Lattice Coding Increases Multicast Rates for Gaussian Multiple-Access Networks

Bobak Nazer and Michael Gastpar
University of California, Berkeley
Wireless Foundations Research Center, Dept. of EECS
Berkeley, CA 94720-1770, USA
Email: bobak, gastpar@eecs.berkeley.edu

***Abstract*— We study the multicasting problem for additive white Gaussian noise (AWGN) networks with both point-to-point and multiple-access links. The recent discovery of network coding has shown that routing is suboptimal for networks of point-to-point channels: messages must be mixed at intermediate nodes. We showed in earlier work that if a network includes multiple-access channels (MACs) then converting these channels into bit pipes with a channel-network separation-based approach is also suboptimal: the MACs should be used to compute functions reliably as part of an overall network code. In this paper, we give a lattice-based coding strategy for reliably computing linear functions over Gaussian MACs. We then give a computable statement of rates achieved by the resulting network code. In many interesting cases, our achieved rates are higher than those accessible to a separation-based scheme. Our results show that structured codes can be used to derive achievable rates for multi-user communication problems that have been outside the reach of standard i.i.d. random coding arguments.**

I. Introduction

The celebrated paper of Ahlswede et al. showed that routing is insufficient to achieve the multicast capacity of point-to-point channel networks [1]. Instead, some intermediate nodes in the network should only send out a function of their incoming messages. The receivers, given an appropriate set of functions can successfully decode the original messages. This strategy is referred to as *network coding* and much work has focused on simplifying and linearizing the underlying code construction (see, for instance, [2]–[6]).

If the channel network includes multi-user channels as well as point-to-point links, it is intuitively clear that network coding will be part of the overall solution. It has also been shown that separating channel and network coding is not always possible. For instance, for deterministic broadcast networks, Ratnakar and Kramer showed that only joint channel-network coding is optimal [7]. In earlier work, we showed that if the channel network includes finite field multiple-access channels (MACs), these channels should ideally be used to compute reliable functions of their messages rather than be reduced to bit pipes via channel-network separation [8], [9]. The key insight is that functions well-matched to the *structure* of the probability transition matrix of the MAC can

This work was supported by the National Science Foundation under CAREER Grant CCF-0347298 and NeTS-ProWin Grant CCF-0627024 as well a Graduate Research Fellowship.

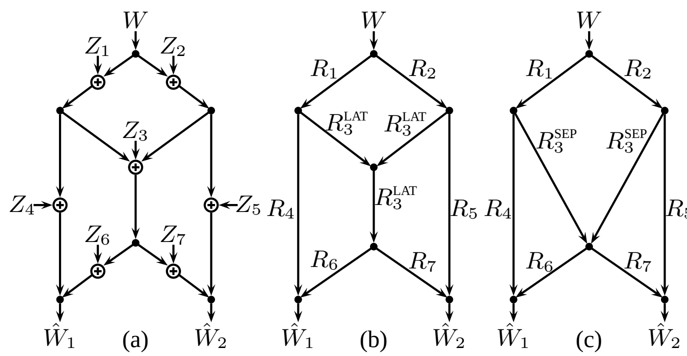

Fig. 1. (a) Example of a Gaussian Multiple-Access Network. (b) Using a lattice-based code, we can achieve any rate on the original network that is reachable on this network. Here, $R_3^{\text{LAT}} = \frac{1}{2}\log\left(\frac{1}{2} + \frac{P}{N_3}\right)$ where P is the per user power of the MAC. (c) With a standard random coding argument, we can only achieve rates on this network where $R_3^{\text{SEP}} = \frac{1}{4}\log\left(1 + \frac{2P}{N_3}\right)$

be reliably computed at a much higher rate using a structured random code. In many cases of interest, the rate increase is proportional to the number of users. This *structural gain* is quite different from *collaborative gain* where the transmitters use message dependencies to access more favorable channel input distributions. For network coding, additive MACs are ideal as they are extremely well suited for reliably computing linear functions. In this paper, we show how to efficiently exploit Gaussian MACs as part of an overall network code.

To take advantage of the structure of Gaussian MACs, we introduce a new strategy: *compute-and-forward*. Nodes in the network that observe the output of a MAC will effectively only decode a linear function of the messages. This allows us to examine a reduced network where the MACs are replaced with nodes with the same connectivity. The capacities of these links will be given by the *linear processing rate* of the MAC. As we will see, any multicast rate achievable on the reduced point-to-point network will be achievable on the original MAC.

Other authors have independently considered using the additive structure of wireless MACs for network coding "on the air." Bhadra, Gupta, and Shakkottai use channel fading and the channel's summation to achieve the multicast capacity of a fading finite-field MAC network as the field size grows [10].

Zhang, Liew, and Lam take a communications perspective and use uncoded transmission to send mod-2 sums of bits at lower bit error rates than possible with separate transmissions [11]. Katti, Gollakota and Katabi simulate a practical system that uses a clever scheme to add bits using the phases transmitted signals [12]. However, from an information theoretic perspective, uncoded transmission cannot be effective over a large network. The error from these uncoded transmissions can quickly build up over several hops and will drive the resulting rate to zero as the noise will become part of the signal we are attempting to relay to the receivers. In order to avoid propagating noise, we need to reliably decode the functions at every node.

Structured codes are essential for efficiently reliably computing functions over MACs. The underlying idea follows from a classic result of Körner and Marton [13]. In their consideration, a central decoder wants to reconstruct the parity of two correlated sources seen by separate encoders. By using the same linear codebook at each encoder, the decoder can recover the parity from the sum of the received codewords. Their work was (to the best of our knowledge) the first situation where structured codes were needed to complete the achievability proof. In earlier work, we showed that linear codes can be used as part of a compute-and-forward strategy for achieving the multicast capacity of a noisy finite field multiple-access network [9]. In this paper, we use lattices as a building block for our overall code showing that structured codes are useful for solving AWGN network problems as well. A similar conclusion was reached in [14] where lattices are used to achieve rates inaccessible to standard i.i.d. random codes for a MAC with interference known at the transmitters. In [15], the authors use lattice codes to increase rates for a two-way relay AWGN problem. In [16], lattices are used to improve the sum rate for the distributed compression of the difference of correlated Gaussians.

