Low Complexity Schemes for the Random Access Gaussian Channel

Or Ordentlich
MIT
ordent@mit.edu

Yury Polyanskiy
MIT
yp@mit.edu

***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,

$$\rho \triangleq \frac{k \cdot K_a}{n}, \tag{2}$$

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]–[5], as well as that of [6], in our definition of successful decoding. We only require the decoder to output a list $\mathcal{L}(\mathbf{y}) = (w_1, \ldots, w_J) \subset [M]^J$ 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. In other words, the decoder is only required to declare which messages were transmitted, without associating the messages to the users that transmitted them. 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

$$P_e = \max_{|(s_1,\ldots,s_{K_{\text{tot}}})|=K_a} \frac{1}{K_a} \sum_{i=1}^{K_{\text{tot}}} s_i \cdot \Pr\left(W_i \notin \mathcal{L}(\mathbf{y})\right), \tag{3}$$

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

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

procedure.

Let ϵ be the target error probability, measured according to (3). 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

$$\frac{E_b}{N_0} \triangleq \frac{nP}{2k}, \tag{4}$$

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 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, as we explain below, both schemes have severe limitations in our regime of interest.[1]

For TIN, the highest rate that can be achieved by an active user is $\frac{1}{2}\log_2(1+\frac{P}{1+(K_a-1)P})$. The sum-rate is therefore upper bounded by $\frac{\log_2(e)\cdot K_a}{2(K_a-1)}$, and consequently, when K_a is such that $\rho > \log_2(e)/2$, this scheme cannot succeed with any E_b/N_0. When finite blocklength effects are taken into account, the highest ρ that can be achieved by TIN becomes even smaller under the assumption that $K_a = \mu n$ [2].

In slotted-ALOHA, the block is split to $V = K_a/\alpha$ sub-blocks, where $0 < \alpha \leq 1$, each of size n/V, and each active user only transmits in one of these sub-blocks, selected uniformly at random, independently across users. Whenever only one user transmitted in a sub-block, its message is decoded, but when collisions occur, all colliding codewords are not decoded. Thus, P_e is mainly dictated by the collision probability, which is approximately $1-e^{-\alpha}$ for large K_a. The disadvantage of this scheme is that the effective blocklength for each active user is decreased by a factor of α w.r.t. TDMA, such that the effective spectral efficiency is increased to ρ/α. The required E_b/N_0 for this scheme is therefore at least $\frac{2^{2\rho/\alpha}-1}{2\rho/\alpha} \approx -\frac{2^{-2\rho/\ln(1-\epsilon)}-1}{2\rho/\ln(1-\epsilon)}$, which becomes very large when the required error probability ϵ is small. The performance of TIN and slotted-ALOHA for our parameters of interest is depicted in Figure 1.

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 the 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 [9] (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 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

[1]Another appealing alternative is coded slotted ALOHA [7], [8]. However, the non-asymptotic performance of this scheme is currently not fully understood, which precludes including it in our comparisons. In particular, in our regime of interest the blocklength (per user) is short, which leads to non-negligible losses due to pointers to locations of repetitions. Moreover, the number of active users per block varies from tens to hundreds, and therefore, asymptotic analysis of the successive cancellation decoding scheme is not valid.

decoded efficiently [10], 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 [11] and have since then appeared and was generalized in various works, see e.g., [10], [12]. A particularly related work is [12] 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 [9], [13]–[16], and has its roots in the work of Körner and Marton [17]. However, the combination of the these two components for providing a low complexity scheme 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 schemes of similar complexity in some regimes of practical interest.

