\Gamma(\delta) > 0$ and is independent of N and $|A|$.

5.2 Inner Bounds

In this section I derive inner bounds on the resource costs for consensus over the proposed wireless model. I begin with the case of infinite-rate links.

Theorem 5.1. *For any consensus algorithm, with probability approaching 1 as $N \rightarrow$*

∞ :

$$T_\epsilon = B_\epsilon = \Omega(1) \quad (5.15)$$

$$E_\epsilon = \Omega(N^{1-\alpha/2}). \quad (5.16)$$

Proof. The bounds on T_ϵ and B_ϵ are trivial. To prove the bound on E_ϵ , observe that every node n must transmit its measurement $z_n(0)$ to at least one of its neighbors. The energy required for each node to transmit to its nearest neighbor can be expressed as

$$\begin{aligned} E_\epsilon &\geq \sum_{n=1}^N \min_{m \neq n} \gamma h_{mn}^{-2} \\ &= \gamma \sum_{n=1}^N d_{\min}^\alpha(n), \end{aligned} \quad (5.17)$$

where $d_{\min}(n)$ is the distance between node n and its nearest neighbor. It is well-known (e.g., in [99]), that $d_{\min}(n) = \Theta(N^{-1/2})$ with high probability, so

$$\begin{aligned} E_\epsilon &\geq \gamma \sum_{n=1}^N \Theta(N^{-\alpha/2}) \\ &= \Omega(N^{1-\alpha/2}). \end{aligned} \quad (5.18)$$

□

In the case of consensus with rate-limited links, I derive an inner bound on the tradeoff between resources and estimation error.

Theorem 5.2. *For any consensus algorithm with rate-limited links, any achievable*

tradeoff in performance metrics satisfies the following with high probability:

$$T = B = \Omega(1) \quad (5.19)$$

$$E = \sum_{n=1}^N \Omega(N^{x_n - \alpha/2}) \quad (5.20)$$

$$\sigma^2 = \frac{1}{N} \sum_{n=1}^N \Omega(N^{-2x_n}), \quad (5.21)$$

for $x_n > 0$. In particular, choosing each $x_n = x$ yields

$$E = \Omega(N^{1+x-\alpha/2}) \quad (5.22)$$

$$\sigma^2 = \Omega(N^{-2x}). \quad (5.23)$$

Proof. As in the ideal-link case, the bounds on T and B are trivial. To bound the tradeoff between energy and estimation error, momentarily consider a single node n . Suppose a genie supplies node n with z_{ave} , and further suppose that only node n 's nearest neighbor, denoted by m , needs to compute the average. In this case, the optimal strategy is for n to quantize z_{ave} and transmit it directly to node m . In principle, other nodes could transmit their measurements to m , but since they are no closer order-wise, and since they have only partial knowledge of the average, any energy they expend would be better used by node n .

Without loss of generality, let $P_n = N^{x_n - \alpha/2}$ denote the transmit power used by node n to transmit z_{ave} . Since again the distance between nearest neighbors is $\Theta(N^{-1/2})$ with high probability, the size of the quantization alphabet is $L = \Theta(N^{x_n})$. Therefore, the square quantization error at node n on z_{ave} is $|e_n|^2 = \Theta(L^{-2}) = \Theta(N^{-2x_n})$. Repeating the argument for each n gives the result. \square

5.3 Gossip Algorithms

In this section I characterize existing gossip algorithms with respect to the metrics defined in Section 5.1. There are, of course, too many instantiations of gossip for to analyze, so I focus on two variants¹ that provide a relatively comprehensive look at the state of the art: randomized gossip [55], which is probably the best-known approach to gossip, and path averaging [27], which is order optimal in terms of convergence speed. The first task is to adapt the graphical nature of gossip to the wireless model. The key criterion is the required transmit power. In order to achieve consensus, it is necessary to choose the topology of the network such that the resulting graph is connected. In [100] it is shown that, with high probability, a necessary and sufficient condition for connectedness is that each node be connected to every node within a radius of $\Theta(\sqrt{\log N/N})$. In terms of the neighborhood $\mathcal{N}_n(t)$, this implies that, for every node n that transmits during time slot t ,

$$\mathcal{N}_n(t) = \{m : \|\mathbf{x}_m - \mathbf{x}_n\|_2 < \Theta(\sqrt{\log N/N})\}.$$

By (5.3), the transmit power must satisfy

$$P_n(t) = \gamma r^\alpha = \Theta(\gamma(\log N/N)^{\alpha/2}), \quad (5.24)$$

for every node n transmitting during time slot t . This holds for both gossip algorithms considered in this section.

¹Due to its similarity with hierarchical averaging, one might suspect that multiscale gossip [28] has superior performance to the gossip algorithms studied here. However, one can show that the performance of multi-scale gossip is rather similar to that of path-averaging.

5.3.1 Randomized Gossip

Here I study the *synchronized* randomized gossip of [55]. At each time slot t , each node is randomly paired up with one of its neighbors. Paired nodes exchange estimates and average the estimates together, which results in the following dynamics:

$$\mathbf{z}(t) = \frac{1}{2}(\mathbf{W}(t) + \mathbf{I})\mathbf{z}(t-1),$$

where $\mathbf{W}(t)$ is a randomly-chosen permutation matrix such that $w_{mn} = 1$ only if nodes m and n are neighbors.

In [55, Theorem 9] the convergence of randomized gossip is characterized. It is shown that the averaging time satisfies

$$T_\epsilon = \Theta\left(N \frac{\log \epsilon^{-1}}{\log N}\right). \quad (5.25)$$

Using these facts, one can derive statements about the performance of randomized gossip with respect to the proposed metrics.

Theorem 5.3. *For randomized gossip, the resource consumption scales as follows with high probability:*

$$T_\epsilon = \Theta\left(N \frac{\log \epsilon^{-1}}{\log N}\right), \quad (5.26)$$

$$B_\epsilon = \Theta\left(N \log \epsilon^{-1}\right), \quad (5.27)$$

$$E_\epsilon = \Theta(N^{2-\alpha/2}(\log N)^{\alpha/2-1} \log \epsilon^{-1}). \quad (5.28)$$

Proof. The bound on T_ϵ follows from (5.25). Since every node transmits during every

time slot t , and since γ is a constant, $P_n(t) = \Theta((\log N/N)^{\alpha/2})$ for each t . Therefore

$$\begin{aligned} E_\epsilon &= \sum_{t=1}^{T_\epsilon} \sum_{n=1}^N \Theta\left(\frac{\log N}{N}\right)^{\alpha/2} \\ &= T_\epsilon \Theta(N^{1-\alpha/2} (\log N)^{\alpha/2}) \\ &= \Theta(N^{2-\alpha/2} (\log N)^{\alpha/2-1} \log \epsilon^{-1}). \end{aligned}$$

Next, the required connectivity radius means that each neighborhood is defined by a region of area $\Theta(\log N/N)$. By Lemma 5.1, each neighborhood size satisfies

$$\begin{aligned} (1 - \delta)\pi \log N &\leq |\mathcal{N}_n(t)| \leq (1 + \delta)\pi \log N \\ |\mathcal{N}_n(t)| &= \Theta(\log N) \end{aligned}$$

with high probability. Plugging this into (5.10) yields

$$\begin{aligned} B_\epsilon &= T_\epsilon \Theta(\log N) \\ &= \Theta(N \log \epsilon^{-1}). \end{aligned}$$

□

5.3.2 Path Averaging

Next I look at *path averaging*, a more sophisticated gossip algorithm proposed in [27]. Instead of exchanging estimates with a neighbor, in path averaging each node chooses a geographically distant node with which to exchange; the exchange is facilitated by multi-hop rounding. In addition to facilitating the exchange, the routing nodes add their estimates “along the way,” allowing many nodes to average together in a single round. Once the average of all the nodes’ estimates is computed at the destination,

the result is routed back to the source.

Path averaging is described in an asynchronous framework in which nodes independently “wake up,” initiate multi-hop exchanges, and return to an idle state sufficiently quickly that no two exchanges overlap in time. Placing path averaging into the proposed synchronous framework, suppose that at time t a pair of nodes n, m is randomly selected to engage in a multi-hop exchange. Letting $\mathcal{P}(t)$ be the set of nodes routing from n to m , suppose that the $2(|\mathcal{P}(t)| - 1)$ transmissions required to route from n to m and back happen sequentially and thus require $2(|\mathcal{P}(t)| - 1)$ time slots. At time slot $t + 2(|\mathcal{P}(t)| - 1)$, a new pair is chosen. The dynamics for path averaging has the following form:

$$z_n(t + 2(|\mathcal{P}(t)| - 1)) = \begin{cases} \frac{1}{|\mathcal{P}|} \sum_{m \in \mathcal{P}(t)} z_m(t), & n \in \mathcal{P}(t) \\ z_n(t), & \text{otherwise} \end{cases}. \quad (5.29)$$

In [27, Theorem 2] it is shown that, for a random uniform network¹, the expected path length is $E[|\mathcal{P}(t)|] = \Theta(\sqrt{N/\log N})$ and the number of exchanges required to achieve ϵ -consensus is $\Theta(\sqrt{N \log N} \log \epsilon^{-1})$. Combining these facts, the total number of required transmissions is $\Theta(N \log \epsilon^{-1})$.

In casting path averaging in the synchronous framework, I have retained the assumption that multi-hop exchanges do not overlap in time. In principle one could construct a synchronous path-averaging gossip in which multiple exchanges occur simultaneously, perhaps reducing the total amount of time required to achieve consensus. In the following theorem, I provide a rather optimistic bound on the resource consumption of any such synchronous formulation.

Theorem 5.4. *For any synchronous path-averaging gossip, the resource consumption*

¹Technically, the convergence speed of path averaging is proven over a torus, so the results proven in the sequel apply to the torus. Later I provide numerical results that establish empirically that the same results apply to a square network.

scales as follows with high probability:

$$T_\epsilon = B_\epsilon = \Omega\left(\sqrt{\frac{N}{\log N}}\right) \quad (5.30)$$

$$E_\epsilon = \Theta(N^{1-\alpha/2} \log \epsilon^{-1}). \quad (5.31)$$

Proof. The bound on T_ϵ and B_ϵ follows from the fact that each route has $\Theta(\sqrt{\frac{N}{\log N}})$ hops. Even in the ideal case in which every round of gossip occurs simultaneously, $T_\epsilon = \Omega(\sqrt{N/\log N})$ sequential transmissions are still required. Supposing optimistically that constant bandwidth is sufficient to accommodate the multiple exchanges, the same bound applies to B_ϵ .