The paper is organized as follows. In Section II, we give a formal problem statement. In Section III, we state our main result and outline the proof. In Section IV, we show how to connect the multicast problem to a source-channel problem which is easier to analyze. In Section V, we give definitions and outline useful results from lattice channel and source coding theory. In Section VI, we give our lattice-based code for Gaussian-multiple access networks. In Section VII, we look an AWGN butterfly network as an example. Finally, in Section VIII we discuss open problems and future directions.

II. Problem Statement

We now give a formal problem statement for multicasting information over Gaussian multiple-access networks. Both the point-to-point and the multiple-access channels in the network are AWGN channels. To simplify the statement of our results, we assume that all transmitters face the same power constraint and that the network is acyclic. The signal-to-noise ratio (SNR) for each channel will be set by adjusting the noise variances.

Remark 1: Note that setting all the power constraints to be equal enforces that for a given MAC, each user has the same SNR. This limitation can be removed for a more general statement of our main theorem and we provide a brief sketch of the argument in Section VII. Similarly, we can also characterize cyclic networks but this is also beyond the scope of this paper.

Definition 1: A *point-to-point AWGN channel* is given by:

$$Y[i] = X[i] + Z[i] \tag{1}$$

where $X[i]$, $Z[i]$, and $Y[i]$ are the channel input, noise, and channel output at time i respectively. The noise Z is an i.i.d. Gaussian random variable with mean 0 and fixed variance.

Remark 2: We could replace the point-to-point AWGN channels in the network with any memoryless channels with the same capacities and our main result would remain the same. However, it is convenient to work in terms of Gaussian channels.

Definition 2: A *Gaussian multiple-access channel* is given by:

$$Y[i] = \sum_{j=1}^{J} X_j[i] + Z[i] \tag{2}$$

where $X_1[i], X_2[i], \dots, X_J[i]$, $Z[i]$, and $Y[i]$ are the channel inputs, noise, and channel output at time i respectively. The noise Z is an i.i.d. Gaussian random variable with mean 0 and fixed variance.

Definition 3: An *average power constraint* is given by:

$$\frac{1}{n}\sum_{i=1}^{n}(x[i])^2 \leq P, \tag{3}$$

where $P \in \mathbb{R}_+$ and $x[i] \in \mathbb{R}$ is the i^{th} symbol in the codeword

As stated earlier, all channel inputs face identical transmit power constraints. Also, when we say a label from the integers is *unique* we mean from all previously assigned labels.

Definition 4: A *Gaussian multiple-access network*, $\mathcal{G}_{\text{MAC}}$, consists of the following elements:

1) $\mathcal{V}_N$: the encoder/decoder nodes of the network. Each node, v, has a unique label taken from the positive integers, $v \in \mathbb{Z}_+$, and consists of a decoding function $g_{v_j v}$ for each incoming edge (v_j, v) and an encoding function $f_{v v_k}$ for each outgoing edge (v, v_k).
2) v^S: the source node. One element of $\mathcal{V}_N$. The source sees the message, $W \in \{1, 2, \dots, 2^{nR}\}$.
3) $(v_1^R, v_2^R, \dots, v_L^R)$: the receiver nodes. Each one is an element of $\mathcal{V}_N$.
4) $\mathcal{V}_{\text{MAC}}$: the MACs in the network. Each MAC, m, has a unique integer label, $m \in \mathbb{Z}_+$. Each MAC has a noise variance $N_m \in \mathbb{R}_+$.
5) $\mathcal{E}_{NN}$: the directed point-to-point channels in the network. Each channel has a unique integer label, $e_{NN} \in \mathbb{Z}_+$, and the labels of its inputs and output nodes are given by the functions $v_{\text{IN}}(e_{NN})$ and $v_{\text{OUT}}(e_{NN})$ respectively. The noise variance for the channel is given by $N_e \in \mathbb{R}_+$.
6) $\mathcal{E}_{NM}$: the input edges from nodes to MACs. Each edge has a unique integer label, $e_{NM} \in \mathbb{Z}_+$, and the labels

of its inputting node and destination MAC are given by the functions $v_{\text{IN}}(e_{NM})$ and $v_{\text{OUT}}(e_{NM})$ respectively.

7) $\mathcal{E}_{MN}$: the output edges from a MAC to a node. We assume that each MAC only has an input into one node. Each edge has a unique integer label, $e_{MN} \in \mathbb{Z}_+$, and the label of its MAC and destination node are given by the functions $v_{\text{IN}}(e_{MN})$ and $v_{\text{OUT}}(e_{NM})$ respectively.
8) $X_{v_j v_k}[i]$: the channel input on the edge (v_j, v_k) at time i. The encoders are constrained to only produce channel inputs from time $i = 1$ to time $i = n$. All encoders face the same power constraint for each edge.
9) $Y_{v_j v_k}[i]$: the channel output on the edge (v_j, v_k) at time i.

We also assume that there are a finite number of nodes and channels in the network, $|\mathcal{V}_N| + |\mathcal{V}_{\text{MAC}}| + |\mathcal{E}_{NN}| + |\mathcal{E}_{NM}| + |\mathcal{E}_{MN}| < \infty$.

Definition 5: A multicast rate, R, is *achievable* if $\forall \epsilon \in (0, 1)$ and n large enough there exist encoding and decoding functions for the network such that the average probability of error is less than ϵ:

$$\begin{aligned} &\hat{W}_\ell = f_{v_\ell^R}(Y^n_{v_\ell^R}) \\ &\Pr\left(\{\hat{W}_1 \neq W\} \cup \cdots \cup \{\hat{W}_L \neq W\}\right) < \epsilon, \end{aligned} \tag{4}$$

where $W \in \{1, 2, \ldots, 2^{nR}\}$.

Definition 6: The *multicast capacity* is the supremum of all achievable multicast rates.

Definition 7: A *point-to-point AWGN network*, $\mathcal{G}_{\text{POINT}} = (\mathcal{V}_N, \mathcal{E}_{NN})$, is just a Gaussian multiple-access network without any multiple-access nodes, $\mathcal{V}_{\text{MAC}} = \mathcal{E}_{NM} = \mathcal{E}_{MN} = \emptyset$.

Definition 8: A *unit bit pipe network*, $\mathcal{G}_{\text{PIPE}} = (\mathcal{V}, \mathcal{E})$, is just a point-to-point AWGN network except all of the channels, $\mathcal{E}$, are taken to be noiseless bit pipes with unit capacity. The encoding/decoding nodes are given by the set $\mathcal{V}$.