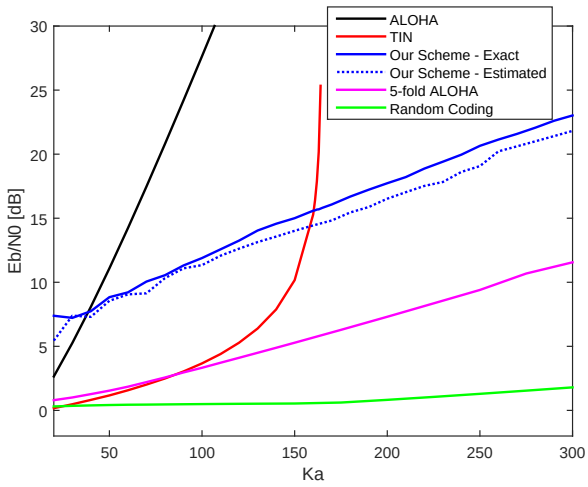

Fig. 1. Comparison between the E_b/N_0 required by various schemes for the setup $k = 100$ bits, $n = 30,000$ channel uses, number of active users K_a varies, and $\epsilon = 0.05$.

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

$$\mathcal{C}_{\text{lin}} = \left\{ \mathbf{aG} \ : \ \mathbf{a} \in \mathbb{F}_2^{1 \times \bar{n}R_{\text{lin}}} \right\}. \tag{5}$$

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

$$\mathcal{C} = \{ \mathbf{c}_{\text{BAC}} \mathbf{G} \ : \ \mathbf{c}_{\text{BAC}} \in \mathcal{C}_{\text{BAC}} \}. \tag{6}$$

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

$$\mathbf{c}_i = \mathbf{c}_{\text{BAC},i} \mathbf{G} \in \mathcal{C}. \tag{7}$$

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 on a sub-block by sub-block basis. 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 manner. 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

$$\begin{aligned} \mathbf{y}_1 &= \sum_{j=1}^{L} \mathbf{x}_{i_j} + \mathbf{z}_1 \\ &= 2\sqrt{V \cdot P} \left(\sum_{j=1}^{L} \mathbf{c}_{i_j} + \frac{\mathbf{z}_1}{2\sqrt{V \cdot P}} - \frac{L}{2} \right). \end{aligned} \tag{8}$$

We assume the number of active users L within the sub-block is known to the receiver, and justify this assumption in Section II-B. If $L > T$, the receiver outputs $\mathcal{L}_1 = \emptyset$. Otherwise, it computes

$$\begin{aligned} \mathbf{y}_{\text{CoF},1} &= \left[\frac{1}{2\sqrt{V \cdot P}} \mathbf{y}_1 + \frac{L}{2} \right] \bmod 2 \\ &= \left[\sum_{j=1}^{L} \mathbf{c}_{i_j} + \tilde{\mathbf{z}}_1 \right] \bmod 2, \end{aligned} \tag{9}$$

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

$$\mathbf{y}_{\text{CoF},1} = \left[\mathbf{c}_1^{\oplus} + \tilde{\mathbf{z}}_1\right] \bmod 2, \tag{10}$$

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 (7) 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

$$\mathbf{y}_{\text{BAC},1} = \sum_{j=1}^{L} \mathbf{c}_{\text{BAC},i_j} \bmod 2. \tag{11}$$

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 (11). We denote the corresponding error event by

$$E_3 \triangleq \left\{\tilde{\mathcal{L}}(\mathbf{y}_{\text{BAC},1}) \neq \{\mathbf{c}_{\text{BAC},i_1}, \ldots, \mathbf{c}_{\text{BAC},i_L}\}\right\}, \tag{12}$$

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 $\mathcal{L}_1 \in [M]^L$.

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,

$$P_e \leq \Pr(E_{1,i}) + \Pr(E_2|\bar{E}_{1,i}) + \Pr(E_3|\bar{E}_{1,i}, \bar{E}_2) \tag{13}$$

For the event $E_{1,i}$ we have that

$$\begin{aligned}\Pr(E_{1,i}) &= \Pr\left(\text{Binomial}\left(K_a - 1, \frac{1}{V}\right) \geq T\right) \\ &= 1 - \Pr\left(\text{Binomial}\left(K_a - 1, \frac{\alpha T}{K_a}\right) < T\right) \triangleq \epsilon_1\end{aligned} \tag{14}$$