To bound E_ϵ , note that, as with randomized gossip, $P_n(t) = \Theta((\log N/N)^{\alpha/2})$ is required for every transmission. Since path-averaging requires $\Theta(N \log \epsilon^{-1})$ transmissions, the overall energy consumption scales as

$$E_\epsilon = \Theta(N^{1-\alpha/2} (\log N)^{\alpha/2} \log \epsilon^{-1}). \quad (5.32)$$

□

5.4 Hierarchical averaging

In this section I introduce *hierarchical averaging*. Much like multi-scale gossip [28] and the hierarchical cooperation of [87], in hierarchical averaging the network is recursively partitioned into geographically defined clusters. Each cluster achieves internal consensus by mutually broadcasting estimates. Nodes within a cluster then cooperatively broadcast their identical estimates to neighboring clusters at the next level. The process continues until the entire network achieves consensus. In the following subsection I describe the recursive partition, after which I describe the algorithm in

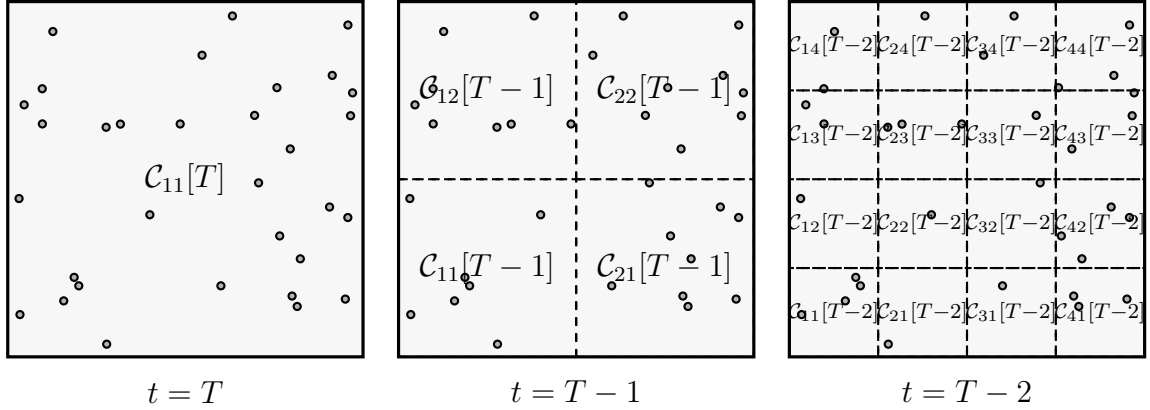


Figure 5.1: Hierarchical partition of the network. Each square cell is divided into four smaller cells, which are each divided into four smaller cells, and so on.

detail and characterize its resource requirements.

5.4.1 Hierarchical Partitioning

The network is partitioned into T sub-network layers, one for each round of consensus, as depicted in Figure 5.1. At the top layer, which corresponds to the final round $t = T$ of consensus, there is a single cell. At the next-highest level $t = T - 1$, the network is divided into four equal-area square cells. Continuing, each cell is recursively divided into four smaller cells until the lowest layer $t = 1$, which corresponds to the first round of consensus. At each level t there are 4^{T-t} cells, formally defined as

$$\mathcal{C}_{jk}(t) = \{n : \mathbf{r} \in [(j-1)2^{t-T}, j2^{t-T}) \times [(k-1)2^{t-T}, k2^{t-T})\}, \quad (5.33)$$

where $1 \leq j, k \leq 2^{T-t}$ index the geographical location of the cell.

Let $\mathcal{C}(n, t)$ denote the unique cell at layer t containing node n . Using the Pythagorean theorem, one can easily bound the maximum distance between any two nodes:

$$M(t) = \sqrt{2} \cdot 4^{\frac{t-T}{2}} = \Theta(4^{\frac{t-T}{2}}), \quad (5.34)$$

where the maximum is achieved when two nodes lie on opposite corners of the cell.

The number of layers is chosen as $T = \lceil \log_4(N^{1-\kappa}) \rceil$, where $\kappa > 0$ is a small constant. In the following I characterize the order-wise cardinality of each cell, which I will use in deriving the resource consumption of hierarchical averaging.

Lemma 5.2. *For every $1 \leq j, k \leq 2^{T-t}$ and $1 \leq t \leq T$,*

$$|\mathcal{C}_{jk}(t)| = \Theta(4^t N^\kappa), \quad (5.35)$$

with probability greater than $1 - N^{2-2\kappa}/16 \cdot e^{-\Gamma(\delta)N^\kappa}$.

Proof. The area of each cell at layer $t = 1$ is, by construction

$$\begin{aligned} A &= 4^{1-T} \\ 4^{1-\log_4(N^{1-\kappa})+1} &\leq A \leq 4^{1-\log_4(N^{1-\kappa})} \\ N^{\kappa-1} &\leq A \leq 4N^{\kappa-1}. \end{aligned}$$

Then, by Lemma 5.1, the cardinality of each cell at layer $t = 1$ is bounded by

$$(1 - \delta)N^\kappa \leq |\mathcal{C}_{jk}(t)| \leq (1 + \delta)4N^\kappa, \quad (5.36)$$

with probability greater than $1 - N^{1-\kappa}/16 \cdot e^{-\Gamma(\delta)N^\kappa}$.

Define $E_{jk}(1)$ as the event in which $|\mathcal{C}_{jk}(1)|$ is outside the bounds specified in (5.36). Clearly $\Pr\{E_{jk}(1)\} \leq N^{1-\kappa}/16 \cdot e^{-\Gamma(\delta)N^\kappa}$. Therefore, by the union bound,

$$\Pr\left(\bigcup_{1 \leq j, k \leq 2^{T-1}} E_{j,k}(1)\right) \leq \sum_{1 \leq j, k \leq 2^{T-1}} N^{1-\kappa}/16 \cdot e^{-\Gamma(\delta)N^\kappa} \quad (5.37)$$

$$\leq N^{2-2\kappa}/16 \cdot e^{-\Gamma(\delta)N^\kappa} \rightarrow 0. \quad (5.38)$$

Therefore, every cell at $t = 1$ simultaneously satisfies $|\mathcal{C}_{jk}(1)| = \Theta(N^\kappa)$ with the desired probability. Now, since each cell at layer t is composed of 4^{t-1} cells at layer 1, with the same probability

$$|\mathcal{C}_{jk}(t)| = \Theta(4^t N^\kappa). \quad (5.39)$$

□

5.4.2 Algorithm Description

Next I lay out the details of hierarchical averaging. Each node n requires the following information about the network: the total number of nodes N , its own location \mathbf{r}_n , and the number of consensus rounds T .

At time slot $t = 1$, each node broadcasts its initial estimate $z_n(0)$ to every member of its cluster $\mathcal{C}(n, 1)$. In order to ensure that $n \in \mathcal{N}_m(t)$ for every $m \in \mathcal{C}(n, 1)$, each node transmits at power

$$P_n(1) = \gamma \max_{m \in \mathcal{C}(n, 1)} h_{nm}^\alpha \leq \gamma M(1)^\alpha = O(N^{(\kappa-1)\alpha/2}). \quad (5.40)$$

Each node n takes a weighted average of the estimates in its cluster:

$$z_n(1) = \frac{1}{4^{1-T} N} \sum_{m \in \mathcal{C}(n, 1)} z_m(0). \quad (5.41)$$

The nodes use the approximate normalization factor $1/4^{1-T} N$ instead of the exact factor $1/|\mathcal{C}(n, 1)|$ so that nodes at higher levels of the hierarchy need not know the cardinality of the cells. This approximation introduces no error into the final estimate.

After time slot $t = 1$, each node in each cluster $\mathcal{C}_{jk}(1)$ has the same estimate, denoted by $z_{\mathcal{C}_{jk}(1)}(1)$. At each subsequent time slot $2 \leq t \leq T$, each cluster $\mathcal{C}(n, t-1)$

cooperatively transmits its estimate to its parent cluster at layer t . Each $P_n(t)$ is taken to be a constant. The transmit power required depends on the phase of the channel gains, as discussed in Section 5.1.1. When the phases are fixed and identical, by (5.4), the transmit powers must satisfy

$$\begin{aligned}
 & \left(\sum_{m \in \mathcal{C}(n, t-1)} h_{mn} P_m^{1/2}(t) \right)^2 = \gamma \\
 \Rightarrow P_m(t) &= \frac{\gamma}{\left(\sum_{m \in \mathcal{C}(n, t-1)} h_{mn} \right)^2} \\
 &\leq \frac{\gamma M(t)^\alpha}{|\mathcal{C}(n, t-1)|^2} \\
 &= O\left(\frac{(4^t N^{\kappa-1})^{\alpha/2}}{4^{2t} N^{2\kappa}} \right) \\
 &= O\left(4^{(\alpha/2-2)t} N^{-\alpha/2+\kappa(\alpha/2-2)} \right). \tag{5.42}
 \end{aligned}$$

When the phases are random and uniform, on the other hand, by (5.5), they must satisfy

$$\begin{aligned}
 & \sum_{m \in \mathcal{C}(n, t-1)} h_{mn}^2 P_m(t) = \gamma \\
 \Rightarrow P_m(t) &= \frac{\gamma}{\sum_{m \in \mathcal{C}(n, t-1)} h_{mn}^2} \\
 &\leq \frac{\gamma M(t)^\alpha}{|\mathcal{C}(n, t-1)|} \\
 &= O\left(4^{(\alpha/2-1)t} N^{(\kappa-1)\alpha/2} \right). \tag{5.43}
 \end{aligned}$$

After receiving estimates from the other sub-clusters, each node updates its estimate

by taking the sum:

$$\begin{aligned} z_n(t) &= \frac{1}{4} \sum_{\mathcal{C}(n,t-1) \subset \mathcal{C}(n,t)} z_{\mathcal{C}(n,t-1)} \\ &= \frac{1}{4^{t-T} N} \sum_{m \in \mathcal{C}(n,t)} z_m(0), \end{aligned}$$

where the second equality follows straightforwardly by induction. At time t , the identical estimate at each cluster is a weighted average of the measurements from within that cluster.