Our scheme will give achievable rates for any $\mathcal{G}_{\text{MAC}}$. We express the achievable rate through a new point-to-point AWGN network that results from an appropriate transformation of our original network. The achievable rate is then given by the multicast capacity of the point-to-point network. We will also demonstrate that in some cases our achievable rates coincide with the simple upper bound due to the max-flow min-cut theorem of Ford and Fulkerson.

We now briefly review some results for multicasting over point-to-point channel networks. In [1], it was shown that for a unit bit pipe network the multicast capacity is given by the max-flow min-cut theorem. For each receiver, calculate the maximum information flow across all cuts that separate the source node from that receiver. The multicast capacity is the minimum of all these max-flow values taken over all cuts and receivers. In [2] and [3], it was shown that linear encoding and decoding over a finite field is sufficient to achieve the multicast capacity. Bounds are also given on the required field size. It was independently and concurrently shown by Ho et al. in [17], Jaggi et al. in [5], and Sanders et al. [6] that the field size only needs to be larger than the number of receivers. We reproduce the version from [17] below as it will be useful to us in proving our main theorems.

Definition 9: Let $\mathcal{G}_{\text{PIPE}} = (\mathcal{V}, \mathcal{E})$ and let $\mathbb{F}_q$ be a finite field of size q. An *algebraic network code* is a set of linear functions for a unit bit pipe network. Specifically, each encoding function at a node v is constrained to be a linear function of its observations from each incoming edge:

$$X_v[i] = \sum_j \alpha_{vj} Y_{vj}[i]. \tag{5}$$

where $Y_{vj}[i]$ is the value seen by node v at time i on its j^{th} incoming edge and $Y_{vj}[i], \alpha_{vj} \in \mathbb{F}_q$ for all $v \in \mathcal{V}$.

Lemma 1 (Ho et al.): Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a unit bit pipe network with a single source and L receivers. The multicast capacity is given by the max-flow min-cut bound and can be achieved by an algebraic network code over any finite field larger than L ($\mathbb{F}_q, \quad q > L$).

For a full proof, see [4]. In [18] it was shown that there is a channel-network separation theorem for multicasting over point-to-point channel networks. In fact, it is straightforward to find an algebraic network code for a point-to-point channel network. This is captured in the following lemma which will be useful for the main theorem.

Lemma 2: Choose any $\delta > 0$. Given any point-to-point channel network, $\mathcal{G}_{\text{POINT}} = (\mathcal{V}_N, \mathcal{E}_{NN})$, with multicast capacity C, employing capacity-achieving channel codes on each link coupled with an appropriate algebraic network code can achieve any multicast rate $R = C - \delta$.

Proof: Let the capacity of each channel $c \in \mathcal{E}_{NN}$ be given by C_c. Choose capacity-achieving codes for each channel such that with high probability we get a noiseless channel with rate $\hat{C}_c = C_c - \frac{\delta}{2|\mathcal{E}_{NN}|}$. Call the resulting "noiseless" network, $\hat{\mathcal{G}}$. Now choose $\lambda > 0$ such that:

$$\max_{c \in \mathcal{E}_{NN}} \left(\hat{C}_c - \lambda \left\lfloor \frac{\hat{C}_c}{\lambda} \right\rfloor\right) < \frac{\delta}{2|\mathcal{E}_{NN}|} \tag{6}$$

Create a λ bit pipe network, $\mathcal{G}_{\text{PIPE}} = (\mathcal{V}, \mathcal{E})$, where the nodes are the same as in $\mathcal{G}_{\text{POINT}}$, $\mathcal{V} = \mathcal{V}_N$. For each channel $c \in \mathcal{E}_{NN}$ with capacity C_c in $\mathcal{G}_{\text{POINT}}$, place $\left\lfloor \frac{\hat{C}_c}{\lambda} \right\rfloor$ noise-free channels with capacity Λ in the bit pipe network with the same connectivity. Since all channels in $\mathcal{G}_{\text{PIPE}}$ have the same capacity, we are free to generate an algebraic code that achieves the multicast capacity using Lemma 1. This algebraic code is sent over the channels in $\mathcal{G}_{\text{POINT}}$ using the channel codes chosen above and a timesharing approach. For instance, for a channel with capacity C_c, we consider the n total channel uses in chunks of $\left(\left\lfloor \frac{C_c}{\lambda} \right\rfloor\right)^{-1} n$. Each of these chunks can be used to send λn bits reliably and thus can be used to send one function. Over all cuts in the max-flow min-cut characterization, the largest reduction in rate is at most $\frac{\delta}{2}$ (due to the gap to capacity). Considering the channels only in units of λ also causes at most a $\frac{\delta}{2}$ rate reduction over the worst possible cut. Thus, we can achieve the rate $R = C - \delta$. ∎

Since we are free to choose δ in the proof above we can approach capacity arbitrarily closely with the described scheme.

III. Main Result

In this section, we state our main result and outline the steps in the proof. The naive approach to a Gaussian multiple-access network would be to first convert all channels, including MACs, into bit pipes using standard i.i.d. random coding arguments. Network coding could then be performed over the resulting bit pipes. This ignores the potential of the MAC to reliably compute as part of an overall network code. Our strategy uses a lattice code to reliably compute linear functions over the MACs in the network. This effectively converts the MACs into linear processing units. We can also reduce the network to a bit pipe network but in our case the MACs become nodes. The key question is what rates should we assign to the incoming and outgoing links to these nodes in our reduction. We will call this the *linear processing rate* and we will demonstrate that it is at least:

$$R_{LP} = \frac{1}{2}\log\left(\frac{1}{J_m} + \frac{P}{N_m}\right) \tag{7}$$

where N_m is the noise variance associated with that MAC and J_m is the number of users.

We now give a formal statement of our result.