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: We begin with discussing the design of $\mathcal{C}_{\text{lin}}$. This code should allow decoding of $\mathbf{c}_1^{\oplus}$ from the channel (10), with error probability smaller than some target ϵ_2. The channel (10) is a binary-input memoryless output-symmetric (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 that follows, we refrain from committing to a particular code, and use the fundamental coding limits of the channel (10) 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 [18]

$$R_{\text{lin}} \approx C(P) - \sqrt{\frac{V(P)}{\bar{n}}} Q^{-1}(\epsilon_2) \tag{15}$$

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 density $P_{\tilde{Z}}$, where $\bar{Z} \sim \mathcal{N}(0, 1/4VP)$, and set

$$i(\tilde{Z}) = \log_2\left(\frac{P_{\tilde{Z}}(\tilde{Z})}{\frac{1}{2}P_{\tilde{Z}}(\tilde{Z}) + \frac{1}{2}P_{\tilde{Z}}([\tilde{Z}-1] \bmod 2)}\right), \tag{16}$$

$$C(P) = \mathbb{E}\ i(\tilde{Z}),\ V(P) = \text{Var}\ i(\tilde{Z}). \tag{17}$$

Code for BAC phase: Next, we discuss the task of choosing a suitable code $\mathcal{C}_{\text{BAC}}$ for the BAC phase. This code should enables 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 $\mathcal{C}_{\text{BAC}}$ [11]. 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 [19] 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.[2] 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

$$\begin{aligned}\epsilon_3 &\triangleq \Pr(E_3|\bar{E}_{1,i}, \bar{E}_2) \\ &= \Pr\left(\cup_{i\neq j}\{W_i = W_j\}\right) \leq \frac{\binom{T}{2}}{2^k - 1}\end{aligned} \tag{18}$$

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

Next, we describe low complexity encoding and decoding algorithms for the BCH-based $\mathcal{C}_{\text{BAC}}$, whose computational

[2] In some cases the dimension may be smaller that kT. Nevertheless, the code we will use for $\mathcal{C}_{\text{BAC}}$, as elaborated in the main text, will always have dimension exactly kT.

cost is polynomial in k and T. Let $GF(2^k)$ be a Galois field. Our 2^k-1 possible messages can be identified with the vectors $\mathbb{F}_2^k \setminus \{\mathbf{0}\}$, where each of these vectors naturally corresponds to the element of $GF(2^k) \setminus \{0\}$ with matching coefficients in its polynomial representation. Thus, the BAC encoder maps the message vector $\mathbf{w}_i$ of user i to its corresponding element $\alpha_i \in GF(2^k) \setminus \{0\}$, and constructs the vector $\mathbf{v}_i = (\alpha_i, \alpha_i^3, \ldots, \alpha_i^{2T-1}) \in GF^T(2^k)$. The codeword $\mathbf{c}_i \in \mathbb{F}_2^{kT}$ is constructed by writing the binary coefficients of the polynomial representation of each element in $\mathbf{v}_i$, one after the other. It follows that each message vector from $\mathbb{F}_2^k \setminus \{\mathbf{0}\}$ is indeed mapped to a different column of the BCH parity check matrix. This procedure requires $O(T^2)$ multiplication in $GF(2^k)$.

The decoding procedure shares many similarities with standard Gorenstein-Peterson-Zierler (GPZ) decoding of BCH codes [19], 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 polynomial in k and T. We now describe the steps in decoding $\{\hat{\mathbf{c}}_{\text{BAC},1}, \ldots, \hat{\mathbf{c}}_{\text{BAC},L}\}$ from $\mathbf{y}_{\text{BAC},1}$. The procedure below is quite similar to the one described in [10]. Let $\alpha_1, \ldots, \alpha_L$ be the $GF(2^k) \setminus \{0\}$ elements corresponding to the messages $\mathbf{w}_{i_1}, \ldots, \mathbf{w}_{i_L}$ of the active users transmitting in the first sub-block.

