Low Complexity Schemes for the Random Access Gaussian Channel

Or Ordentlich and Yury Polyanskiy

***Abstract*—We consider an uncoordinated Gaussian multiple access channel with a relatively large number of active users within each block. A low complexity coding scheme is proposed, which is based on a combination of compute-and-forward and coding for a binary adder channel. For a wide regime of parameters of practical interest, the energy-per-bit required by each user in the proposed scheme is significantly smaller than that required by popular solutions such as slotted-ALOHA and treating interference as noise.**

I. Introduction

One of the key challenges in the design of next generation's wireless networks is to allow for a large number of bursty users, each with a small amount of data, to transmit simultaneously in a grantless fashion. This need, which was already identified by Gallager three decades ago [1], is now returned to the research forefront due to explosion of the number of wireless devices [2].

To model this scenario, we consider a Gaussian multiple access channel where communication is performed in blocks of n channel uses. There are K_{tot} possible users that can transmit over the channel, but only K_a of them are active within each block, such that the receiver observes

$$\mathbf{y} = \sum_{i=1}^{K_{\text{tot}}} s_i \mathbf{x}_i + \mathbf{z}, \tag{1}$$

where $(s_1, \ldots, s_{K_{\text{tot}}}) \in \{0,1\}^{K_{\text{tot}}}$ is the "activity pattern" vector whose Hamming weight is K_a, $\mathbf{x}_i \in \mathbb{R}^n$ is the codeword transmitted by user i assuming it was active, and $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is additive white Gaussian noise (AWGN). We further assume that all users have the same message set $[M] \triangleq \{1, \ldots, M\}$, such that if user i is active, its message W_i is uniformly distributed over $[M]$, and that all users are subject to the same power constraint $\|\mathbf{x}_i\|^2 \leq nP$, $i = 1, \ldots, K_{\text{tot}}$. The activity pattern is assumed *unknown* to the decoder, and known only locally to the transmitters, i.e., each user only knows whether or not it is active, but does not know which of the other users are active.

The typical regime of interest is where the total number of devices connected to the network K_{tot} is orders of magnitude greater than the coding blocklength n, the number of active users K_a within each block scales as $K_a = \mu n$, with some $\mu \ll 1$, and the length in bits, $k \triangleq \log_2 M$, of each active user's message does not scale with n. Thus, although each user only has a small number of bits to send, the total number of bits per channel use that needs to be decoded, $\rho \triangleq \frac{k \cdot K_a}{n}$, is fixed. We refer to ρ as the required *spectral efficiency*. For example, in LP-WANs such as LoRaWAN and Weightless, $K_{\text{tot}} \approx 10^7$, $n \approx 10^4$, $K_a \approx 100$, and $k \approx 100$, such that ρ typically takes a moderate value between a fraction of a bit to a few bits per channel use.

Our formulation diverges from traditional multiple access literature [3], as well as that of [4], in our definition of successful decoding. We only require the decoder to output a list $\mathcal{L}(\mathbf{y}) = (w_1, \ldots, w_J) \subset [M]$ of no more than K_a messages (i.e., $J \leq K_a$) that should contain most messages that were transmitted by the active users, where the order in which the messages appear in the list is of no significance. Our model therefore decouples the user identification problem ("who was active") and the data transmission problem ("what messages were sent"), and is more consistent with the network theoretic studies. There, it is common to think of MAC layers job as that of delivering packets and not identifying who sent them. The reasoning is that part of the payload (headers) contains identifying information. The scheme's error probability is therefore defined as

$$P_e = \max_{|(s_1, \ldots, s_{K_{\text{tot}}})| = K_a} \frac{1}{K_a} \sum_{i=1}^{K_{\text{tot}}} s_i \cdot \Pr\left(W_i \notin \mathcal{L}(\mathbf{y})\right), \tag{2}$$

where $|\cdot|$ denotes Hamming weight. An advantage of this formulation is that it allows to set $K_{\text{tot}} = \infty$, which consequently leads to leaving the parameter K_{tot} out of our model, as we do in the sequel. See [2] for further justification of the model. Note that the assumption of $K_{\text{tot}} = \infty$ naturally leads to schemes where all users transmit from the same codebook, possibly with some additional randomization in the encoding procedure.

Let ϵ be the target error probability, measured according to (2). For fixed n, k, K_a, ϵ, our goal is to design a scheme with $P_e \leq \epsilon$ which requires the smallest possible transmission power P. In particular, we measure performance in terms of the energy per bit $\frac{E_b}{N_0} \triangleq \frac{nP}{2k}$ required for each user, where P is the minimal power such that $P_e \leq \epsilon$.

The grantless nature of the communication precludes the use of orthogonalization methods (TDMA,FDMA, orthogonal

OO and YP are with the Department of Electrical Engineering and Computer Science, MIT. e-mail: {ordent,yp}@mit.edu. OO is also with the Department of Electrical and Computer Engineering, Boston University.