Theorem 1: Let $\mathcal{G}_{\text{MAC}}$ be an arbitrary Gaussian multiple-access network. We construct a new point-to-point AWGN network, $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$, by the following transformations. First, let the set of encoder/decoder nodes, $\mathcal{V}'$, in the new network be given by the original encoder/decoder nodes as well as the original MACs, $\mathcal{V}' = \mathcal{V}_N \cup \mathcal{V}_{\text{MAC}}$. Second, let the channels in the new network, $\mathcal{E}'$, be given by the original point-to-point channels as well as the input and output edges to the MACs, $\mathcal{E}' = \mathcal{E}_{NN} \cup \mathcal{E}_{NM} \cup \mathcal{E}_{MN}$. The connectivity of these edges is the same as in the original network. Finally, we set the capacity of the edges taken from $\mathcal{E}_{NM}$ and $\mathcal{E}_{MN}$ to be

$$\frac{1}{2}\log\left(\frac{1}{J_m} + \frac{P}{N_m}\right) \tag{8}$$

where N_m is the noise variance of the MAC m associated to these edges in the original network and J_m is the number of users. Any multicast rate achievable on the new network is achievable on the original network. As usual, the maximum achievable multicast rate on a point-to-point channel network is given by the minimum of the min cuts across all receivers.

The proof will proceed in several steps. First, we will show that it is sufficient to consider sending Gaussian sources over the network at specified distortions and then connecting this performance to a bit rate. Next, we will give an achievable scheme for sending linear functions of Gaussian sources over a Gaussian MAC. We will then show that an appropriate network code over the reals exists for our network. Finally, we will show that the desired distortions are achievable.

IV. Achievable Rates from Joint Source-Channel Schemes

In this section, we will show that if we can build a joint source-channel code for multicasting Gaussian sources with distortion D over a Gaussian multiple-access network, then we can also build a channel code for multicasting at rate less than $R(D)$ over the same network. This proof trick does not seem fundamental to the problem but it significantly simplifies the analysis.

Lemma 3: Let $\mathcal{G}_{\text{MAC}}$ be a Gaussian multiple-access channel network. Assume, for all k large enough and $n = k\ell,\ \ell \in \mathbb{Z}_+$ channel network uses, that the source node can transmit an i.i.d. Gaussian sequence, $\mathbf{s}$, of length-k with mean 0 and variance σ_S^2 such that the receivers can make estimates $\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \ldots, \hat{\mathbf{s}}_L$, each with mean-squared error (MSE) D_ℓ. Then for any $\epsilon > 0$, we can design encoders and decoders for the network for multicasting $W \in \{1, 2, \ldots, 2^{NR}\}$ to the receivers such that they can make estimates $\hat{W}_1, \hat{W}_2, \ldots, \hat{W}_L$ such that the average probability of error $P_e = \Pr\left(\{\hat{W}_1 \neq W\} \cup \cdots \cup \{\hat{W}_L \neq W\}\right) < \epsilon$ so long as

$$R < \frac{1}{2\ell}\log\left(\frac{\sigma_S^2}{D_\ell}\right) \tag{9}$$

Proof: Fix k and ℓ. Choose the encoders and decoder in the network such that we achieve the specified distortions at the receivers. We have by the data processing inequality:

$$\frac{k}{2}\log\left(\frac{\sigma_S^2}{D_\ell}\right) \leq I(S^k, ; \hat{S}_\ell^k) \tag{10}$$

$$\leq I(X^n, ; Y_\ell^n). \tag{11}$$

Thus we know that there exists a multiletter input distribution, $p(x^n)$, such that the mutual information to each receiver is lower bounded by the rate-distortion function. We now define supersymbols of length n, $\tilde{X}[i] = [X[(i-1)*n+1]X[(i-1)*n+2]\cdots X[in]]$ and $\tilde{Y}_\ell[i] = [Y_\ell[(i-1)*n+1]Y_\ell[(i-1)*n+2]\cdots Y_\ell[in]]$. The supersymbols $\tilde{X}$ and $\tilde{Y}$ take values in the alphabets $\tilde{\mathcal{X}} = \mathbb{R}^n$ and $\tilde{\mathcal{Y}} = \mathbb{R}^n$ respectively.

Keep all encoders and decoders in the network to be the same as in the distortion-achieving case except for those at the source and the receivers, $v_S, v_1^R, \ldots, v_L^R$. Thus, $p(\tilde{y}_\ell|\tilde{x})$ is a memoryless channel. Generate a random codebook with 2^{NR} length N codewords (one for each message in $\{1, 2, \ldots, 2^{NR}\}$) with each symbol drawn i.i.d. from $\tilde{\mathcal{X}}$ for some $N \in \mathbb{Z}_+$ and $R > 0$. The ℓ^{th} receiver upon seeing $\tilde{Y}_\ell^N$ uses a maximum likelihood rule to infer the original message W. Denote this estimate by $\hat{W}_\ell$. It follows from [19] that for such a channel and N large enough, the average probability of error over codebooks and messages for receiver ℓ, $\bar{\Pr}(\hat{W}_\ell \neq W)$ can be made less than $\frac{\epsilon}{L}$ if $R < I(\tilde{X}; \tilde{Y})$.

It follows from the union bound that the probability that any receiver is in error averaged over all codebooks and messages satisfies:

$$\bar{P}_e \leq \sum_{\ell=1}^{L} \bar{\Pr}(\hat{W}_\ell \neq W) < \epsilon. \tag{12}$$

Finally, we get that there exists at least one fixed codebook with average probability of error, P_e at most $\bar{P}_e$, otherwise the average over all codebooks would not hold. This completes the proof. ■

V. Computation with Lattices

We now state some results on lattices that will be useful in constructing our network code. The essential fact is that there exists lattices which are simultaneously good for both AWGN channel coding and Gaussian source coding. We will also show how to construct *computation codes* from these lattices to compute sums reliably over a Gaussian MAC.

Definition 10: An n-dimensional *lattice*, Λ, is a set of points in $\mathbb{R}^n$ such that if $\mathbf{x}, \mathbf{y} \in \Lambda$, then $\mathbf{x} + \mathbf{y} \in \Lambda$, and if $\mathbf{x} \in \Lambda$, then $-\mathbf{x} \in \Lambda$. A lattice can always be written in terms of a generator matrix $\mathbf{G} \in \mathbb{R}^{n \times n}$:

$$\Lambda = \{\mathbf{x} = \mathbf{z}\mathbf{G} : \mathbf{z} \in \mathbb{Z}^n\} \tag{13}$$

where $\mathbb{Z}$ represents the integers.