1) Syndrome computations: Let $S_t = \sum_{i=1}^{L} \alpha_i^t$, $t = 1, \ldots, 2L$, be the set of required syndromes, and note that $\mathbf{y}_{\text{BAC},1}$ as a vector in $GF^T(2^k)$ is the vector of odd syndromes $(S_1, S_3, \ldots, S_{2T-1})$, where $L \leq T$ by assumption. We can easily recover the required even syndromes by recalling that $S_{2i} = S_i^2$ for all i.
2) Construction of error locator polynomial: We apply the Berlekamp-Massey algorithm to compute from $S_1, \ldots, S_{2L}$ the error locator polynomial
$$\sigma(X) = 1 + \sum_{t=1}^{L} \sigma_t X^t = \prod_{i=1}^{L} (1 + \alpha_i X), \qquad (19)$$
where $\{\sigma_t\}$ are the coefficients computed by the algorithm.
3) Finding the roots of $\sigma(X)$: We apply the probabilistic root finding algorithm from [20] (see also [21]) in order to find $(\alpha_1^{-1}, \ldots, \alpha_L^{-1})$.
4) Inversion of the roots: We invert α_i^{-1}, to get the desired α_i, $i = 1, \ldots, L$, whose binary polynomial representation coefficients are the message vectors $\mathbf{w}_i$. This is done by recalling that for any $\alpha \in GF(2^k) \setminus \{0\}$ we have that $1 = \alpha^{2^k-1} = \alpha \cdot \alpha^{2^k-2}$. Thus, $\alpha^{-1} = \alpha^{2^k-2}$, and can be computed by $(k-1)$ consecutive squaring of α, and multiplication of the result by α^{k-2}.

The number of sums and multiplications over $GF(2^k)$ required by the four steps is $O(T^2)$ for step 1, $O(T^2)$ for step 2 [19], $O(kT \log^2 T \log\log T)$ for step 3 [21], and $O(kT)$ for step 4.

B. Further comments

1) *Detection of number of active users per sub-block:* Our description of the decoder's operation assumed that the number of active users L in the sub-block is known. Although this is not the case in practice, we can apply the decoder $T+1$ times, each time assuming L took a different value in the set $\{0, 1, \ldots, T\}$. For each such "guess" of L the decoder will produce a list of decoded messages. We can use this list in order to create the corresponding codewords and subtract their sum from $\mathbf{y}_1$. If the correct value of L was guessed and the decoder was successful for this value, the resulting vector would be a pure AWGN vector $\mathbf{z}_1$. Otherwise, the result would be $\mathbf{z}_1$ plus the sum of certain codewords. We can therefore easily detect which value of L was the correct one, with a negligible error probability.
Alternatively, we could have used the fact that the first $\bar{n}R$ symbols of $\mathbf{y}_1$ are essentially i.i.d., due to the fact that $\mathcal{C}_{\text{lin}}$ is systematic, and the fact that the first $\bar{n}R$ symbols in our BCH-based code $\mathcal{C}_{\text{BAC}}$ are uniform on $\mathbb{F}_2^{\bar{n}R} \setminus \{\mathbf{0}\}$. The distribution of these (almost) i.i.d. random variables is dictated by L, and therefore we could estimate L from those symbols and bound the error probability by standard concentration of measure arguments.
We remark further that the value of L is only used by the decoder in order to shift the constellation used by each transmitters from $c\{-1,1\}$ to $2c\{0,1\}$, where $c = \sqrt{V \cdot P}$. The need for knowing L at the decoder can be bypassed altogether by using instead the constellation $\sqrt{2c}\{0,1\}$ at each transmitter, to begin with. The E_b/N_0 loss for using this asymmetric constellation which is less power efficient, instead of $c\{-1,1\}$, is 3dB.
2) *Dithers:* In compute-and-forward schemes, it is common to use random dithers, independent across users, known to the encoders and decoders (common randomness), in order to create a slightly better effective channel from $\mathbf{c}_1^{\oplus}$ to $\mathbf{y}_{\text{CoF},1}$, via linear minimum mean square error (MMSE) processing [9]. In our case creating such form of common randomness is impossible, as the decoder does not know which of the K_{tot} possible users were active within each sub-block, and consequently it does not know which of the dithers were used. One could generate the same random dither for all users, but it is not clear what performance can be guaranteed in this case.
3) *Codes with larger alphabets and shaping:* The performance of the CoF phase in our scheme can be improved by replacing our binary codebook $\mathcal{C}_{\text{lin}}$ mapped to a BPSK constellation, with a Voronoi codebook based on a "good" nested-lattice pair [15], [22]. The possible improvement is two-fold: i)A shaping gain of up to $10\log_{10}(2\pi e/12) \approx 1.53$dB, due to using an high-dimensional coarse lattice