Consensus is achieved at round T , where the four sub-clusters at level $t = T - 1$ broadcast their estimates to the entire network. Evaluating (5.44) for $t = T$, note that hierarchical averaging achieves perfect consensus; there is no need for a tolerance parameter ϵ . This somewhat surprising result is the consequence of combining the flexibility of wireless, which permits the adjustment of network connectivity at will, with the simplifying assumption of infinite-rate links. In the next section I will revisit this assumption.

In the following theorem I derive the resource requirements of hierarchical averaging.

Theorem 5.5. *With high probability, the resource consumption of hierarchical averaging scales according to*

$$T_\epsilon = B_\epsilon = O(N^\kappa), \tag{5.44}$$

$$E_\epsilon = \begin{cases} O(N^{1-\alpha/2+\kappa\alpha/2}), & \text{for fixed phase} \\ O(N^{\kappa\alpha/2}), & \text{for uniform phase} \end{cases}, \tag{5.45}$$

for any path-loss exponent $2 \leq \alpha < 4$, for any $\epsilon > 0$ and for any $\kappa > 0$.

Proof. The bound on T_ϵ follows by construction; I chose $T = \lceil \log_4 N^{1-\kappa} \rceil = O(N^\kappa)$

layers of hierarchy and constructed the algorithm such that consensus is achieved to within any tolerance $\epsilon > 0$,

I derive the bound on B_ϵ by examining the cardinality of the neighborhoods for each node. At time slot $t = 1$, by (5.40) each node transmits at power $P_n(1) = O(N^{(\kappa-1)\alpha/2})$. The neighborhood size of each node therefore scales as the number of nodes in a circle of radius $O(N^{\kappa-1})$. By Lemma 5.1, this number is $|\mathcal{N}_n(1)| = O(N^\kappa)$ with probability approaching 1 as $N \rightarrow \infty$. Thus $B(1) = O(N^\kappa)$.

For rounds $2 \leq t \leq T$, it is necessary to bound the number of *clusters* in range of each node. In (5.42) the transmit powers were chosen such that the clusters transmit to each node in a circle of area $\pi M^2(t) = O(4^t N^{1-\kappa})$. By construction, each cluster $\mathcal{C}(n, t)$ covers an area of $O(4^t N^{1-\kappa})$. Therefore, the number of clusters that can fit into the circle is constant, so $B(t) = O(1)$. Summing over all rounds yields

$$B_\epsilon = \sum_{t=1}^T B(t) = O(N^\kappa) + \sum_{t=2}^T O(1) = O(N^\kappa). \quad (5.46)$$

Finally, I derive the bounds on E_ϵ . For fixed phase, (5.40) and (5.42) imply

$$\begin{aligned} E_\epsilon &= \sum_{t=1}^T \sum_{n=1}^N P_n(t) \\ &= N \cdot O(N^{(\kappa-1)\alpha/2}) + N \sum_{n=2}^T O(4^{(\alpha/2-2)t} N^{-\alpha/2+\kappa(\alpha/2-2)}) \\ &\leq O(N^{1-\alpha/2+\kappa\alpha/2}) + O\left(N^{1-\alpha/2+\kappa(\alpha/2-2)} \sum_{t=0}^{T-1} 4^{(\alpha/2-2)t}\right) \\ &= O(N^{1-\alpha/2+\kappa\alpha/2}) + O\left(N^{1-\alpha/2+\kappa(\alpha/2-2)} \frac{1 - 4^{(\alpha/2-2)T}}{1 - 4^{\alpha/2-2}}\right) \end{aligned} \quad (5.47)$$

$$= O(N^{1-\alpha/2+\kappa\alpha/2}) + O(N^{1-\alpha/2+\kappa(\alpha/2-2)} N^{(\alpha/2-2)(1-\kappa)}) \quad (5.48)$$

$$= O(N^{1-\alpha/2+\kappa\alpha/2}) + O(N^{-1})$$

$$= O(N^{1-\alpha/2+\kappa\alpha/2}), \quad (5.49)$$

where (5.47) follows from the finite geometric sum identity, and (5.48) holds only when $\alpha \leq 4$. For uniform phase, note first that the condition in (5.43) is more strict than that of (5.40), so

$$P_n(t) = O\left(4^{(\alpha/2-1)t} N^{(\kappa-1)\alpha/2}\right) \quad (5.50)$$

for all n, t . Substituting (5.50) into the definition of E_ϵ yields

$$E_\epsilon = \sum_{t=1}^T \sum_{n=1}^N P_n(t) \quad (5.51)$$

$$= O\left(N^{1+(\kappa-1)\alpha/2} \sum_{t=0}^{T-1} 4^{(\alpha/2-1)t}\right) \quad (5.52)$$

$$= O\left(N^{1-\alpha/2+\kappa\alpha/2} (N^{\alpha/2-1} - 1)\right) \quad (5.53)$$

$$= O(N^{\kappa\alpha/2}), \quad (5.54)$$

where again I have employed the finite geometric sum identity. \square

Hierarchical averaging achieves resource scaling arbitrarily close to the lower bound of Theorem 5.1 when phase is fixed. When phase is uniform, however, the energy consumption is strictly suboptimal for $\alpha > 2$. Note that the resource scaling does not depend on the channel phases for $\alpha = 2$. For free-space propagation, hierarchical averaging is order optimal regardless of phase.

5.4.3 Numerical Results

I next examine the empirical performance of the several consensus algorithms presented. Choosing $\gamma = 10\text{dB}$, $\alpha = 4$, $\epsilon = 10^{-4}$, and $\kappa = 0$, I let N run from 10 to 1000, averaging performance over 20 random initializations for each value of N . Figure 5.2 displays the average transmit energy E_ϵ and time-bandwidth product B_ϵ . (Since the

data for T_ϵ are rather similar to that of B_ϵ , I do not plot them.)

With respect to time-bandwidth product, hierarchical averaging performs best, the required number of sub-channel uses growing slowly with N . The remaining two schemes perform comparably, the required number of sub-channels growing approximately linearly in N . Note that, while I bounded the time-bandwidth product of path averaging with a strictly sub-linear term, this bound applied to hypothetical instantiations of the scheme in which multiple transmissions occur simultaneously. These simulations used the ordinary algorithm, which requires $\Theta(N)$ sub-channel uses.

With respect to total transmit energy, hierarchical averaging performs best so long as the phases are fixed, in which case the performance is on par order-wise with the lower bound. When phases are uniform, however, performance depends on N . Even though path averaging has better scaling than hierarchical averaging under uniform phase, for small N hierarchical averaging requires less power. Finally, as expected, randomized gossip requires the most energy in any regime.

5.5 Quantization

In this section I examine consensus with quantization. As in the case with ideal links, I first characterize the performance of existing quantized consensus algorithms with respect to the metrics specified in Section 5.1.3. It is impossible to survey every approach in the literature, so I focus on the *quantized consensus* of [64], in which consensus is modified to preserve the average of quantized estimates each round. After deriving bounds on its performance, I turn to hierarchical averaging, showing that it achieves the lower bound of Theorem 5.2 when phases are fixed.

5.5.1 Quantized Consensus

In ordinary gossip, the primary difficulty of quantization is that quantizing estimates in general alters the average across the network. Thus, even if consensus is achieved, the dynamics will not in general converge on the true average of the (quantized) measurements. In *quantized consensus* [64], a family of consensus algorithms is proposed that preserves the average at each round; it converges to near-consensus around the true average.

Recall from Section 5.1.3 that \mathcal{Z} is the set of L points evenly distributed across $[0, 1)$, separated by quantization bin width $\Delta = 1/L$. Quantized consensus operates only on quantized values, so first nodes must quantize the real-valued measurements $z_n(0)$:

$$q_n(0) = \phi(z_n(0)), \quad (5.55)$$

where ϕ is the dithered quantizer described in Section 5.1.3. Let $e_n(0) = \phi(z_n(0)) - z_n(0)$ denote the quantization error.

Much like in randomized gossip, at each round every node randomly selects a neighboring node and mutually averages, with the caveat that one node rounds “up” to the nearest member of \mathcal{Z} while the other rounds “down.” Letting i and j denote the two nodes in the exchange, the dynamics¹ are

$$q_i(t) = \left\lceil \frac{q_i(t-1) + q_j(t-1)}{2} \right\rceil_{\mathcal{Z}} \quad (5.56)$$

$$q_j(t) = \left\lfloor \frac{q_i(t-1) + q_j(t-1)}{2} \right\rfloor_{\mathcal{Z}}, \quad (5.57)$$

where $\lceil \cdot \rceil_{\mathcal{Z}}$ and $\lfloor \cdot \rfloor_{\mathcal{Z}}$ represent rounding up and down to the nearest element of \mathcal{Z} , respectively. In [64, Theorem 1] this algorithm is proven to converge on near-consensus:

¹In fact, [64] proposes a family of algorithms, and the one used here is only one possibility. The convergence properties exploited in the following are independent of the specific algorithm chosen.

in the limit, each $q_n(t)$ differs by at most a single bin, and the sum of the quantized measurements is preserved. It is difficult to bound the convergence speed of this process in general due to the non-linearity of the updates. However, for the case of a fully-connected graph, in [64, Lemma 6] it is shown that quantized consensus requires $\Omega(N^2)$ transmissions over $\Omega(N)$ consensus rounds. Using this fact, I bound the overall performance.

Theorem 5.6. *The performance of quantized gossip scales, with high probability, as*

$$T = \Omega(N), \tag{5.58}$$

$$B = \Omega(N \log N), \tag{5.59}$$

$$E = \Omega(N^{2-\alpha/2+x}(\log N)^{\alpha/2}), \text{ and} \tag{5.60}$$

$$\sigma^2 = \Omega(N^{-2x-1}), \tag{5.61}$$

for any $x \geq 0$.