Definition 11: An (n, R) *lattice code*, $\mathcal{C}$, is a code with elements taken from the intersection of some n-dimensional lattice Λ, shifted by $\rho_C \in \mathbb{R}^n$, and a convex n-dimensional shape T (which is usually chosen to meet some type of power constraint.)

$$\mathcal{C} = \{\Lambda + \rho_C\} \cap T \tag{14}$$

$$|\mathcal{C}| = 2^{nR} \tag{15}$$

Definition 12: A *lattice quantizer* is a map, $Q : \mathbb{R}^n \to \Lambda$, that sends a point, $\mathbf{x}$, to the nearest lattice point in Euclidean distance after shifting it by some constant, $\rho_S \in \mathbb{R}^n$:

$$\mathbf{x_q} = Q(\mathbf{x} + \rho_S) = \arg\min_{\mathbf{l} \in \Lambda} ||\mathbf{x} + \rho_S - \mathbf{l}||_2 \tag{16}$$

Definition 13: Let $[\mathbf{x}] \bmod \Lambda = \mathbf{x} - Q(\mathbf{x})$. For all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, the mod Λ operation satisfies:

$$[[\mathbf{x} \bmod \Lambda] + \mathbf{y}] \bmod \Lambda = [\mathbf{x} + \mathbf{y}] \bmod \Lambda. \tag{17}$$

As lattices have infinite extent (and thus violate the power constraint), much effort was focused on finding lattices that when intersected with an n-dimensional ball of radius $\sqrt{nP}$ centered at 0 form a good code. Urbanke and Rimoldi showed that such lattices indeed exist in [20]. Further work by Erez and Zamir has focused on proving that decoding to the closest lattice point also achieves capacity [21]. Erez, Litsyn and Zamir showed in [22] that there exist lattices that are simultaneously good source codes and good channel codes. These will be extremely useful in our distributed refinement scheme.

We now consider the problem of transmitting the sum of Gaussian sources over a Gaussian MAC at the minimal mean-squared error. This will be the key to using Gaussian MACs to compute linear functions. Below we give an achievable scheme, computation coding, for refining the sum over many channel uses. This result originally appeared in [9].

Each encoder, $\mathcal{E}_j$, sees an independent identically distributed (i.i.d.) Gaussian sequence $\{S_j[i]\}_{i=1}^k$ with mean 0 and variance σ_S^2. The encoders each face one terminal of a Gaussian MAC with equal transmit power P and noise variance N as in Definition 2. For every k source symbols, we are allotted $n = \ell k$ channel uses where $\ell \in \mathbb{Z}_+$. Our goal is to reconstruct the sum of the sources, $U = S_1 + S_2 + \cdots + S_J$, at the decoder with the lowest possible distortion. Distortion is measured by the usual mean-squared error criterion:

$$D_\ell = \frac{1}{k} \sum_{i=1}^{k} E[(U_i - \hat{U}_i)^2] \tag{18}$$

Theorem 2: The following distortion is achievable[1] for sending k sums of i.i.d. Gaussian sources over a Gaussian MAC with $n = \ell k,\ \ell \in \mathbb{Z}_+$ channel uses:

$$D_\ell = J\sigma_S^2 \left(\frac{N}{N + JP}\right) \left(\frac{JN}{N + JP}\right)^{\ell - 1}. \tag{19}$$

Proof: We provide a brief description of the scheme here but refer the interested reader to [9] for a full proof. In [24], Kochman and Zamir develop an elegant joint source-channel lattice scheme for sending a Wyner-Ziv Gaussian source over a dirty paper channel. Our distributed refinement scheme consists of two main steps. First, we use uncoded transmission to send a noisy sum to the decoder. Then, we have each encoder run a version of the Kochman-Zamir scheme targeted at the desired sum, U. Unfortunately, there is a penalty for this form of distributedness. The lattice at each encoder results in channel outputs that violate the power constraint by a factor of J. Therefore, we must scale down our inputs to meet the power constraint and accept the resulting increase in distortion at the decoder.

Assume $\ell = 2$. We thus have $2k$ channel uses to convey k sums. We will use the first k channel uses for an uncoded transmission phase. The decoder will then form an MMSE estimate $\hat{U}$ of the sum $U = S_1 + \cdots + S_J$ and use this as side information for the next phase. Thus, $U = Q + \hat{U}$ where Q is an i.i.d. Gaussian sequence with mean 0 and variance $J\sigma_S^2 \frac{N}{N+JP}$.

Choose a sequence of good lattices, Λ_k, using [22] and scale them such that the normalized second moment of the lattice is JP. Let $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_J$ be independent dither vectors drawn uniformly over the fundamental Voronoi region, $\mathbf{d}_j \sim \text{Unif}(\mathcal{V}_{0,k})$, and made available to the encoders and decoder.

Each encoder transmits $\frac{1}{\sqrt{J}}\mathbf{x}_j$ where:

$$\mathbf{x}_j = [\gamma \mathbf{s}_j + \mathbf{d}_j] \bmod \Lambda_k. \tag{20}$$

The channel output is given by:

$$\mathbf{y} = \frac{1}{\sqrt{J}} \sum_{j=1}^{J} \mathbf{x}_j + \mathbf{z}.$$

The decoder then computes:

$$\hat{\hat{\mathbf{u}}} = \beta \left[\alpha \mathbf{y} - \left(\sum_{j=1}^{J} \mathbf{d}_j + \gamma \hat{\mathbf{u}}\right)\right] \bmod \Lambda_k + \hat{\mathbf{u}}$$

We define the following constants: $\alpha = \frac{JP\sqrt{J}}{JP+N}$, $\gamma_0 = \sqrt{\frac{JP}{\sigma_Q^2}\left(1 - \frac{JN}{JP+N}\right)}$ and let $\gamma \to \gamma_0$ from below as $k \to \infty$.

[1]In [23], we claimed a slightly lower distortion as achievable. We thank the authors of [15] for pointing out an error in our distortion calculation.

We also set: $\beta = \frac{\sigma_Q^2 \gamma}{JP}$. As $k \to \infty$, we get that the achieved distortion is $D = J\sigma_S^2 \frac{N}{N+JP} \frac{JN}{N+JP}$. For all higher values of ℓ, the scheme can be repeated with the final estimate from the last refinement taken as side information for the next stage. ■

VI. AWGN Network Coding

Before we give a full proof of Theorem 1, we will need two auxiliary results. First, we show that given a full rank system of equations over a prime-sized finite field, the same system of linear equations over the reals is also full rank.