(instead of the one-dimensional cubic shaping lattice used by the scheme described above). ii)The capacity achieving input distribution of a modulo-reduced additive noise channel (as is the channel (10)) is uniform on the modulo interval (Voronoi region). Our codebook $\mathcal{C}_{\text{lin}}$ on the other hand induces a distribution on a two point constellation. Using linear codes over larger prime fields $\mathbb{F}_p$ may therefore result in better performances [9], [15], [16], [22].

When the underlying field for the CoF phase is $\mathbb{F}_p$, the induced channel for the BAC phase will be a T-user modulo-p adder MAC. A capacity achieving codebook for this channel can be obtained using the parity check matrix of a $[n = p^s - 1, n - k = 2T]$ Reed-Solomon code, constructed over the field GF_{p^s-1}. More specifically, if $H = [\mathbf{h}_1 | \dots | \mathbf{h}_n]$ is the parity check matrix of this code, we construct the code

$$\mathcal{C}_{\text{BAC}} = \{\alpha \cdot \mathbf{h}_i \ : \ \alpha \in GF_{p^s-1} \setminus \{0\}, i = 1, \dots, n\}, \tag{20}$$

whose rate is $\frac{\log(p^s-1)^2}{2T \cdot s} \approx \frac{\log(p)}{T}$.

Despite this opportunity for improvement, this paper restricts attention to the case $p = 2$, as we believe that the severe computational requirements on the encoders in our setup makes the binary choice most practical.

III. Extensions

In this section we briefly describe three extensions of the basic scheme proposed in Section II. The first deals with the possibility to decode the *real* sum of the L codewords in the CoF phase, rather than their modulo-2 sum, the second deals with the scenario where the spectral efficiency is too high for binary codes to be applied, and third deals with a "near-far" users scenario, where the channel gains significantly differ across the different users.

A. Real sum decoding in the CoF phase

Assume the two error events $\{E_1, E_2\}$ did not occur. In this case, the decoder has access to $\mathbf{c}_1^{\oplus}$. We would now like to use $\mathbf{c}_1^{\oplus}$ and $\mathbf{y}_1$ in order to further decode $\sum_{j=1}^{L} \mathbf{c}_{\text{BAC},i_j} \in \{0, \dots, L\}^{\bar{n}R_{\text{lin}}}$. Let $\tilde{\mathbf{y}}_\mathbf{1} = (y_1, \dots, y_{\bar{n}R_{\text{lin}}})$ and (with some abuse of notation) $\tilde{\mathbf{z}}_\mathbf{1} = (z_1, \dots, z_{\bar{n}R_{\text{lin}}})/2\sqrt{V \cdot P}$, and set

$$\mathbf{y}_{1,\text{uncoded}} = \frac{1}{2\sqrt{V \cdot P}} \tilde{\mathbf{y}}_\mathbf{1} + \frac{L}{2} - \mathbf{y}_{\text{BAC},1} = \mathbf{y}_{\text{BAC},1}^{\text{uncoded}} + \tilde{\mathbf{z}}_1,$$

where $\mathbf{y}_{\text{BAC},1}$ is as defined in (11) and

$$\mathbf{y}_{\text{BAC},1}^{\text{uncoded}} \triangleq \sum_{j=1}^{L} \mathbf{c}_{\text{BAC},i_j} - \left[\sum_{j=1}^{L} \mathbf{c}_{\text{BAC},i_j} \right] \bmod 2 \in (2\mathbb{Z})^{\bar{n}R_{\text{lin}}}.$$