The research was supported in part by the NSF grant CCF-13-18620 and then NSF CAREER award CCF-12-53205.

CDMA), and alternative efficient coding schemes are needed for this *random access* channel. Two popular solutions are "treat interference as noise" (TIN), which is implemented in practice via un-coordinated CDMA with a matched filter detector (i.e., no multi-user detection), and slotted-ALOHA. Unfortunately, both schemes have severe limitations in our regime of interest, as can be seen from Figure 1.[1]See [6] for further details.

While the performance of TIN is limited by noise accumulation, the large E_b/N_0 required by slotted-ALOHA is due to the fact that the scheme only supports single-user decoding. On the other extreme, if we had a computationally unlimited decoder, we could let all active users transmit simultaneously from the same (randomly constructed) codebook and perform joint decoding. A finite blocklength achievability bound for this setup was derived in [2, Theorem 1], and corresponds to the "random coding" curve in Figure 1.

As a compromise between these two extremes, we propose an approach referred to as T*-fold ALOHA*. This approach is similar to standard slotted-ALOHA in the sense that the block is split to sub-blocks and each active user only transmits in one random sub-block. However, in T-fold ALOHA, the code is designed such that if at most T users transmitted during the same sub-block, the decoder can decode all corresponding messages, whereas when more than T users simultaneously transmitted within the same sub-block, nothing is decoded. Thus 1-fold ALOHA is just slotted-ALOHA, whereas K_a-fold ALOHA corresponds to the scheme described in the previous paragraph. A random coding achievability bound for the E_b/N_0 required by 5-fold ALOHA, with a joint decoder applied within each sub-block, is plotted in Figure 1. However, to make T-fold ALOHA a practical solution, low complexity schemes for the random access channel with T active users are needed. In this paper, we propose such a scheme, which works well for moderate values of T.

A high-level description of the proposed coding scheme is as follows. First, the n channel uses are split into V sub-blocks of length $\bar{n} = n/V$, and each active user randomly chooses only one of these sub-blocks, over which it transmits. All users encode their messages using the same codebook $\mathcal{C} \subset \mathbb{F}_2^{\bar{n}}$, which is then mapped to a BPSK constellation. The code $\mathcal{C}$ is a concatenation of two codes. The first is an inner binary linear code, whose goal is to enable the receiver to decode the modulo-2 sum of all codewords transmitted within the same sub-block. We refer to recovering this modulo-2 sum as the *compute-and-forward [7] (CoF) phase*. The second code, is an outer code whose goal is to enable the receiver to recover the individual messages that participated in the modulo-2 sum. We refer to recovering the individual messages from their modulo-2 sum as the *binary adder channel (BAC) phase*.

The success probability of the CoF phase in our scheme is independent of the actual number of users that transmitted within the same sub-block. The outer code, however, is designed such that if at most T active users approached the channel during the same sub-block, it is possible to determine the individual messages from their modulo-2 sum, essentially with zero error probability. Thus, loosely speaking, for any active user, the probability that its message is not in the list $\mathcal{L}(\mathbf{y})$ is dictated by the probability that the compute-and-forward phase was unsuccessful in the sub-block where it transmitted, and the probability that more than T users approached the channel within this sub-block.

The design of an inner code for the CoF phase, reduces to that of finding codes that perform well over a binary-input memoryless output-symmetric (BMS) channel, for which many off-the-shelf codes can be used. We construct the outer code for the BAC phase from the columns of a T-error correcting BCH codes, and show that this code can be decoded efficiently [8], even though the blocklength for the underlying BCH code is orders of magnitudes greater than the allowed number of operations that can be performed by a practical decoder.

Both components of our scheme are not new and there is a large body of literature on each of them separately. The observation that BCH-codes can be used for constructing zero-error codes with rate $1/T$ for the T-ary modulo-2 adder channel dates back to Lindström [9] and have since then appeared and was generalized in various works, see e.g., [8], [10]. A particularly related work is [10] where the authors used a similar concatenated code to construct a code with good minimum Hamming distance for the T-user modulo-2 adder channel. The use of linear codes for decoding modulo sums of codewords from the output of a Gaussian multiple access channel is more recent, and has its roots in the work of Körner and Marton [11]. However, the combination of the these two components, in conjunction with T-fold ALOHA, for providing a low complexity scheme with low energy per bit for the Gaussian random access channel is novel, and, as can be seen in Figure 1, leads to performance that cannot be attained by other known schemes of similar complexity, in some regimes of practical interest. The recent works [12], [13] propose coding schemes of similar flavor to ours, but those are less suitable for our regime of interest, where the number of possible users is unbounded, the message length of each user is small, and the target is to minimize the energy per bit.