Lemma 4: Let $\mathbb{F}$ be a field of size q where q is prime. Let $\mathbf{A} \in \mathbb{F}^{m \times m}$ be a full rank matrix with elements a_{ij}. Finally, let $\tilde{\mathbf{A}}$ be a matrix in $\mathbb{R}^{m \times m}$ whose elements, $\tilde{a}_{ij}$, are the same as those of $\mathbf{A}, \tilde{a}_{ij} = a_{ij}$. Then, $\tilde{\mathbf{A}}$ is also full rank.

Proof: A matrix is full rank if its determinant is non-zero. One way to express the determinant is through the Leibniz formula. For a matrix over $\mathbb{F}$, this is

$$\det(\mathbf{A}) = \left(\sum_{\sigma \in S_m} \mathrm{sgn}(\sigma) \prod_{i=1}^{m} a_{i\sigma(i)} \right) \mod q \qquad (21)$$

where S_m is the set of all permutations of $\{1, 2, \ldots, m\}$, sgn(σ) is sign of a permutation, and the sum and product are over the reals but are mapped back into $\mathbb{F}$ by the modulo operation. Recall that the sign of a permutation is the 1 if it takes an even number of switches to get from $\{1, 2, \ldots, m\}$ to the permutation σ and -1 otherwise. The determinant of the same matrix over the reals is

$$\det(\tilde{\mathbf{A}}) = \sum_{\sigma \in S_m} \mathrm{sgn}(\sigma) \prod_{i=1}^{m} a_{i\sigma(i)}. \qquad (22)$$

Clearly, $\det(\mathbf{A}) \neq 0$ implies that $\det(\tilde{\mathbf{A}}) \neq 0$ since $b \mod q \neq 0$ is a stronger requirement than $b \neq 0$. ■

Finally, we need to show that given J Gaussian sources at each terminal of a MAC, we can refine a linear function of these sources at the same rate that is available for refining a sum of J i.i.d. Gaussian sources.

Lemma 5: Let $S_1, S_2, \ldots, S_J$ be Gaussian sources with variances $\sigma_1^2, \sigma_2^2, \ldots, \sigma_J^2$ respectively. Let $\sigma_{\text{MAX}}^2 = \max_j \sigma_j^2$. Each source is seen at one encoder with power P which faces a Gaussian MAC with noise variance N. Let q be a positive prime number and let $U = \beta_1 S_1 + \beta_2 S_2 + \cdots + \beta_J S_J$ where $\beta_j \in \{0, 1, 2, \ldots, q-1\}$. Then given k vectors of source symbols and n channel uses where $n = k\ell$, the decoder can make an estimate of U at distortion

$$D_\ell = J(q-1)^2 \sigma_{\text{MAX}}^2 \left(\frac{JN}{N+P} \right)^{\ell}. \qquad (23)$$

Proof: The bulk of the work is done by Theorem 2. To send a linear function, we choose a lattice, Λ, for use in refining a sum with Theorem 2 as if all the sources had variance $(q-1)^2 \sigma_{\text{MAX}}^2$. At each terminal, we quantize $T_j = \beta_j S_j + W_j$ onto the lattice Λ where W_j is an i.i.d. Gaussian random variable available as common randomness to both the encoder and the decoder. Its variance is chosen such that the variance of T_j is matched to the design variance of Λ, $\sigma_{W_j}^2 = (q-1)^2 \sigma_{\text{MAX}}^2 - \beta_j^2 \sigma_j^2$. This lattice point is transmitted and the decoder makes an estimate of the sum $T_1 + T_2 + \cdots + T_J$ and removes the common randomness variables, $W_j, \quad j = 1, 2, \ldots, J$, to get an estimate of U at the desired distortion. Note that as this works in expectation over the W_j, there exist fixed constants $w_1, w_2, \ldots, w_J$ that can serve the same role. ■

Remark 3: Note that the $\left(\frac{N}{N+JP} \right)$ term does not appear in the distortion claimed above. This is because we are not assuming the sources are i.i.d. so we do not make use of the uncoded transmission phase. Due to the dither random variables, independence is unnecessary for the sources. Further, note that as the dithers work in expectation, there exist fixed constants that can serve the same purpose.

We are now ready to prove our main theorem. The proof will proceed in several steps. First, we reduce our original network to a point-to-point channel network. We will then find a set of channel codes and an algebraic network code using Lemma 2 to achieve the multicast capacity of this virtual network. Finally, we will show how to map this overall code onto the original network, especially the MACs.

Proof: [Theorem 1] Choose q to be a prime number such that $q > L$ where L is the number of receivers. We will use the channel network $n = km\ell$ times to convey Gaussian sources of length k. We will then connect the distortion performance back to sending bits using Lemma 3. Construct a new point-to-point channel network using $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ as in the statement of the theorem. Let C' be the multicast capacity of the $\mathcal{G}'$ which is given by the usual max-flow min-cut characterization. We would like to show that for any $\delta, \epsilon > 0$, we can achieve a multicast rate $R = C' - \delta$ on the original network with average probability of error not exceeding ϵ.

First, using Lemma 2, find an algebraic network code for $\mathcal{G}'$ that achieves rate $R = C - \frac{\delta}{2}$. Recall that this involves creating yet another virtual network, $\mathcal{G}_{\text{PIPE}}$, this time one consisting only of λ rate pipes for some appropriately chosen λ. Each of these λ rate bit pipes is designated to carry exactly one linear function. To see how these functions map to both the new point-to-point network and the original network, recall that we can just look at each channel in terms of chunks of channel uses. For instance, for a channel $e \in \mathcal{E}'$ with capacity C_e, we use $\ell \left\lfloor \frac{C_e}{\lambda} \right\rfloor$ chunks of length $\left(\left\lfloor \frac{C_e}{\lambda} \right\rfloor \right)^{-1} mk$. We assume that the blocklengths are large enough so that any loss in chunk assignment due to rounding error is negligible.

Let $P_\lambda, N_\lambda > 0$ be chosen such that $\lambda = \frac{1}{2} \log \left(1 + \frac{P_\lambda}{N_\lambda} \right)$. For each ℓ chunks of channel uses, each node must send out exactly one function of its inputs. The number of inputs is clearly upper bounded by the number of edges in the virtual bit pipe network which itself is upper bounded by:

$$|\mathcal{E}_{\text{UPPER}}| = \left(\max_{e \in \mathcal{E}'} \left\lfloor \frac{C_e}{\lambda} \right\rfloor \right) |\mathcal{E}'| \qquad (24)$$

which is a constant that does not depend on n.