Proof. Choose an alphabet size L , which may vary with N . Then, in order to maintain connectivity, links must have signal-to-noise ratios $\gamma = \Theta(L)$ at radius $\sqrt{\log N/N}$, which implies

$$P_n(t) = \Theta \left(L \left(\frac{\log N}{N} \right)^{\alpha/2} \right). \tag{5.62}$$

By [64, Lemma 6], consensus requires $\Omega(N)$ rounds for fully-connected graphs, and the performance for random graphs cannot be any better. As in the proof of unquantized randomized gossip, the neighborhood size scales as $\Theta(\log N)$, so the time-bandwidth product scales as

$$B = \Omega(N \log N). \tag{5.63}$$

Since $\Omega(N^2)$ total transmissions are required for consensus,

$$E = \Omega(LN^{2-\alpha/2}(\log N)^{\alpha/2}). \quad (5.64)$$

Finally, I examine the mean-squared error. In the best case, the dynamics converge on true consensus, meaning that $q_n(T)$ is the same for each n . In this case the final estimates are merely the average of the quantized measurements $z_n(0)$. Therefore,

$$\begin{aligned} q_n(T) &= \frac{1}{N} \sum_{n=1}^N q_n(0) \\ &= \frac{1}{N} \sum_{n=1}^N (z_n(0) - e_n(0)) \\ &= z_{\text{ave}} - \frac{1}{N} \sum_{n=1}^N e_n(0), \end{aligned}$$

where $e_n(0)$ is the quantization error of the initial estimate. In the worst case, each $|e_n(0)| = \Delta/2 = L^{-1}/2$. Since the errors are uncorrelated, the squared error follows

$$\begin{aligned} \sigma^2 &= E \left[\left(\frac{1}{N} \sum_{n=1}^N e_n(0) \right)^2 \right] \\ &\geq \frac{1}{N^2} \sum_{n=1}^N E[|e_n(0)|^2] \\ &= N^{-1} L^{-2}/2 \\ &= \Omega(N^{-1} L^{-2}). \end{aligned}$$

Choosing $L = N^x$ gives the result. □

I hasten to point out that the bounds here are rather generous, since they presuppose that convergence on a random graph is as fast as on a fully-connected graph. In practice, as will be shown in numerical results presented later, the performance is

somewhat worse.

5.5.2 Hierarchical Averaging

Here I characterize the performance of hierarchical averaging with quantization. As before, cells of nodes at lower layers achieve local consensus, after which they broadcast their estimates to nearby clusters, continuing the process until global consensus is achieved. Here, however, each estimate is quantized prior to transmission, which introduces error that accumulates during consensus.

As in the previous subsection, nodes employ the dithered quantizer. For the uniform quantization alphabet with cardinality L , let the quantized version of the estimate $z_n(t)$ be denoted

$$q_n(t) = \phi(z_n(t)). \quad (5.65)$$

Each quantized value can be written as

$$q_n(t) = z_n(t) + v_n(t), \quad (5.66)$$

where each $v_n(t)$ is uniform over $[-\Delta/2, \Delta/2)$ and independent for every n, t .

Choose $T = \lceil \log_4 N^{1-\kappa} \rceil$ and define the cells $\mathcal{C}_{jk}(t)$ as before. At time slot $t = 1$, each node n quantizes its initial measurement $z_n(0)$ and broadcasts the quantized value to the nodes in $\mathcal{C}(n, 1)$. Following (5.40), this requires

$$P_n(1) = O(LN^{(\kappa-1)\alpha/2}), \quad (5.67)$$

where the dependence on L arises since $\gamma = \Theta(L)$ and L , and therefore γ , may depend on N . Each node n updates its estimate by averaging the quantized estimates in its

cluster:

$$z_n(1) = \frac{1}{4^{1-T}N} \sum_{m \in \mathcal{C}(n,1)} q_m(0) \quad (5.68)$$

$$= \frac{1}{4^{1-T}N} \sum_{m \in \mathcal{C}(n,1)} z_m(0) + v_m(0). \quad (5.69)$$

As before nodes use the normalization factor $1/4^{1-T}N$ in order to avoid nodes' needing to know the cluster cardinalities. Next, at time slot $2 \leq t \leq T$, each cluster at layer $t-1$ quantizes its estimate and cooperatively broadcasts to the members of its parent cluster at layer t . Following (5.42) and (5.43), this requires

$$P_n(t) = \begin{cases} O(L4^{(\alpha/2-2)t}N^{-\alpha/2+\kappa(\alpha/2-2)}), & \text{for fixed phases} \\ O(L4^{(\alpha/2-1)t}N^{(\kappa-1)\alpha/2}), & \text{for uniform phases} \end{cases}. \quad (5.70)$$

At time step $t = 2$, each node n averages together the estimates from each of the subclusters $\mathcal{C}(m, t-1) \subset \mathcal{C}(n, t)$, yielding

$$\begin{aligned} z_n(2) &= \frac{1}{4} \sum_{\mathcal{C}(m,1) \subset \mathcal{C}(n,2)} q_{\mathcal{C}(m,1)}(1) \\ &= \frac{1}{4} \sum_{\mathcal{C}(m,1) \subset \mathcal{C}(n,2)} z_{\mathcal{C}(m,1)}(1) + v_{\mathcal{C}(m,1)}(1) \\ &= \frac{1}{4} \sum_{\mathcal{C}(m,1) \subset \mathcal{C}(n,2)} \left(\frac{1}{4^{1-T}N} \sum_{k \in \mathcal{C}(m,1)} z_k(0) + v_k(0) \right) + v_m(1) \\ &= \frac{1}{4^{2-T}N} \sum_{k \in \mathcal{C}(n,2)} (z_k(0) + v_k(0)) + \frac{1}{4} \sum_{\mathcal{C}(m,1) \subset \mathcal{C}(n,2)} v_{\mathcal{C}(m,1)}(1). \end{aligned}$$

Continuing by induction, at arbitrary round t the estimate is

$$z_n(t) = \frac{1}{4^{t-T}N} \sum_{k \in \mathcal{C}(n,t)} (z_k(0) + v_k(0)) + \sum_{s=1}^{t-1} \sum_{M \in R_n(t,s)} 4^{s-t} v_M(s), \quad (5.71)$$

where $R_n(t, s)$ is the set of all clusters $\mathcal{C}(m, s)$ that are subsets of $\mathcal{C}(n, t)$. In other words, at round t nodes have the total average so far, corrupted by quantization noise from each of round $s < t$.

In the following theorem, I detail the resource-estimate tradeoff achieved by this scheme.

Theorem 5.7. *Using dithered quantization, hierarchical averaging achieves the following tradeoff between resource consumption and estimation error with high probability:*

$$T = B = O(N^\kappa) \quad (5.72)$$

$$E = \begin{cases} O(N^{1-\alpha/2+\kappa\alpha/2+x}), & \text{for fixed phases} \\ O(N^{\kappa\alpha/2+x}), & \text{for uniform phases} \end{cases} \quad (5.73)$$

$$\sigma^2 = \Theta(N^{-2x}), \quad (5.74)$$

for any $x \geq 0$, $\kappa > 0$, and $2 \leq \alpha < 4$. In particular, for $x = 0$ the estimation error is constant in the network size using the same amount of energy as in the non-quantized case.

Proof. Choose an alphabet size L . Since the number of rounds and the cluster geometry is unchanged from the non-quantized case, I repeat the argument from Theorem 5.5, yielding $T = B = O(N^\kappa)$. Since the transmit power is changed only by a factor of L , I can repeat the arguments from Theorem 5.5, which yields

$$E = \begin{cases} O(LN^{1-\alpha/2+\kappa\alpha/2}), & \text{for fixed phases} \\ O(LN^{\kappa\alpha/2}), & \text{for uniform phases} \end{cases}.$$

All that remains is to bound the estimation error. Evaluating (5.71) for $t = T$,

for every n

$$\begin{aligned} z_n(T) &= \frac{1}{N} \sum_{k=1}^N (z_k(0) + v_k(0)) + \sum_{s=1}^{T-1} \sum_{M \in R_n(t,s)} 4^{s-T} v_M(s) \\ &= z_{\text{ave}} + \sum_{s=0}^{T-1} \sum_{M \in R_n(t,s)} 4^{s-T} v_M(s). \end{aligned}$$

The mean squared estimation error is therefore

$$\begin{aligned} \sigma^2 &= E \left[\left| \sum_{s=0}^{T-1} \sum_{M \in R_n(t,s)} 4^{s-T} v_M(s) \right|^2 \right] \\ &= \sum_{s=0}^{T-1} \sum_{M \in R_n(t,s)} 4^{2(s-T)} E[|v_M(s)|^2], \end{aligned}$$

where the equality is due to the independence of the quantization error terms. Since each $v_M(s)$ is uniformly distributed across $[-\Delta, \Delta)$, $E[|v_m(2)|^2] = \Theta(\Delta^2) = \Theta(L^{-2})$. Therefore,

$$\begin{aligned} \sigma^2 &= \Theta(L^{-2}) \sum_{s=0}^{T-1} \sum_{M \in R_n(t,s)} 4^{2(s-T)} \\ &= \Theta(L^{-2}) \sum_{s=0}^{T-1} 4^{T-s} 4^{2(s-T)} \\ &= \Theta(L^{-2}) \sum_{s=0}^{T-1} 4^{s-T} \\ &= \Theta(L^{-2} N^{-1}) \frac{1 - 4^T}{1 - 4} \\ &= \Theta(L^{-2}), \end{aligned}$$

since $4^T = \Theta(N)$. Choosing $L = N^x$ yields the result. □

5.5.3 Numerical Results

Here I examine the empirical performance of the quantized consensus discussed. I also run simulations for randomized gossip, employing dithered quantization to accommodate the finite-rate links. I again choose $\gamma = 10\text{dB}$, $\kappa = 0$, let N run from 10 to 1000, and average performance over 20 initializations, but here $\alpha = 2$. Choosing γ constant means that the quantization error Δ is constant in N , and the minimum quantization error is itself constant. Figure 5.3 shows the energy E plotted against the mean-square error σ^2 .

The energy expenditure for hierarchical averaging is consistent with theory; however, note that uniform phase results in higher expenditure than fixed phase, even though the scaling laws are the same. The energy expenditure for randomized gossip increases roughly linearly in N , suggesting that the energy burden with fixed γ is similar to the non-quantized case. As expected, quantized consensus performs worse than predicted by Theorem 5.6. The energy consumption is on par with randomized gossip, but it accrues estimation error as N increases. The other schemes have bounded or decreasing error.