II. The Basic Coding Scheme

Our scheme has two design parameters, T which is the maximal number of users that can simultaneously transmit in the same sub-block without incurring an error, and $\alpha \in [0, 1]$, such that the number of sub-blocks is $V = K_a/(\alpha T)$.

Code construction: We construct one codebook $\mathcal{C} \subset \mathbb{F}_2^{\bar{n}}$ with $|\mathcal{C}| = 2^k = 2^{\bar{n}R}$ codewords, to be used by all active transmitters, where $\bar{n} = \frac{n}{V} = \alpha T \frac{n}{K_a}$ and $R = \frac{\rho}{\alpha T}$.

The codebook $\mathcal{C}$ is a concatenated code. The "inner" code is a systematic binary linear code $\mathcal{C}_{\text{lin}} \subset \mathbb{F}_2^{\bar{n}}$ of rate R_{lin}, with

[1]Another appealing alternative is coded slotted ALOHA [5]. See [2] for an optimistic estimate of its performance in term of energy per bit.

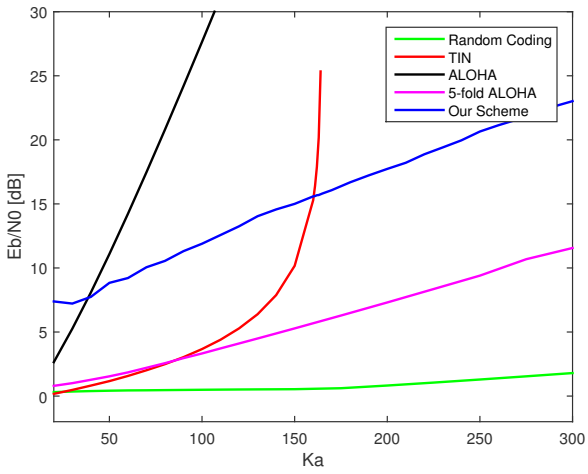

Fig. 1. Comparison between the E_b/N_0 required by various schemes for the setup $k = 100$ bits, $n = 30,000$ channel uses, number of active users K_a varies, and $\epsilon = 0.05$.

generating matrix $\mathbf{G} \in \mathbb{F}_2^{\bar{n}R_{\text{lin}} \times \bar{n}}$, such that

$$\mathcal{C}_{\text{lin}} = \left\{ \mathbf{aG} \ : \ \mathbf{a} \in \mathbb{F}_2^{1 \times \bar{n}R_{\text{lin}}} \right\}. \tag{3}$$

The "outer" code is a binary code (not necessarily linear) $\mathcal{C}_{\text{BAC}} \subset \mathbb{F}_2^{\bar{n}R_{\text{lin}}}$ with rate R_{BAC}. The concatenated binary code $\mathcal{C} \subset \mathbb{F}_2^{\bar{n}}$ with rate $R = R_{\text{lin}} \cdot R_{\text{BAC}} = \frac{k}{\bar{n}}$ is defined as

$$\mathcal{C} = \{ \mathbf{c}_{\text{BAC}}\mathbf{G} \ : \ \mathbf{c}_{\text{BAC}} \in \mathcal{C}_{\text{BAC}} \}. \tag{4}$$

The roles played by the inner code and outer code, as well as the criteria according to which they should be chosen, will be discussed in the sequel.

Encoding: Each active user i first encodes its message W_i to a codeword $\mathbf{c}_{\text{BAC},i} \in \mathcal{C}_{\text{BAC}}$, and then uses $\mathbf{G}$ to generate the codeword

$$\mathbf{c}_i = \mathbf{c}_{\text{BAC},i}\mathbf{G} \in \mathcal{C}. \tag{5}$$

Next, it maps the binary vector $\mathbf{c}_i$ to the real vector $\mathbf{x}_i = 2\sqrt{V \cdot P}\left(\mathbf{c}_i - \frac{1}{2}\right)$, where here and throughout the rest of the paper we interchangeably treat $\{0,1\}$ as either integers or elements of $\mathbb{F}_2$, according to the context. Note that $\|\mathbf{x}_i\|^2 = nP$.

User i transmits the vector $\mathbf{x}_i$ during one and only one of the V sub-blocks. The location of this sub-block is randomly drawn independently across users from the uniform distribution over $\{1, \ldots, V\}$. We denote by $E_{1,i}$ be the event that more than $T-1$ other active users transmitted within the same sub-block as user i.

Decoding: Decoding is done in a sub-block by sub-block manner. For each sub-block $v \in [V]$, the decoder outputs a list $\mathcal{L}_v$ of at most T messages. The list of messages for the entire block is then constructed as $\mathcal{L}(\mathbf{y}) = \cup_{v=1}^{V} \mathcal{L}_v$.