Each node is thus sent a finite number of functions, $U_1^k, U_2^k, \ldots, U_J^k$. Assume that each of these have variance at most σ_U^2. It makes MMSE estimates of these and prepares a new function, $V = \beta_1 U_1 + \beta_2 U_2 + \cdots + \beta_J U_J$ with $\beta_j \in \{0, 1, \ldots, q-1\}$ for each outgoing chunk of channel uses according to the algebraic network code. This function can be transmitted to the receiver with distortion at worst:

$$D_\ell = |\mathcal{E}_{\text{UPPER}}|(q-1)^2 \sigma_U^2 \left(\frac{N_\lambda}{N_\lambda + P_\lambda} \right)^\ell . \quad (25)$$

according to Lemma 5. Note that this holds whether the node in the new network is an actual node in the original network or a MAC. Thus, the processing at a node increases the distortion by at most a factor $|\mathcal{E}_{\text{UPPER}}|(q-1)^2$.

Compute the multicast capacity, C_{PIPE} of $\mathcal{G}_{\text{PIPE}}$. Let $\gamma = \frac{C_{\text{PIPE}}}{\lambda}$ and note that γ is an integer due to the construction of $\mathcal{G}_{\text{PIPE}}$.At the source we will create γ i.i.d. Gaussian sources $S_1^k, S_2^k, \ldots, S_\gamma^k$ of length k and variance 1. These will be relayed to the receivers by means of the algebraic network code and the coding method described above. The receiver will see functions of these original sources at some distortions. First note that the distortions will not exceed $(|\mathcal{V}_N| + |\mathcal{V}_{\text{MAC}}|)|\mathcal{E}_{\text{UPPER}}|(q-1)^2$. This is just the number of processing nodes multiplied by the maximum increase factor due to processing at one node.

The functions of the sources seen at the decoders can be written as a matrix transformation just as the original algebraic network code description. If the algebraic network code induces a transform $\mathbf{A}$ over $\mathbb{F}$ on the sources to a given receiver then the transform for Gaussian case is given by $\tilde{\mathbf{A}}$ which has the same entries as $\mathbf{A}$ but operations are over the reals. Since we assume $\mathbf{A}$ is full rank then $\tilde{\mathbf{A}}$ is full rank as well by Lemma 4. Thus, we can solve for each original source at every receiver at distortion

$$D_\ell = \alpha \left(\frac{N_\lambda}{N_\lambda + P_\lambda} \right)^\ell \quad (26)$$

$$\alpha = \gamma^2 (|\mathcal{V}_N| + |\mathcal{V}_{\text{MAC}}|)|\mathcal{E}_{\text{UPPER}}|(q-1)^2. \quad (27)$$

Finally, we invoke Lemma 3 to get a multicast rate from these distortions. We get that we can achieve any multicast rate satisfying:

$$\begin{aligned} R &< \frac{\gamma}{2\ell} \log \left(\alpha \left(\frac{N_\lambda + P_\lambda}{N_\lambda} \right)^\ell \right) && (28) \\ &= \frac{\gamma}{2} \log \left(1 + \frac{P_\lambda}{N_\lambda} \right) - \frac{\gamma}{2\ell} \log \alpha && (29) \\ &= \gamma\lambda - \frac{\gamma}{2\ell} \log \alpha && (30) \\ &= C_{\text{PIPE}} - \frac{\gamma}{2\ell} \log \alpha && (31) \end{aligned}$$

Choose ℓ large enough such that we can achieve $R = C_{\text{PIPE}} - \frac{\delta}{2}$. Recall that by construction $C_{\text{PIPE}} > C - \frac{\delta}{2}$. Thus, we can achieve a rate $R = C - \delta$ as desired. By making all the appropriate blocklengths large enough, we can make the probability of error arbitrarily small. This completes the proof. ■

Although the proof is somewhat involved, the key fact is that we can refine functions at a certain rate over a MAC. This allows us to eliminate the penalty incurred by increasing function variances over many refinements. We now show that if the noise variance of every channel is the same and no receiver directly views the output of a MAC, then the achievable scheme coincides with the max-flow min-cut bound.

We can generalize the statement of our main theorem to include MACs with unequal power per user. Essentially, we can use a rate splitting approach to create several computational units from the original MAC, each with a different connectivity. For instance, take the minimum input power over all users. We can then take that much power from each user to create a processing node with output rate corresponding to that power. The codewords generated from this can be treated as noise for the next stage which repeats this process with one user removed and all powers suitably reduced. This gives a new type of multiple-access region between different computations at different SNRs.

A closer examination of the proof shows that the main fact is the rate at which the linear function is refined over the Gaussian MAC. This is well-represented by the linear processing rate which is defined below.

Definition 14: The *linear processing rate*, R_{LP}, for a Gaussian MAC is given by:

$$R_{LP} = \lim_{\ell \to \infty} \frac{1}{2\ell} \log \left(\frac{\sigma_S^2}{D_\ell} \right) \quad (32)$$

where D_ℓ is the distortion of a linear function of Gaussian sources at the receiver after ℓ refinements.
This rate is directly connected to the rate at which we can “compute-and-forward” messages. Thus, improving the rate at which a sum of Gaussian sources can be refined with an increasing number of channel uses on a Gaussian MAC will result in a higher achievable multicast rate for an AWGN network.

Overall, we have shown that with the appropriate tools we can exploit the additive function of a Gaussian MAC to compute functions as part of a network code. This approach can be superior to a separation-based approach which decodes all incoming messages rather than just a function of them.

VII. BUTTERFLY

Consider the AWGN channel network in Figure 1 (a). Each vertex on the graph represents a decoder/encoder pair. The sender is at the top of the graph and the two receivers are at the bottom. All encoders must satisfy an average power constraint, $\frac{1}{n}\sum_{i=1}^n x_j[i]^2 \le P$. The $Z_m,\ m = 1, 2, \ldots, 7$ are drawn i.i.d. according to a Gaussian distribution with mean 0 and variance N. This is just the butterfly network considered in [1] except that we have placed a Gaussian MAC in the center.