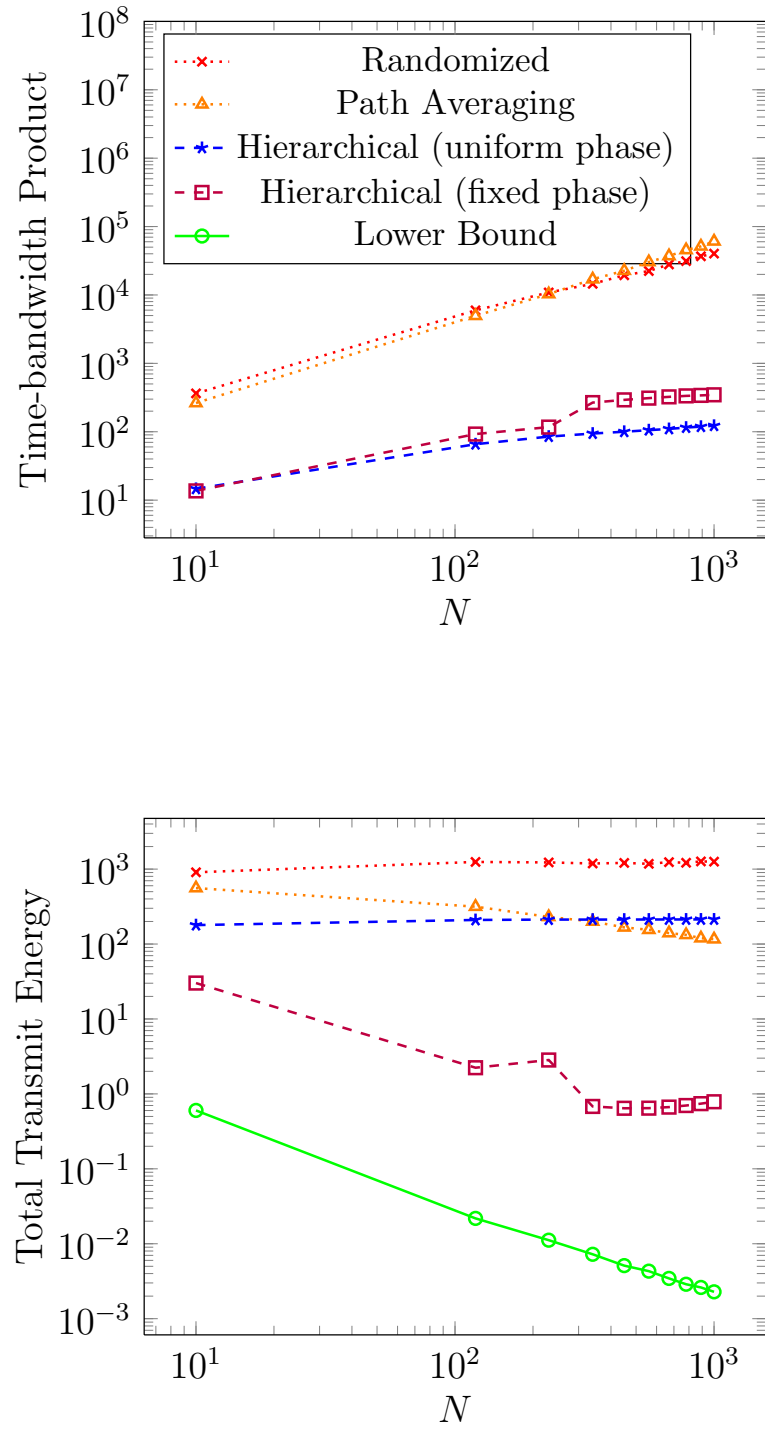


Figure 5.2: Transmit energy E_ϵ and time-bandwidth product B_ϵ for a variety of consensus algorithms.

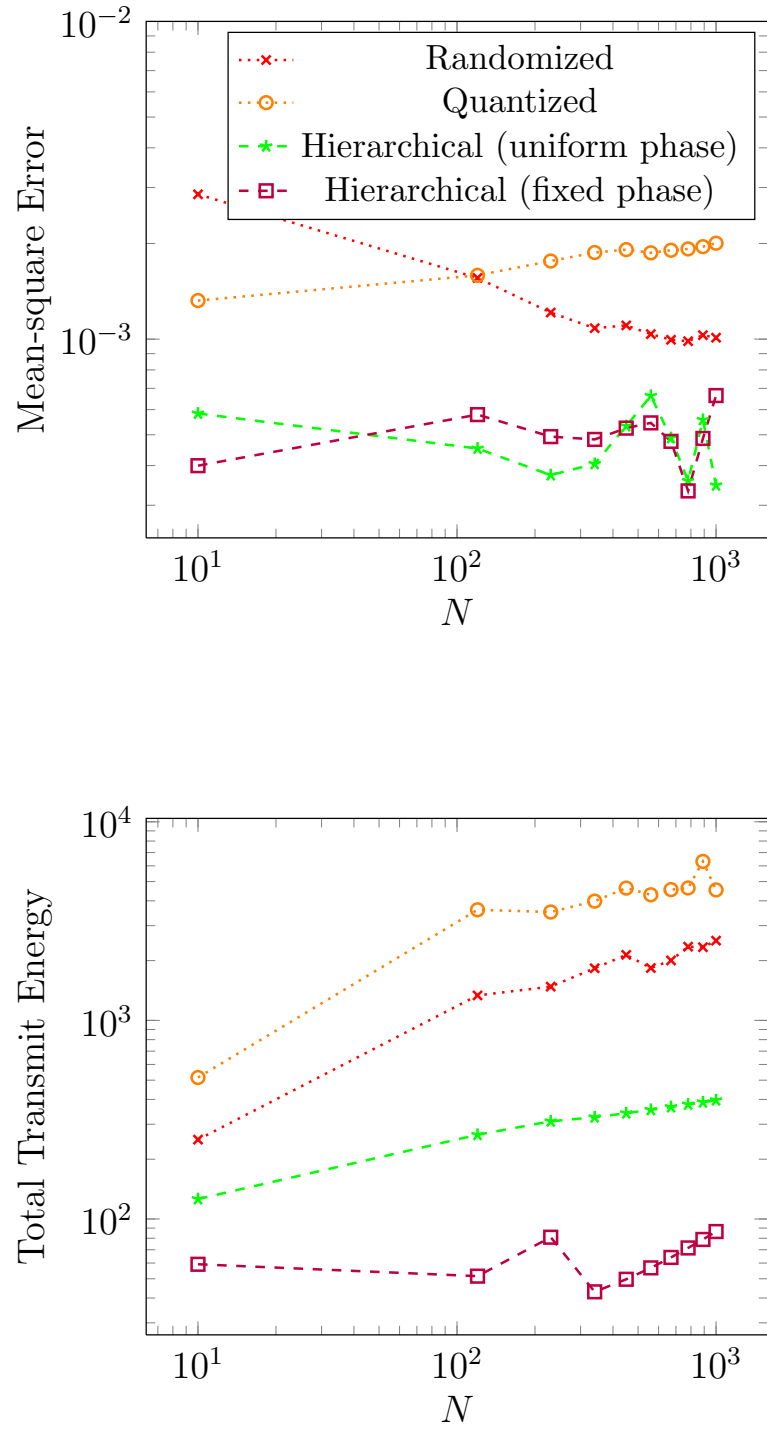


Figure 5.3: Total energy E and mean-square error σ^2 for several quantized consensus algorithms.

Conclusions

6.1 Summary

In this thesis, I have studied computation problems from an explicitly wireless perspective. I have considered both situations in which terminals desire to compute functions of others' digital messages—as exemplified by network coding—and situations in which terminals desire to compute a function of sources—as exemplified by averaging consensus. In both cases, I proposed cooperative schemes. For suitable scenarios, these cooperative schemes are provably near-optimal in a few senses: in terms of diversity-multiplexing tradeoff, in terms of capacity to within a constant gap, or in terms of order optimality.

In Chapter 3, I studied the impact of user cooperation on physical-layer network coding. Constructing a lattice-coding version of block Markov encoding, I presented a strategy that introduces a “decode-and-forward” element into computation coding. Transmitters decode each other's messages, enabling them to transmit resolution information cooperatively to the receivers. This strategy achieves higher computation rates than previous approaches, since transmitters can jointly encode part of their messages, and coherent signals benefit from a beamforming gain. Furthermore, coop-

eration enables an improvement in the diversity-multiplexing tradeoff. For the 2×1 channel, the proposed approach is DMT optimal, achieving the same tradeoff as a two-antenna MISO channel. For more general $M \times 1$ channels, cooperation garners a DMT improvement, obtaining full diversity, but it falls short of the tradeoff of the equivalent MISO channel. For channels with multiple receivers, however, the DMT is mostly unknown, although the proposed approach provides an improvement in achievable rates at finite SNR.

In Chapter 4, I studied the impact of relay cooperation on physical-layer network coding. I considered a two-transmitter, two-receiver network aided by a dedicated relay. I found that the benefits depend on the relay modality. For a standard relay, in which the relay transmission depends on signals received previously, a compress-and-forward scheme improves the achievable rate, but does not provably improve the degrees-of-freedom. For an instantaneous relay, in which the relay transmission may depend on signals *currently* being received, however, an amplify-and-forward scheme provides substantial gains. Amplify-and-forward is provably optimal in the degrees-of-freedom for almost every channel realization. Furthermore, for symmetric channel gains, it obtains computation rates that differ from capacity by only a constant gap for a non-trivial range of channels.

Finally, in Chapter 5 I studied consensus under a wireless framework. I proposed a simple path-loss model which captures the broadcast and superposition properties inherent to wireless communication, and I defined resource consumption in terms of energy, time, and bandwidth. Under this model, I studied existing consensus strategies, showing that while they may be order-optimal with respect to the amount of energy required to achieve consensus, they are strictly suboptimal with respect to the time and bandwidth required. Additionally, I proposed *hierarchical averaging*, a cooperative approach to consensus designed explicitly for the wireless medium. For

free-space propagation (i.e. the path-loss coefficient $\alpha = 2$), hierarchical averaging is nearly order optimal with respect to all three metrics. For $2 < \alpha \leq 4$, optimality depends on assumptions about channel phase. If phases are taken to be constant, hierarchical averaging remains nearly optimal; if they are taken to be random, uniformly distributed, and independent, hierarchical averaging is suboptimal in the required transmit energy. Furthermore, I studied the effects of quantization. Using dithered quantization, I showed that, without expending any additional energy over the non-quantized case, hierarchical averaging suffers estimation error that is constant in the size of the network.