We describe the decoding procedure for the first sub-block. For the other $V-1$ sub-blocks decoding is done in an identical fashion. Let $\mathbf{y}_1 = (y_1, \ldots, y_{\bar{n}})$ and $\mathbf{z}_1 = (z_1, \ldots, z_{\bar{n}})$ be the vectors of channel outputs and channel noise, respectively, corresponding to the first sub-block, and let $i_1, \ldots, i_L$ be the active users that transmitted during this sub-block. We have

$$\mathbf{y}_1 = \sum_{j=1}^{L} \mathbf{x}_{i_j} + \mathbf{z}_1 = 2\sqrt{V \cdot P}\left(\sum_{j=1}^{L} \mathbf{c}_{i_j} + \frac{\mathbf{z}_1}{2\sqrt{V \cdot P}} - \frac{L}{2} \right).$$

We may assume the number of active users L within the sub-block is known to the receiver. This follows since we can first apply the decoding algorithm for all values $L = 1, \ldots, T$, and then choose the produced list that has the "best agreement" with $\mathbf{y}_1$, which should correspond to the actual value of L, or decide that $L > T$ if all the produced lists are not in "good agreement" with $\mathbf{y}_1$ [6]. If $L > T$, the receiver outputs $\mathcal{L}_1 = \emptyset$. Otherwise, it computes

$$\mathbf{y}_{\text{CoF},1} = \left[\frac{1}{2\sqrt{VP}}\mathbf{y}_1 + \frac{L}{2} \right] \bmod 2 = \left[\sum_{j=1}^{L} \mathbf{c}_{i_j} + \tilde{\mathbf{z}}_1 \right] \bmod 2,$$

where the modulo 2 reduction is into the interval $[0, 2)$ and is taken componentwise, and $\tilde{\mathbf{z}}_1 = \frac{\mathbf{z}_1}{2\sqrt{V \cdot P}} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, $\sigma^2 = 1/4VP$. Let $\mathbf{c}_1^{\oplus} \triangleq [\sum_{j=1}^{L} \mathbf{c}_{i_j}] \bmod 2$, and note that since $\mathbf{c}_1^{\oplus}$ is the modulo-2 sum of codewords from the same linear code $\mathcal{C}_{\text{lin}}$, we have that $\mathbf{c}_1^{\oplus} \in \mathcal{C}_{\text{lin}}$. Thus, we constructed an effective memoryless channel

$$\mathbf{y}_{\text{CoF},1} = \left[\mathbf{c}_1^{\oplus} + \tilde{\mathbf{z}}_1 \right] \bmod 2, \tag{6}$$

whose input is a codeword from the linear code $\mathcal{C}_{\text{lin}}$. The decoder ignores the fact that $\mathbf{c}_1^{\oplus}$ is not distributed uniformly on $\mathcal{C}_{\text{lin}}$, and simply performs point-to-point decoding of $\mathcal{C}_{\text{lin}}$ from $\mathbf{y}_{\text{CoF},1}$ to produce the estimate $\hat{\mathbf{c}}_1^{\oplus}$. We denote the erroneous decoding event by $E_2 \triangleq \{\hat{\mathbf{c}}_1^{\oplus} \neq \mathbf{c}_1^{\oplus}\}$.

Now, assuming E_2 did not occur, the decoder proceeds to recover the L messages transmitted by the active users from $\mathbf{c}_1^{\oplus}$. By (5) and the fact that $\mathcal{C}_{\text{lin}}$ is systematic, we have that the first $\bar{n}R_{\text{lin}}$ coordinates of $\mathbf{c}_1^{\oplus}$ correspond to

$$\mathbf{y}_{\text{BAC},1} = \sum_{j=1}^{L} \mathbf{c}_{\text{BAC},i_j} \bmod 2. \tag{7}$$

Thus, the decoder uses $\mathbf{y}_{\text{BAC},1}$ to produce a list of L vectors $\tilde{\mathcal{L}}(\mathbf{y}_{\text{BAC},1}) = \{\hat{\mathbf{c}}_{\text{BAC},1}, \ldots, \hat{\mathbf{c}}_{\text{BAC},L}\} \in \mathcal{C}_{\text{BAC}}^L$ that satisfy (7). We denote the corresponding error event by

$$E_3 \triangleq \left\{ \tilde{\mathcal{L}}(\mathbf{y}_{\text{BAC},1}) \neq \{\mathbf{c}_{\text{BAC},i_1}, \ldots, \mathbf{c}_{\text{BAC},i_L}\} \right\}, \tag{8}$$

where both $\tilde{\mathcal{L}}(\mathbf{y}_{\text{BAC},1})$ and $\{\mathbf{c}_{\text{BAC},i_1}, \ldots, \mathbf{c}_{\text{BAC},i_L}\}$ are sets and therefore there is no significance to the order in which their elements appear.