Now, setting $\hat{\mathbf{y}}_{\text{BAC},1}^{\text{uncoded}} \triangleq 2 \cdot \text{round}(\mathbf{y}_{1,\text{uncoded}}/2)$, we have that

$$\begin{aligned} \epsilon_{2b} &\triangleq E_{2b|\bar{E}_1,\bar{E}_2} \triangleq \Pr\left(\hat{\mathbf{y}}_{\text{BAC},1}^{\text{uncoded}} \neq \mathbf{y}_{\text{BAC},1}^{\text{uncoded}}\right) \\ &\leq 2\bar{n}R_{\text{lin}} \cdot Q\left(2\sqrt{V \cdot P}\right), \end{aligned} \tag{21}$$

where we have used the union bound in the last inequality. Note that in many applications of interest $\bar{n}R_{\text{lin}}$ is of moderate size ($\approx 100 - 1000$), and the total target error probability ϵ of the scheme is not required to be very small ($\approx 10^{-3} - 10^{-1}$). Thus, even though $\mathbf{y}_{\text{BAC},1}^{\text{uncoded}}$ consists of uncoded symbols, the resulting ϵ_{2b} is of acceptable value.

Now, assuming $\{E_1, E_2, E_{2b}\}$ did not occur, the receiver can compute

$$\tilde{\mathbf{y}}_{\text{BAC},1} \triangleq \mathbf{y}_{\text{BAC},1} + \mathbf{y}_{\text{BAC},1}^{\text{uncoded}} = \sum_{j=1}^{L} \mathbf{c}_{\text{BAC},i_j}. \tag{22}$$

Thus, using this modification, after the CoF phase the receiver has access to the output of a binary adder channel *with addition over the reals*, rather than a modulo-2 binary adder MAC as in (11).

The symmetric (per-user) capacity of the real binary adder T-user MAC is $\frac{H(\text{Binomial}(T,1/2))}{T} = \frac{1}{2T}\log_2\left(\frac{T\pi e}{2}\right) + O\left(\frac{1}{T^2}\right)$ bits per channel use [23], rather than only $1/T$ bits per channel use for the modulo-2 binary adder MAC. Thus, the additional step described here can potentially lead to great savings in E_b/N_0. However, an obstacle for realizing these gains in practice is that, to the best of the authors' knowledge, no efficient coding scheme for the real binary adder channel with the same codebook for all users is known to achieve rates greater than $1/T$. We also note in passing that while the restriction that all users transmit codewords from the same codebook does not decrease the symmetric capacity of the real binary adder channel, further insisting that this codebook be *linear* does (even under our setting where the decoded messages do not have to be associated with the users that sent them). Thus, the task of designing low complexity capacity approaching same-codebook schemes for the real binary adder channel seems quite challenging.

B. Higher spectral efficiency

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 (14)). 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 multi-level 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 $\mathbf{w}_i = (\mathbf{w}^a_i, \mathbf{w}^b_i) \in \mathbb{F}_2^{\bar{n}R^a - m} \setminus \{\mathbf{0}\} \times \mathbb{F}_2^{\bar{n}R^b - m} \setminus \{\mathbf{0}\}$. Then, user i draws an m-dimensional binary vector $\mathbf{u}_i$ with i.i.d. uniform entries, and creates the effective message vectors $\tilde{\mathbf{w}}^a_i = (\mathbf{u}_i, \mathbf{w}^a_i) \in \mathbb{F}_2^{\bar{n}R^a} \setminus \{\mathbf{0}\}$ and $\tilde{\mathbf{w}}^b_i = (\bar{\mathbf{u}}_i, \mathbf{w}^b_i) \in \mathbb{F}_2^{\bar{n}R^b} \setminus \{\mathbf{0}\}$, 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, $\tilde{\mathbf{w}}^a_i$ ($\tilde{\mathbf{w}}^b_i$) is encoded to a codeword $\mathbf{c}^a_i$ ($\mathbf{c}^b_i$) in $\mathcal{C}^a$ ($\mathcal{C}^b$) exactly as described in Section II, and the transmitted vector is