6.2 Future Directions

The ideas explored in this thesis, of course, have applications beyond the problems studied herein. The following are a few areas in which these ideas show promise:

- **Practical codes for the relay channel:** The decode-and-forward rates for the three-terminal relay channel have been known for decades, and they even achieve rates within a constant gap of capacity. Although several approaches for *binary* or discrete-input relay channels, as yet practical, near-capacity codes for the AWGN channels do not exist. The lattice block Markov strategy introduced in Chapter 2 and exploited for computation in Chapter 3 may, in principle, be used to construct such codes. Low-density lattice codes, proposed by Sommer et al. [75], have been shown to have low encoding and decoding complexity as well as near-optimal performance, much like LDPC codes over discrete channels. Conceptually, it is straightforward to decompose an LDLC into resolution and vestigial codebooks and carry out block Markov encoding. In practice, of course, there is a plethora of details to work out before such an approach can

be implemented. As of this writing, initial investigations into this approach are underway. They show that, at least under a few scenarios, the proposed scheme indeed provides low-complexity codes with near-capacity performance.

- **Diversity-multiplexing tradeoff of the cooperative computation channel:** In Chapter 3, I fully characterized the DMT of the 2×1 channel. When there are more than two transmitters or, worse, more than one receiver, the DMT is only partially characterized. The primary obstacle to complete DMT characterization is the difficulty in aligning channels to suitable integer combinations of lattice points. However, as mentioned in the introduction, Niesen and Whiting [44] proposed a scheme that combines compute-and-forward with interference alignment and achieves the optimum degrees-of-freedom for the non-cooperative computation channel. Since their scheme involves linear/lattice codes, it is again possible to construct a block Markov version of the scheme via decomposition into resolution and vestigial codebooks. I anticipate that such a scheme would be DMT optimal for the 2×2 channel, and would garner a DMT gain generally for channels with multiple receivers.
- **Computation over layered networks:** The amplify-and-forward scheme of Chapter 4 achieved rates within a constant gap of capacity for the symmetric, instantaneous relay computation channel. It is likely, however, that these results can be extended to more general topologies. Lee and Jafar [97] show that gains in interference channels with an instantaneous relay can be adapted to *layered* interference channels. In the case of computation, an alternating amplify-and-forward/compute-and-forward scheme seems fruitful: at one layer, terminals amplify incoming messages by carefully chosen coefficients in order to align the effective messages at the next layer to a suitable set of integer combinations of lattice codewords. It is straightforward to prove that such a scheme

is sufficient to achieve the optimum degrees-of-freedom of a three-layer network; in the case of symmetric channels, arguments similar to those used in Chapter 4 may establish constant-gap bounds on computation capacity. Furthermore, this scheme can be extended to networks with arbitrarily many layers without loss of performance. So long as every other layer successfully decodes suitable finite-field linear combinations of messages, noise is not amplified in the network, and the performance is essentially identical to that of a network with fewer layers.

- **Information-theoretic treatment of consensus performance:** The model over which the resource cost of consensus is studied in Chapter 5 entails several assumptions made for the sake of tractability. Most critically, interference of sufficiently low power is neglected, and terminals are restricted to simple, one-shot quantization. I envision a more comprehensive research program which relaxes these assumptions and provides an information-theoretic analysis of the performance bounds of consensus. Each terminal possesses a k -length source according to an arbitrary distribution, and desires the element-wise average of all N sources up to some distortion criterion. The performance metric is the *consensus capacity*, defined as the limiting ratio of the number of channel uses needed to achieve consensus and the source length k . Such an asymptotic analysis permits the use of powerful source coding tools, including rate-distortion theory, Wyner-Ziv encoding, etc. Coupling such tools with hierarchical averaging may again yield order-optimal performance. If not, a further search for near-optimal strategies will doubtless shed important insight on wireless sensor networks.

This is but a partial list of possible extensions of the work presented in this thesis. In general, the philosophy of cooperative computation—whether by means of the decomposition of structured codes, hierarchical partitioning of networks, or

both—can be applied to problems across the board in wireless networks.

Mutual information of dithered lattices over the multiple-access channel

Here I prove that the mutual information between dithered lattice codewords and any receiver approaches that of a Gaussian multiple-access channel.

Lemma A.1. *Let*

$$\mathbf{x}_l = \sqrt{P}[\lambda_l + \mathbf{t}_l] \bmod \Lambda_s \quad (\text{A.1})$$

be a collection of independent lattice codewords, dithered across the shaping lattice, for $1 \leq l \leq L$. Let

$$\mathbf{y} = \sum_{l=1}^L h_l \mathbf{x}_l + \mathbf{n}, \quad (\text{A.2})$$

be a noisy sum of the codewords, where the noise \mathbf{n} has i.i.d. elements with variance σ^2 . Then, for any set $\mathcal{B} \in \{1, \dots, L\}$, the normalized mutual information between the transmit signals and the receive signal approaches at least that of a Gaussian multiple-access channel:

$$\lim_{n \rightarrow \infty} \frac{1}{n} I(\mathbf{x}_{\mathcal{B}}; \mathbf{y} | \mathbf{x}_{\mathcal{B}^c}) \geq \frac{1}{2} \log_2 \left(1 + \frac{P \sum_{l \in \mathcal{B}} h_l^2}{\sigma^2} \right). \quad (\text{A.3})$$

When \mathbf{n} is Gaussian, this bound is tight.

Proof. Since \mathbf{y} is the sum of transmitted signals, conditioning entails only subtracting away the known component. Therefore, letting

$$\mathbf{y}_{\mathcal{B}} = \sum_{l \in \mathcal{B}} h_l \mathbf{x}_l + \mathbf{n}, \quad (\text{A.4})$$

the mutual information is

$$\lim_{n \rightarrow \infty} \frac{1}{n} I(\mathbf{x}_{\mathcal{B}}; \mathbf{y} | \mathbf{x}_{\mathcal{B}^c}) = \lim_{n \rightarrow \infty} \frac{1}{n} I(\mathbf{x}_{\mathcal{B}}; \mathbf{y}_{\mathcal{B}}) = \lim_{n \rightarrow \infty} \frac{1}{n} (h(\mathbf{y}_{\mathcal{B}}) - h(\mathbf{n})), \quad (\text{A.5})$$

where $h(\cdot)$ is the differential entropy. Since the Gaussian distribution maximizes the differential entropy for a given variance,

$$\frac{1}{n} h(\mathbf{n}) \leq \frac{1}{2} \log(2\pi e \sigma^2). \quad (\text{A.6})$$

To bound $h(\mathbf{y}_{\mathcal{B}})$, note that in [36, Lemma 8] it was shown that the density function $f_{\mathbf{y}_{\mathcal{B}}}$ is bounded by

$$f_{\mathbf{y}_{\mathcal{B}}} \leq e^{c(n)n} f_{\mathbf{y}^*}, \quad (\text{A.7})$$

where \mathbf{y}^* is an i.i.d. Gaussian vector with variance $P \sum_{l \in \mathcal{B}} h_l^2 + \sigma^2$, and $c(n)$ is a term approaching zero from above as $n \rightarrow \infty$. Plugging this into the definition of

differential entropy, yields, for sufficiently high n ,

$$\frac{1}{n}h(\mathbf{y}_B) \geq -\frac{1}{n} \int e^{c(n)n} f_{\mathbf{y}^*} \log(e^{c(n)n} f_{\mathbf{y}^*}) \quad (\text{A.8})$$

$$= -\frac{1}{n} e^{c(n)n} \int f_{\mathbf{y}^*} \log(f_{\mathbf{y}^*}) - \frac{1}{n} e^{c(n)n} c(n)n \quad (\text{A.9})$$

$$= e^{c(n)n} \left(\frac{1}{n} h(\mathbf{y}^*) - c(n) \right) \quad (\text{A.10})$$

$$\geq \frac{1}{n} h(\mathbf{y}^*) - c(n) \quad (\text{A.11})$$

$$\rightarrow \frac{1}{n} h(\mathbf{y}^*) \quad (\text{A.12})$$

$$= \frac{1}{2} \log \left(2\pi e \left(P \sum_{l \in \mathcal{B}} + \sigma^2 \right) \right), \quad (\text{A.13})$$

where (A.11) follows because $e^{c(n)n} \geq 1$ and for sufficiently high n the term $\frac{1}{n}h(\mathbf{y}^*) - c(n)$ is positive. Combining (A.6) and (A.13), yields

$$\lim_{n \rightarrow \infty} \frac{1}{n} I(\mathbf{x}_B; \mathbf{y} | \mathbf{x}_{B^c}) \geq \frac{1}{2} \log \left(2\pi e \left(P \sum_{l \in \mathcal{B}} h_l^2 + \sigma^2 \right) \right) - \frac{1}{2} \log(2\pi e \sigma^2) \quad (\text{A.14})$$

$$= \frac{1}{2} \log_2 \left(1 + \frac{P \sum_{l \in \mathcal{B}} h_l^2}{\sigma^2} \right). \quad (\text{A.15})$$

When \mathbf{n} is Gaussian, it is well-known that Gaussian inputs are optimal and result in the same mutual information as the bounds just established. In this case the bound is tight. \square