Finally, the decoder re-maps the codewords in $\tilde{\mathcal{L}}(\mathbf{y}_{\text{BAC},1})$ to a list of the corresponding messages $\mathcal{L}_1 \subset [M]$.

Error probability: Assume user i was one of the K_a active users, and without loss of generality, assume further that it transmitted during the first sub-block. Since the role of all active users in the proposed scheme is symmetric, we have

that $P_e = \Pr(W_i \notin \mathcal{L}(\mathbf{y})) \leq \Pr(W_i \notin \mathcal{L}_1)$. Thus,

$$P_e \leq \Pr(E_{1,i}) + \Pr(E_2|\bar{E}_{1,i}) + \Pr(E_3|\bar{E}_{1,i}, \bar{E}_2).$$

For the event $E_{1,i}$, using $V = K_a/\alpha T$ we have that

$$\Pr(E_{1,i}) \quad = 1 - \Pr\left(\text{Binomial}\left(K_a - 1, \frac{\alpha T}{K_a}\right) < T\right) \triangleq \epsilon_1 \tag{9}$$

regardless of the codes $\mathcal{C}_{\text{lin}}, \mathcal{C}_{\text{BAC}}$ that are used. The error probability $\Pr(E_2|\bar{E}_{1,i})$ depends on the choice of $\mathcal{C}_{\text{lin}}$, whereas $\Pr(E_3|\bar{E}_{1,i}, \bar{E}_2)$ depends on the choice of $\mathcal{C}_{\text{BAC}}$. We will therefore treat them in the next subsection.

A. Choice of inner and outer codes

Code for CoF phase: The code $\mathcal{C}_{\text{lin}}$ should allow decoding of $\mathbf{c}_1^{\oplus}$ from the channel (6), with error probability smaller than some target ϵ_2. The channel (6) is a BMS channel, for which the art of designing efficient coding schemes is well advanced. Thus, any off-the-shelf low complexity code with good performance over a BMS (e.g., LDPC, turbo, polar) can be used for $\mathcal{C}_{\text{lin}}$. For the numerical analysis of Figure 1, we refrain from committing to a particular code, and use the fundamental coding limits of the channel (6) for evaluating $\epsilon_2 = \Pr(E_2|\bar{E}_{1,i})$. Specifically, in order to determine the smallest P that allows correct decoding of $\mathbf{c}_1^{\oplus}$ with error probability below ϵ_2, we use the normal approximation [14]

$$R_{\text{lin}} \approx C(P) - \sqrt{\frac{V(P)}{\bar{n}}} Q^{-1}(\epsilon_2) \tag{10}$$

and solve for P. To evaluate the quantities $C(P)$ and $V(P)$ we define the random variable $\tilde{Z} = [\bar{Z}] \bmod 2$ with pdf $f_{\tilde{Z}}$, where $\bar{Z} \sim \mathcal{N}(0, 1/4VP)$, and set

$$i(\tilde{Z}) = \log_2\left(\frac{f_{\tilde{Z}}(\tilde{Z})}{\frac{1}{2}f_{\tilde{Z}}(\tilde{Z}) + \frac{1}{2}f_{\tilde{Z}}([\tilde{Z}-1] \bmod 2)}\right), \tag{11}$$

$$C(P) = \mathbb{E}\ i(\tilde{Z}),\ V(P) = \text{Var}\ i(\tilde{Z}). \tag{12}$$

Code for BAC phase: The code $\mathcal{C}_{\text{BAC}}$ for the BAC phase should enable recovering $(\mathbf{c}_{\text{BAC},i_1}, \ldots, \mathbf{c}_{\text{BAC},i_L})$ from $\mathbf{y}_{\text{BAC},1}$ as long as $L \leq T$. Thus the coding task is equivalent to that of coding for the T-user modulo-2 binary adder channel where all users' codewords are taken from the same codebook $\mathcal{C}_{\text{BAC}}$. An obvious upper bound on the rate of such a code, if a small error probability is desired, is $R_{\text{BAC}} \leq 1/T$. Remarkably, this bound can be achieved using the columns of a binary BCH code parity check matrix as the codewords of $\mathcal{C}_{\text{BAC}}$ [9]. To see this, first recall that if a linear code has minimum distance $2T+1$, then all modulo-2 sums of T or less columns of its parity check matrix are distinct. It is well known [15] that for any $k \geq 3$ and $T < 2^{k-1}$ there exists a binary BCH code with parameters $(n = 2^k - 1, n - k \leq kT, d_{\min} \geq 2T+1)$. Thus, taking the columns of a BCH parity check matrix results in a code $\mathcal{C}_{\text{BAC}} \subset \mathbb{F}_2^{kT}$ of size $|\mathcal{C}_{\text{BAC}}| = 2^k - 1$ with the property that the modulo-2 sum of any set of at most T *distinct* codewords is distinct. Thus, a codebook $\mathcal{C}_{\text{BAC}}$ constructed this way has rate $R_{\text{BAC}} = \log_2(2^k - 1)/kT \approx 1/T$. The error probability associated with this code is

$$\epsilon_3 \triangleq \Pr(E_3|\bar{E}_{1,i}, \bar{E}_2) = \Pr\left(\bigcup_{i \neq j}\{W_i = W_j\}\right) \leq \frac{\binom{T}{2}}{2^k - 1}$$

as errors can only occur if some of the L users that approached the channel during the first sub-block had the same message.