Lemma 6: The following multicast rate is achievable for the channel network in Figure 1(a) is:

$$R_{LAT} = \frac{1}{2}\log\left(1 + \frac{P}{N}\right) + \max\left(0, \frac{1}{2}\log\left(\frac{1}{2} + \frac{P}{N}\right)\right) \quad (33)$$

The proof is just an application of Theorem 1. The achievable rate for a separation-based scheme is given by:

$$R_{DF} = \frac{1}{2}\log\left(1+\frac{P}{N}\right)+\frac{1}{4}\log\left(1+\frac{2P}{N}\right) \tag{34}$$

The lattice-based strategy takes a performance hit "inside the log" and the separation-based scheme takes the hit on the "pre-log". It is clear from the figure below that when the per user SNR is larger than 1.5, the lattice-based scheme is preferable.

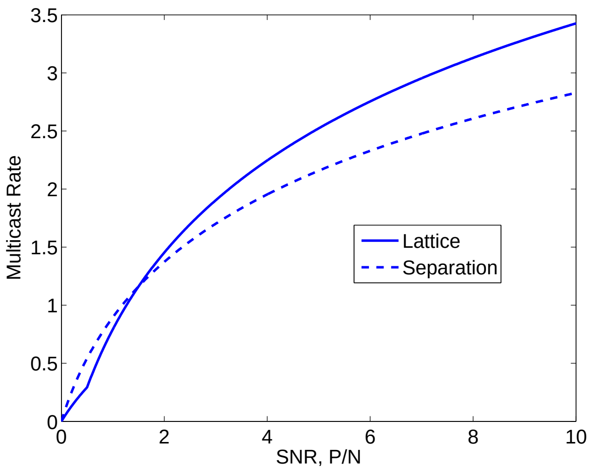

Fig. 2. Multicast Rates for the AWGN Butterfly Network

VIII. Discussion and Open Problems

In this paper, we have found a new characterization of the achievable multicast rate for any Gaussian multiple-access network. These achievable rates are (at least in some cases) higher than those achievable with channel-network separation. Our primary tool was a lattice computation code that allowed us to reliably refine a linear function of Gaussian sources over a Gaussian MAC. Without this coding technique, we would be stuck with the noise buildup from uncoded transmission or a rate penalty associated with the variance inflation as we take sums over the reals. Furthermore, it can be argued that the performance of this computation code is dependent on the structure of the underlying lattice code. Basically, a standard random coding argument cannot be used to construct codes to efficiently compute over MACs as decoding the function requires decoding the individual sources first. The structure of the lattice allows us to sum codewords and decode only the sum of the sources.

It also seems clear that in more general large AWGN networks, structured codes will be required to determine the capacity region. In future work, we will address the multicast capacity of AWGN networks with both broadcast and multiple-access constraints. Beyond network coding, it seems that the lattice techniques developed here could be useful for casting an arbitrary AWGN network into a set of "noisefree" linear equations between the senders and the receivers. The rank of the equations seen at each receiver could then determine the rate transmitted to it.

References

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, July 2000.

[2] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, pp. 371–381, February 2003.

[3] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 782–795, October 2003.

[4] T. Ho, R. Koetter, M. Medard, M. Effros, J. Shi, and D. Karger, "A random linear network coding approach to multicast," *Submitted to IEEE Transactions on Information Theory*, 2006.

[5] S. Jaggi, P. Chou, and K. Jain, "Low complexity algebraic network codes," in *Proceedings of the IEEE International Symposium on Information Theory, Yokohama, Japan*, 2003.

[6] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," in *15th ACM Symposium on Parallel Algorithms and Architectures*, pp. 286–294, 2003.

[7] N. Ratnakar and G. Kramer, "The multicast capacity of deterministic relay networks with no interference," *IEEE Transactions on Information Theory*, vol. 52, pp. 2425–2432, June 2006.

[8] B. Nazer and M. Gastpar, "Computing over multiple-access channels with connections to wireless network coding," in *IEEE ISIT 2006, Seattle, WA*.

[9] B. Nazer and M. Gastpar, "Computation over multiple-access channels," *IEEE Transactions on Information Theory*, vol. 53, pp. 3498–3516, October 2007.

[10] S. Bhadra, P. Gupta, and S. Shakkottai, "On network coding for interference networks," in *IEEE ISIT 2006, Seattle, WA*.

[11] S. Zhang, S. Liew, and P. Lam, "Physical layer network coding," in *ACM Mobicom*, 2006.

[12] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," tech. rep., MIT, 2007.

[13] J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources," *IEEE Trans. Inform. Theory*, vol. 25, pp. 219–221, March 1979.

[14] T. Philosof, A. Khisti, U. Erez, and R. Zamir, "Lattice strategies for the dirty multiple access channel," in *IEEE ISIT 2007, Nice, France*.

[15] K. Narayanan, M. P. Wilson, and A. Sprintson, "Joint physical layer coding and network coding for bi-directional relaying," in *45th Annual Allerton Conference, Monticello, IL, September, 2007.*

[16] D. Krithivasan and S. S. Pradhan, "Lattices for distributed source coding: Jointly gaussian sources and reconstruction of a linear function," arxiv.cs.IT:0707.3461v1.

[17] T. Ho, D. R. Karger, M. Médard, and R. Koetter, "Network coding from a network flow perspective," in *Proceedings of the IEEE International Symposium on Information Theory, Yokohama, Japan*, 2003.

[18] L. Song, R. W. Yeung, and N. Cai, "A separation theorem for single-source network coding," *IEEE Transactions on Information Theory*, vol. 52, pp. 1861–1871, May 2006.

[19] R. G. Gallager, "A simple derivation of the coding theorem and some applications," *IEEE Transactions on Information Theory*, vol. 11, pp. 3–18, January 1965.

[20] R. Urbanke and B. Rimoldi, "Lattice codes can achieve capacity on the AWGN channel," *IEEE Trans. Inform. Theory*, vol. 44, pp. 273–278, January 1998.

[21] U. Erez and R. Zamir, "Achieving $\frac{1}{2}\log(1+\mathrm{SNR})$ on the AWGN channel with lattice encoding and decoding," *IEEE Trans. Inform. Theory*, vol. 50, pp. 2293–2314, October 2004.

[22] U. Erez, S. Litsyn, and R. Zamir, "Lattices which are good for (almost) everything," *IEEE Trans. Inform. Theory*, vol. 51, pp. 3401–3416, October 2005.

[23] B. Nazer and M. Gastpar, "Computation over Gaussian multiple-access channels," in *IEEE ISIT 2007, Nice, France*.

[24] Y. Kochman and R. Zamir, "Analog matching of colored sources to colored channels," in *IEEE ISIT 2006, Seattle, WA*.