References

- [1] C. Shannon, “A mathematical theory of communication,” *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948. 1.1
- [2] —, “Communication in the presence of noise,” *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949. 1.1
- [3] A. Goldsmith and P. Varaiya, “Capacity of fading channels with channel side information,” *Information Theory, IEEE Transactions on*, vol. 43, no. 6, pp. 1986–1992, 1997. 1.1
- [4] R. McEliece and W. Stark, “Channels with block interference,” *IEEE Trans. Inform. Theory*, vol. 30, no. 1, pp. 44 – 53, Jan. 1984. 1.1
- [5] G. Caire, G. Taricco, and E. Biglieri, “Optimum power control over fading channels,” *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1468 –1489, Jul. 1999. 1.1
- [6] E. Telatar, “Capacity of multi-antenna Gaussian channels,” *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 585–595, 1999. 1.1
- [7] C. Shannon, “Coding theorems for a discrete source with a fidelity criterion,” *IRE Nat. Conv. Rec*, vol. 4, no. 142-163, 1959. 1.1
- [8] C. Berrou and A. Glavieux, “Turbo codes,” *Encyclopedia of Telecommunications*, 2003. 1.1
- [9] R. Gallager, “Low-density parity-check codes,” *Information Theory, IRE Transactions on*, vol. 8, no. 1, pp. 21–28, 1962. 1.1
- [10] D. MacKay and R. Neal, “Near Shannon limit performance of low density parity check codes,” *Electronics letters*, vol. 32, no. 18, p. 1645, 1996. 1.1
- [11] E. Arıkan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” *IEEE Trans. Info. Theory*, vol. 55, no. 7, pp. 3051–3073, 2009. 1.1

-
- [12] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *Information Theory, IEEE Transactions on*, vol. 23, no. 3, pp. 337–343, 1977. 1.1
 - [13] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York, NY: John Wiley and Sons, Ltd., 2006. 1.1, 5.1.3
 - [14] E. C. van der Meulen, "Three-terminal communication channels," *Advanced Applied Probability*, vol. 3, pp. 120–154, 1971. 1.1
 - [15] T. Cover and A. E. Gamal, "Capacity theorems for the relay channel," *IEEE Trans. Inform. Theory*, vol. 25, no. 5, pp. 572–584, Sept. 1979. 1.1, 2, 2.5, 2.5, 4.2
 - [16] R. Etkin, D. Tse, and H. Wang, "Gaussian interference channel capacity to within one bit," *IEEE Trans. Inform. Theory*, vol. 54, no. 12, pp. 5534–5562, Dec. 2008. 1.1
 - [17] O. Ordentlich, U. Erez, and B. Nazer, "The approximate sum capacity of the symmetric gaussian k-user interference channel," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 2072–2076. 1.1, 1.2.1
 - [18] A. Avestimehr, S. Diggavi, and D. Tse, "Wireless network information flow: A deterministic approach," *IEEE Trans. Inform. Theory*, vol. 57, no. 4, pp. 1872–1905, Apr. 2011. 1.1, 1.2.1
 - [19] S. H. Lim, Y.-H. Kim, A. E. Gamal, and S.-Y. Chung, "Noisy network coding," *submitted to IEEE Trans. Inform. Theory*, 2010. [Online]. Available: <http://arxiv.org/abs/1002.3188v2> 1.1, 1.2.1
 - [20] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Info. Theory*, vol. 19, no. 4, pp. 471–480, 1973. 1.1
 - [21] T. Berger, "Multiterminal source coding," *The Information Theory Approach to Communications*, vol. 229, pp. 171–231, 1977. 1.1
 - [22] S. Tung, "Multiterminal source coding," Ph.D. dissertation, Cornell University, May 1978. 1.1
 - [23] A. Wagner, S. Tavildar, and P. Viswanath, "Rate region of the quadratic Gaussian two-encoder source-coding problem," *IEEE Trans. Info. Theory*, vol. 54, no. 5, pp. 1938–1961, 2008. 1.1
 - [24] T. Courtade and T. Weissman, "Multiterminal source coding under logarithmic loss," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 761–765. 1.1

-
- [25] R. Koetter, M. Effros, and M. Médard, “A theory of network equivalence—part i: Point-to-point channels,” *IEEE Trans. Info. Theory*, vol. 57, no. 2, pp. 972–995, 2011. 1.1
 - [26] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, “Network information flow,” *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000. 1.1, 1.2.1
 - [27] F. Benezit, A. Dimakis, P. Thiran, and M. Vetterli, “Order-optimal consensus through randomized path averaging,” *Information Theory, IEEE Transactions on*, vol. 56, no. 10, pp. 5150–5167, oct. 2010. 1.1, 1.2.2, 5.3, 5.3.2, 5.3.2
 - [28] K. I. Tsianos and M. G. Rabbat, “Multiscale gossip for efficient decentralized averaging in wireless packet networks,” *submitted to IEEE Trans. Info. Theory*, 2010. 1.1, 1.2.2, 1.3.2, 1, 5.4
 - [29] S.-Y. Li, R. Yeung, and N. Cai, “Linear network coding,” *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003. 1.2.1
 - [30] R. Koetter and M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, oct. 2003. 1.2.1
 - [31] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006. 1.2.1
 - [32] R. Dougherty, C. Freiling, and K. Zeger, “Insufficiency of linear coding in network information flow,” *IEEE Trans. Inform. Theory*, vol. 51, no. 8, pp. 2745–2759, Aug. 2005. 1.2.1
 - [33] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, “XORs in the air: practical wireless network coding,” in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 243–254. 1.2.1
 - [34] S. Katti, S. Gollakota, and D. Katabi, “Embracing wireless interference: analog network coding,” in *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4. ACM, 2007, pp. 397–408. 1.2.1
 - [35] S. Zhang, S. C. Liew, and P. P. Lam, “Physical-layer network coding,” in *Proc. Mobicom*. New York, NY, USA: ACM, 2006, pp. 358–365. 1.2.1
 - [36] B. Nazer and M. Gastpar, “Compute-and-forward: Harnessing interference through structured codes,” *IEEE Trans. Inform. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011. 1.2.1, 2.2, 2.4, 2.4, 3.1.2, 3.2.2, 3.2.2, 3.2.2, 3.3.1, 4.1.2, A
 - [37] —, “Reliable physical layer network coding,” *Proc. IEEE*, vol. 99, no. 3, pp. 438–460, Mar. 2011. 1.2.1

-
- [38] K. Narayanan, M. P. Wilson, and A. Sprintson, "Joint physical layer coding and network coding for bi-directional relaying," in *Proc. Allerton*, Monticello, IL, Sept. 2007. 1.2.1
 - [39] W. Nam, S.-Y. Chung, and Y. Lee, "Capacity of the Gaussian two-way relay channel to within $\frac{1}{2}$ bit," *IEEE Trans. Inform. Theory*, vol. 56, no. 11, pp. 5488–5494, Nov. 2010. 1.2.1
 - [40] D. Gündüz, A. Yener, A. J. Goldsmith, and H. V. Poor, "The multi-way relay channel," *submitted to IEEE Trans. Inform. Theory*, vol. abs/1004.2434, 2010. 1.2.1
 - [41] L. Ong, C. Kellett, and S. Johnson, "Capacity theorems for the AWGN multi-way relay channel," in *Proc. Int. Symp. Information Theory (ISIT)*, Austin, TX, June 2010, pp. 664–668. 1.2.1
 - [42] J. Zhan, U. Erez, M. Gastpar, and B. Nazer, "MIMO compute-and-forward," in *Proc. Int. Symp. Information Theory (ISIT)*, Seoul, Korea, July 2009, pp. 2848–2852. 1.2.1
 - [43] J. Zhan, B. Nazer, U. Erez, and M. Gastpar, "Integer-forcing linear receivers: A new low-complexity MIMO architecture," in *Proc. Vehicular Technology Conference*, Taipei, Taiwan, Sept. 2010. 1.2.1
 - [44] U. Niesen and P. Whiting, "The degrees of freedom of compute-and-forward," in *Proc. Int. Symp. Information Theory (ISIT)*, Aug. 2011, pp. 1081–1085. 1.2.1, 3.1.3, 4.1.3, 4.2, 6.2
 - [45] M. Maddah-Ali, A. Motahari, and A. Khandani, "Communication over MIMO X channels: Interference alignment, decomposition, and performance analysis," *IEEE Trans. Inform. Theory*, vol. 54, no. 8, pp. 3457–3470, Aug. 2008. 1.2.1
 - [46] V. Cadambe and S. Jafar, "Interference alignment and degrees of freedom of the k -user interference channel," *IEEE Trans. Inform. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008. 1.2.1
 - [47] B. Nazer, M. Gastpar, S. A. Jafar, and S. Vishwanath, "Ergodic interference alignment," *submitted to IEEE Trans. Inform. Theory*, Jan. 2009. [Online]. Available: <http://arxiv.org/abs/0901.4379v2> 1.2.1
 - [48] A. S. Motahari, S. O. Gharan, M.-A. Maddah-Ali, and A. K. Khandani, "Real interference alignment: Exploiting the potential of single antenna systems," *submitted to IEEE Trans. Inform. Theory*, Aug. 2009. [Online]. Available: <http://arxiv.org/abs/0908.2282v2> 1.2.1

-
- [49] G. Cybenko, “Dynamic load balancing for distributed memory multiprocessors,” *Journal of Parallel and Distributed Computing*, vol. 7, no. 2, pp. 279 – 301, 1989. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/074373158990021X> 1.2.2
 - [50] J. N. Tsitsiklis, “Problems in decentralized decision making and computation,” Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, Nov. 1984. 1.2.2
 - [51] J. Tsitsiklis, D. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Trans. Automatic Control*, vol. 31, no. 9, pp. 803 – 812, Sept. 1986. 1.2.2
 - [52] J. Duchi, A. Agarwal, and M. Wainwright, “Dual averaging for distributed optimization: Convergence analysis and network scaling,” *IEEE Trans. Automatic Control*, vol. 57, no. 3, pp. 592 – 606, Mar. 2012. 1.2.2
 - [53] V. Saligrama, M. Alanyali, and O. Savas, “Distributed detection in sensor networks with packet loss and finite capacity links,” *IEEE Trans. Signal Processing*, vol. 54, no. 11, pp. 4118–4132, Nov. 2006. 1.2.2
 - [54] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, “Approximate distributed Kalman filtering in sensor networks with quantifiable performance,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, 2005. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1147685.1147711> 1.2.2
 - [55] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Trans. Info. Theory*, vol. 52, no. 6, pp. 2508 – 2503, June 2006. 1.2.2, 5.1.1, 5.3, 5.3.1
 - [56] A. D. G. Dimakis, A. D. Sarwate, and M. J. Wainwright, “Geographic gossip: Efficient averaging for sensor networks,” *IEEE Trans. Signal Processing*, vol. 56, no. 3, pp. 1205–1216, March 2008. 1.2.2
 - [57] D. Ustebay, B. Oreshkin, M. Coates, and M. Rabbat, “Rates of convergence for greedy gossip with eavesdropping,” in *Proc. Allerton*, Monticello, IL, 2008. 1.2.2
 - [58] T. C. Aysal, M. E. Yildiz, A. D. Sarwate, and A. Scaglione, “Broadcast gossip algorithms for consensus,” *IEEE Trans. Signal Processing*, vol. 57, no. 7, pp. 2748–2761, July 2009. 1.2.2
 - [59] B. Nazer, A. G. Dimakis, and M. Gastpar, “Local interference can accelerate gossip algorithms,” *IEEE J. Selected Topics Signal Proc.*, vol. 5, no. 4, pp. 876–887, Aug. 2011. 1.2.2