The encoding procedure for this codebook merely consists of mapping a message to a corresponding element α in $GF(2^k)$, and then computing its odd powers $(\alpha, \alpha^3, \ldots, \alpha^{2T-1})$, which requires $O(T^2)$ multiplications in $GF(2^k)$. The decoding procedure shares many similarities with standard Gorenstein-Peterson-Zierler (GPZ) decoding of BCH codes [15], but is far less demanding computationally. In particular, the standard BCH decoding algorithm has complexity linear in the blocklength. Since for our underlying BCH code the blocklength is $2^k - 1$, such a computational cost is prohibitive even for relatively small k, say $k \approx 100$. Luckily, the most demanding operations in the GPZ algorithm are not needed for our purposes and the computational cost becomes $\mathcal{O}(kT^2)$ additions and multiplications in $GF(2^k)$. The exact encoding and decoding algorithms we use, which are quite similar to [8], are described in [6]

III. Extension to Multilevel Codes

The CoF phase in the scheme proposed in Section II reduces the L-user Gaussian MAC channel into an L-user binary input modulo-2 Gaussian MAC. As such, the rate of the linear code is limited by $R_{\text{lin}} < 1$ total bits per channel use. As $R_{\text{lin}} = \rho/\alpha$, this restricts both the total spectral efficiency of the scheme, and the regime of valid choices for α (which is related to ϵ_1 by (9)). In order to circumvent this problem, while keeping the many practical advantages of binary codes, we propose to modify the basic scheme from Section II using a multilevel code design. We only describe below a scheme that uses two layers, and can therefore attain $0 < R_{\text{lin}} < 2$, but the extension to an arbitrary number of layers is straightforward.

We construct two codebooks $\mathcal{C}^a, \mathcal{C}^b \in \mathbb{F}_2^{\bar{n}}$ with rates R^a, R^b, respectively, each according to the same code construction described in Section II. Thus, $\mathcal{C}^a$ ($\mathcal{C}^b$) is a concatenation of an inner code $\mathcal{C}^a_{\text{lin}}$ ($\mathcal{C}^b_{\text{lin}}$) and $\mathcal{C}^a_{\text{BAC}}$ ($\mathcal{C}^b_{\text{BAC}}$), with rates R^a_{lin} and R^a_{BAC} (R^b_{lin} and R^b_{BAC}), respectively.

Let $0 < m < \bar{n} \cdot \min\{R^a, R^b\}$ be an integer. Each active user i has a message vector $\mathbf{w}_i = (\mathbf{w}_i^a, \mathbf{w}_i^b) \in \mathbb{F}_2^{\bar{n}R^a - m} \setminus \{\mathbf{0}\} \times \mathbb{F}_2^{\bar{n}R^b - m} \setminus \{\mathbf{0}\}$. Then, user i draws an m-dimensional binary vector $\mathbf{u}_i$ with i.i.d. uniform entries, and creates the effective message vectors $\tilde{\mathbf{w}}_i^a = (\mathbf{u}_i, \mathbf{w}_i^a) \in \mathbb{F}_2^{\bar{n}R^a} \setminus \{\mathbf{0}\}$ and $\tilde{\mathbf{w}}_i^b = (\bar{\mathbf{u}}_i, \mathbf{w}_i^b) \in \mathbb{F}_2^{\bar{n}R^b} \setminus \{\mathbf{0}\}$, where $\bar{\mathbf{u}}_i$ is the complement of $\mathbf{u}_i$ such that $\mathbf{u}_i + \bar{\mathbf{u}}_i = \mathbf{1} \bmod 2$. Now, $\tilde{\mathbf{w}}_i^a$ ($\tilde{\mathbf{w}}_i^b$) is encoded to a codeword $\mathbf{c}_i^a$ ($\mathbf{c}_i^b$) in $\mathcal{C}^a$ ($\mathcal{C}^b$) exactly as described in Section II, and the transmitted vector is

$$\mathbf{x}_i = \sqrt{\frac{V \cdot P}{5}}\left(2\left(\mathbf{c}_i^a - \frac{1}{2}\right) + 4\left(\mathbf{c}_i^b - \frac{1}{2}\right)\right),$$