-
- [60] S. Vanka, V. Gupta, and M. Haenggi, "Power-delay analysis of consensus algorithms on wireless networks with interference," *Int. J. Systems, Control, and Communications*, vol. 2, no. 3, pp. 256–274, 2010. 1.2.2
 - [61] S. Kar, S. Aldosari, and J. Moura, "Topology for distributed inference on graphs," *Signal Processing, IEEE Transactions on*, vol. 56, no. 6, pp. 2609–2613, june 2008. 1.2.2
 - [62] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33 – 46, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0743731506001808> 1.2.2
 - [63] S. Kar and J. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise," *IEEE Trans. Signal Processing*, vol. 57, no. 1, pp. 355–369, Jan. 2009. 1.2.2
 - [64] A. Kashyap, T. Basar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192 – 1203, 2007. 1.2.2, 5.5, 5.5.1, 5.5.1, 1, 5.5.1
 - [65] F. Benezit, P. Thiran, and M. Vetterli, "The distributed multiple voting problem," *IEEE J. Selected Topics Signal Proc.*, vol. 5, no. 4, pp. 791–804, Aug. 2011. 1.2.2
 - [66] T. Aysal, M. Coates, and M. Rabbat, "Distributed average consensus with dithered quantization," *Signal Processing, IEEE Transactions on*, vol. 56, no. 10, pp. 4905–4918, oct. 2008. 1.2.2, 5.1.3
 - [67] M. Yildiz and A. Scaglione, "Coding with side information for rate-constrained consensus," *IEEE Trans. Signal Processing*, vol. 56, no. 8, pp. 3753–3764, Aug. 2008. 1.2.2
 - [68] J. Forney, G.D., "Coset codes - I: Introduction and geometrical classification," *IEEE Trans. Info. Theory*, vol. 34, no. 5, pp. 1123–1151, Sept. 1988. 1.2.3
 - [69] —, "Coset codes - II: Binary lattices and related codes," *IEEE Trans. Info. Theory*, vol. 34, no. 5, pp. 1152–1187, Sept. 1988. 1.2.3
 - [70] R. de Buda, "Some optimal codes have structure," *IEEE J. Select Areas Commun.*, vol. 7, no. 6, pp. 893–899, Aug. 1989. 1.2.3
 - [71] R. Urbanke and B. Rimoldi, "Lattice codes can achieve capacity on the AWGN channel," *IEEE Trans. Inform. Theory*, vol. 44, no. 1, pp. 273–278, Jan. 1998. 1.2.3
 - [72] G. Poltyrev, "On coding without restrictions for the AWGN channel," *IEEE Trans. Inform. Theory*, vol. 40, no. 2, pp. 409–417, Mar. 1994. 1.2.3, 2.1, 2.5

-
- [73] H.-A. Loeliger, “Averaging bounds for lattices and linear codes,” *IEEE Trans. Inform. Theory*, vol. 43, no. 6, pp. 1767–1773, Nov. 1997. 1.2.3, 2.2
 - [74] U. Erez and R. Zamir, “Achieving $\frac{1}{2} \log(1 + \text{SNR})$ on the AWGN channel with lattice encoding and decoding,” *IEEE Trans. Inform. Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004. 1.2.3, 2.2, 2.3, 2.3, 2.5, 3.2.1, 3.2.2, 3.2.2
 - [75] N. Sommer, M. Feder, and O. Shalvi, “Low-density lattice codes,” *IEEE Trans. Info. Theory*, vol. 54, no. 4, pp. 1561–1585, July 2008. 1.2.3, 6.2
 - [76] R. Zamir, S. Shamai, and U. Erez, “Nested linear/lattice codes for structured multiterminal binning,” *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1250–1276, June 2002. 1.2.3
 - [77] D. Krithivasan and S. Pradhan, “Lattices for distributed source coding: Jointly Gaussian sources and reconstruction of a linear function,” *IEEE Trans. Inform. Theory*, vol. 55, no. 12, pp. 5628–5651, Dec. 2009. 1.2.3
 - [78] A. Wagner, “On distributed compression of linear functions,” *IEEE Trans. Inform. Theory*, vol. 57, no. 1, pp. 79–94, Jan. 2011. 1.2.3
 - [79] X. He and A. Yener, “Providing secrecy with structured codes: Tools and applications to two-user Gaussian channels,” *submitted to IEEE Trans. Inform. Theory*, 07 2009. [Online]. Available: <http://arxiv.org/abs/0907.5388v1> 1.2.3
 - [80] S. Agrawal and S. Vishwanath, “On the secrecy rate of interference networks using structured codes,” in *Proc. Int. Symp. Information Theory (ISIT)*, Seoul, Korea, July 2009, pp. 2091–2095. 1.2.3
 - [81] J.-C. Belfiore and F. Oggier, “Secrecy gain: A wiretap lattice code design,” in *Int. Symp. on Information Theory and its Applications (ISITA)*, Taichung, Taiwan, Oct. 2010, pp. 174–178. 1.2.3
 - [82] W. Nam, S.-Y. Chung, and Y. H. Lee, “Nested lattice codes for Gaussian relay networks with interference,” *submitted to IEEE Trans. Inform. Theory*, Feb. 2009. [Online]. Available: <http://arxiv.org/abs/0902.2436v1> 1.2.3
 - [83] A. Ozgur and S. Diggavi, “Approximately achieving Gaussian relay network capacity with lattice codes,” in *Proc. Int. Symp. Information Theory (ISIT)*, Austin, TX, June 2010, pp. 669–673. 1.2.3
 - [84] M. Nokleby and B. Aazhang, “Lattice coding over the relay channel,” in *Proc. Int. Conf. Commun.*, Kyoto, Japan, June 2011. 1.2.3
 - [85] Y. Song and N. Devroye, “Lattice codes for the Gaussian relay channel: Decode-and-forward and compress-and-forward,” *submitted to IEEE Trans. Inform. Theory*, Nov. 2011. [Online]. Available: <http://arxiv.org/abs/1111.0084v1> 1.2.3

-
- [86] H. Weingarten, Y. Steinberg, and S. Shamai, "The capacity region of the Gaussian multiple-input multiple-output broadcast channel," *IEEE Trans. Inform. Theory*, vol. 52, no. 9, pp. 3936–3964, Sept. 2006. 1.3.1, 3.2.1, 3.2.1
 - [87] A. Ozgur, O. Leveque, and D. Tse, "Hierarchical cooperation achieves optimal capacity scaling in ad hoc networks," *IEEE Trans. Info. Theory*, vol. 53, no. 10, pp. 3549–3572, Oct. 2007. 1.3.2, 5.1, 5.4
 - [88] C. A. Rogers, "Lattice coverings of space," *Mathematika*, vol. 6, no. 1, pp. 33–39, 1959. 2.1
 - [89] U. Erez, S. Litsyn, and R. Zamir, "Lattices which are good for (almost) everything," *IEEE Trans. Inform. Theory*, vol. 51, no. 10, pp. 3401–3416, Oct. 2005. 2.1
 - [90] D. Krithivasan and S. S. Pradhan, "A proof of the existence of good nested lattices." [Online]. Available: <http://www.eecs.umich.edu/techreports/systems/cspl/cspl-384.pdf> 2.2
 - [91] G. D. Forney, Jr., "On the role of MMSE estimation in approaching the information-theoretic limits of linear Gaussian channels: Shannon meets Wiener," in *Proc. Allerton*, Monticello, IL, Oct. 2004. 1
 - [92] L. Zheng and D. Tse, "Diversity and multiplexing: a fundamental tradeoff in multiple-antenna channels," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, pp. 1073–1096, May 2003. 3.1.3, 3.3.1, 3.3.2
 - [93] M. Nokleby and B. Aazhang, "Unchaining from the channel: Cooperative computation over multiple-access channels," in *Proc. Information Theory Workshop (ITW)*, Paraty, Brazil, Oct. 2011, pp. 593–597. 1
 - [94] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *Information Theory, IEEE Transactions on*, vol. 22, no. 1, pp. 1 – 10, Jan 1976. 4.2
 - [95] A. El Gamal and N. Hassanpour, "Relay-without-delay," in *Proc. Int. Symp. Information Theory (ISIT)*, Sept. 2005, pp. 1078 –1080. 4.3, 4.3
 - [96] H. Chang and S.-Y. Chung, "Capacity of strong and very strong Gaussian interference relay-without-delay channels," *submitted to IEEE Trans. Info. Theory*, 2011. 4.3
 - [97] N. Lee and S. Jafar, "Aligned interference neutralization and the degrees of freedom of the 2 user interference channel with instantaneous relay," *submitted to IEEE Trans. Info. Theory*, Feb. 2011. 4.3, 6.2

- [98] F. Kuhn and R. Wattenhofer, “On the complexity of distributed graph coloring,” in *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, ser. PODC '06. New York, NY, USA: ACM, 2006, pp. 7–15. [Online]. Available: <http://doi.acm.org/10.1145/1146381.1146387> 5.1.1
- [99] M. Penrose, *Random geometric graphs*. Oxford, UK: Oxford University Press, 2003, vol. 5. 5.2
- [100] P. Gupta and P. Kumar, “Critical power for asymptotic connectivity,” in *Proc. Conference on Decision and Control*, vol. 1, 1998, pp. 1106 –1110. 5.3