and as long as either $\mathcal{C}^a$ or $\mathcal{C}^b$ (or both) are such that for

a random codeword $\mathbf{c}^a$ ($\mathbf{c}^b$) uniformly distributed over $\mathcal{C}^a$ ($\mathcal{C}^b$) we have $\mathbb{E}(\mathbf{c}^a - \frac{1}{2}) = \mathbf{0}$ ($\mathbb{E}(\mathbf{c}^b - \frac{1}{2}) = \mathbf{0}$), we have that $\mathbb{E}\|\mathbf{x}_i\|^2 \leq nP$. Note that here we can only guarantee that the power constraint is maintained on average, and not with probability 1 as in the single layer construction. Each active user then chooses one sub-block in which it transmits its codeword exactly as in the basic scheme from Section II.

The decoding is performed layer by layer in each sub-block. As before, we only describe the decoding process in the first sub-block. We first compute

$$\mathbf{y}_{\text{CoF},1} = \frac{1}{2}\sqrt{\frac{5}{V \cdot P}}\left(\mathbf{y}_1 + \frac{3L}{2}\right) = \sum_{j=1}^{L} \mathbf{c}_{i_j}^a + 2\sum_{j=1}^{L} \mathbf{c}_{i_j}^b + \tilde{\mathbf{z}}_1^a,$$

where $\tilde{\mathbf{z}}_1^a = \frac{\sqrt{5}\mathbf{z}_1}{\sqrt{4V \cdot P}} \sim \mathcal{N}(\mathbf{0}, \sigma_a^2 \mathbf{I})$, $\sigma_a^2 = \frac{5}{4VP}$. Now, setting $\mathbf{y}_{\text{CoF},1}^a = [\mathbf{y}_{\text{CoF},1}] \bmod 2$, and continuing exactly as in the basic scheme from Section II, we can recover $\{\tilde{\mathbf{w}}_{i_1}^a, \ldots, \tilde{\mathbf{w}}_{i_L}^a\}$. This allows us to form $\sum_{j=1}^{L} \mathbf{c}_{i_j}^a$, and then construct

$$\begin{aligned}\mathbf{y}_{\text{CoF},1}^b &= \left[\frac{1}{2}\left(\mathbf{y}_{\text{CoF},1} - \sum_{j=1}^{L} \mathbf{c}_{i_j}^a\right)\right] \bmod 2 \\ &= \left[\sum_{j=1}^{L} \mathbf{c}_{i_j}^b + \tilde{\mathbf{z}}_1^b\right] \bmod 2,\end{aligned}$$

where $\tilde{\mathbf{z}}_1^b \sim \mathcal{N}(\mathbf{0}, \sigma_b^2 \mathbf{I})$, $\sigma_b^2 = \frac{5}{16VP}$. We can now recover $\{\tilde{\mathbf{w}}_{i_1}^b, \ldots, \tilde{\mathbf{w}}_{i_L}^b\}$, exactly as in the basic scheme from Section II. The effective channel $\tilde{\mathbf{z}}_1^b$ is "cleaner" than $\tilde{\mathbf{z}}_1^a$, therefore we will choose $\mathcal{C}_{\text{lin}}^a, \mathcal{C}_{\text{lin}}^b$ such that $R_{\text{lin}}^a \leq R_{\text{lin}}^b$, where their exact values should be optimized w.r.t. the target error probability and to $V \cdot P$. The codes $\mathcal{C}_{\text{BAC}}^a, \mathcal{C}_{\text{BAC}}^b$ for the BAC phase are both BCH-based codes of rate $R_{\text{BAC}}^a = R_{\text{BAC}}^b = 1/T$, as described in Section II-A, where they only differ in their blocklengths $\bar{n}R_{\text{lin}}^a$ and $\bar{n}R_{\text{lin}}^b$, respectively.

The final step is to use the two lists $\{\tilde{\mathbf{w}}_{i_1}^a, \ldots, \tilde{\mathbf{w}}_{i_L}^a\}$ and $\{\tilde{\mathbf{w}}_{i_1}^b, \ldots, \tilde{\mathbf{w}}_{i_L}^b\}$ in order to construct a single list $\{\mathbf{w}_{i_1}, \ldots, \mathbf{w}_{i_L}\}$. This is done by first constructing L pairs, that should ideally be of the form $\tilde{\mathbf{w}}_{i_j} = (\tilde{\mathbf{w}}_{i_j}^a, \tilde{\mathbf{w}}_{i_j}^b)$, and then removing the prefixes $\mathbf{u}_{i_j}, \bar{\mathbf{u}}_{i_j}$ to get the messages $\mathbf{w}_{i_j}$. The problem in doing this is that the messages in each of the two lists are decoded "un-indexed". Thus, the pairing operation is done by matching the random prefixes $\{\mathbf{u}_{i_1}, \ldots, \mathbf{u}_{i_L}\}$ from the first list to the prefixes $\{\bar{\mathbf{u}}_{i_1}, \ldots, \bar{\mathbf{u}}_{i_L}\}$ of the second list. As long as the L prefixes $\{\mathbf{u}_{i_1}, \ldots, \mathbf{u}_{i_L}\}$ drawn by the users are distinct, the pairing is successful. Thus, the error probability associated with this step is

$$\epsilon_4 = 1 - \prod_{\ell=1}^{L-1}(1 - \ell 2^{-m}) \leq T(T-1) \cdot 2^{-(m+1)}. \qquad (13)$$

Once the target ϵ_4 is chosen, it therefore suffices to take $m = \lceil \log_2(T(T-1)/\epsilon_4) \rceil - 1$, where the clear disadvantage of increasing m is that it requires the linear codes to operate with higher rates in order to deliver the k information bits.

IV. Numerical Evaluation

Fix k, n, K_a, and target error probability P_e. As the error probability for the BAC phase decays exponentially with k, it can be ignored. Thus, P_e is essentially dictated by the "T-collision" probability ϵ_1, the error probability for the CoF phase ϵ_2, and when a multilevel code is used, also the "pairing" error probability ϵ_4.

We fix target probabilities $\epsilon_1, \epsilon_2, \epsilon_4$ such that $\epsilon_1 + \epsilon_2 + \epsilon_4 = \epsilon$, and assume temporarily that T is also fixed. We set α to be the solution of the equation (9) in α. Assume we are using a multilevel code with $\tau \geq 1$ layers. The "pairing" procedure increases the effective length of each user's message from k to $k(1+\tau\gamma)$, where $\gamma \triangleq m/k$, and $m = \lceil \log_2(T(T-1)/\epsilon_4) \rceil - 1$. Recalling that $R_{\text{BAC}} = 1/T$ for the BCH-based construction, and that $R = R_{\text{BAC}} \cdot R_{\text{lin}}$, we see that the rate the of linear code must satisfy $R_{\text{lin}} = Tk/\bar{n} = \rho/\alpha$ in the single level case, and in the multilevel case the sum of linear codes rates must be $\rho(1+\tau\gamma)/\alpha$, whereas the blocklength for this code (or codes) is $\bar{n} = \alpha T n / K_a$. The required average power $P \cdot V$ in order to achieve error probability ϵ_2 with this rate and blocklength, over the channel (6) can be computed using (10), and the resulting E_b/N_0 after optimization on T, τ and the choices of $\epsilon_1, \epsilon_2, \epsilon_4$ that sum up to ϵ, is shown in Figure 1.

Acknowledgment

The authors are grateful to Uri Erez, Krishna Narayanan and Bobak Nazer for valuable discussions.

References

[1] R. Gallager, "A perspective on multiaccess channels," *IEEE Trans. Inf. Theory*, vol. 31, no. 2, pp. 124–142, Mar 1985.

[2] Y. Polyanskiy, "A perspective on massive random-access," in *Proc. of ISIT*, 2017.

[3] A. El Gamal and Y.-H. Kim, *Network information theory*. Cambridge University Press, 2011.

[4] X. Chen, T. Chen, and D. Guo, "Capacity of Gaussian many-access channels," 2016. [Online]. Available: http://arxiv.org/abs/1607.01048

[5] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted aloha," *IEEE Trans. Comm.*, vol. 59, no. 2, pp. 477–487, Feb. 2011.

[6] O. Ordentlich and Y. Polyanskiy, "Low complexity schemes for the random access Gaussian channel," preprint. [Online]. Available: http://www.mit.edu/∼ordent/publications/RandomAccessFull.pdf

[7] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Trans. Inf. Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.

[8] I. Bar-David, E. Plotnik, and R. Rom, "Forward collision resolution-a technique for random multiple-access to the adder channel," *IEEE Trans. Inf. Theory*, vol. 39, pp. 1671–1675, Sep 1993.

[9] B. Lindström, "Determination of two vectors from the sum," *Journal of Combinatorial Theory*, vol. 6, no. 4, pp. 402 – 407, 1969.

[10] T. Ericson and V. I. Levenshtein, "Superimposed codes in the hamming space," *IEEE Trans. Inf. Theory*, vol. 40, pp. 1882–1893, Nov 1994.

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

[12] J. Goseling, C. Stefanovic, and P. Popovski, "Sign-compute-resolve for random access," in *Proc. of Allerton*, 2014, pp. 675–682.

[13] N. Lee and S. N. Hong, "Coded compressive sensing: A compute-and-recover approach," in *Proc. of ISIT*, 2016, pp. 2359–2363.

[14] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.

[15] S. Lin and D. J. Costello, *Error control coding*. Pearson Education, 2004.