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

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

$$\mathbf{y}_{\text{CoF},1} = \frac{1}{2}\sqrt{\frac{5}{V \cdot P}} \left(\mathbf{y}_1 + \frac{3L}{2} \right) = \sum_{j=1}^{L} \mathbf{c}^a_{i_j} + 2 \sum_{j=1}^{L} \mathbf{c}^b_{i_j} + \tilde{\mathbf{z}}^a_1, \tag{23}$$

where $\tilde{\mathbf{z}}^a_1 = \frac{\sqrt{5}\mathbf{z}_1}{\sqrt{4V \cdot P}} \sim \mathcal{N}(\mathbf{0}, \sigma_a^2 \mathbf{I})$, $\sigma_a^2 = \frac{5}{4VP}$. Now, setting $\mathbf{y}^a_{\text{CoF},1} = [\mathbf{y}_{\text{CoF},1}] \bmod 2$, and continuing exactly as in the basic scheme from Section II, we can recover $\{\tilde{\mathbf{w}}^a_{i_1}, \ldots, \tilde{\mathbf{w}}^a_{i_L}\}$. This allows us to form $\sum_{j=1}^{L} \mathbf{c}^a_{i_j}$, and then construct

$$\begin{aligned} \mathbf{y}^b_{\text{CoF},1} &= \left[\frac{1}{2} \left(\mathbf{y}_{\text{CoF},1} - \sum_{j=1}^{L} \mathbf{c}^a_{i_j} \right) \right] \bmod 2 \\ &= \left[\sum_{j=1}^{L} \mathbf{c}^b_{i_j} + \tilde{\mathbf{z}}^b_1 \right] \bmod 2, \end{aligned} \tag{24}$$

where $\tilde{\mathbf{z}}^b_1 \sim \mathcal{N}(\mathbf{0}, \sigma_b^2 \mathbf{I})$, $\sigma_b^2 = \frac{5}{16VP}$. We can now recover $\{\tilde{\mathbf{w}}^b_{i_1}, \ldots, \tilde{\mathbf{w}}^b_{i_L}\}$, exactly as in the basic scheme from Section II. The effective channel $\tilde{\mathbf{z}}^b_1$ is "cleaner" than $\tilde{\mathbf{z}}^a_1$, therefore we will choose $\mathcal{C}^a_{\text{lin}}, \mathcal{C}^b_{\text{lin}}$ such that $R^a_{\text{lin}} \leq R^b_{\text{lin}}$, where their exact values should be optimized w.r.t. the target error probability and to $V \cdot P$. The codes $\mathcal{C}^a_{\text{BAC}}, \mathcal{C}^b_{\text{BAC}}$ for the BAC phase are both BCH-based codes of rate $R^a_{\text{BAC}} = R^b_{\text{BAC}} = 1/T$, as described in Section II-A, where they only differ in their blocklengths $\bar{n}R^a_{\text{lin}}$ and $\bar{n}R^b_{\text{lin}}$, respectively.

The final step is to use the two lists $\{\tilde{\mathbf{w}}^a_{i_1}, \ldots, \tilde{\mathbf{w}}^a_{i_L}\}$ and $\{\tilde{\mathbf{w}}^b_{i_1}, \ldots, \tilde{\mathbf{w}}^b_{i_L}\}$ 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 $\tilde{\mathbf{w}}_{i_j} = (\tilde{\mathbf{w}}^a_{i_j}, \tilde{\mathbf{w}}^b_{i_j})$, 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

$$\epsilon_4 = 1 - \prod_{\ell=1}^{L-1} (1 - \ell 2^{-m}) \leq T(T-1) \cdot 2^{-(m+1)}. \tag{25}$$

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.

C. Unbalanced channel gains

In the schemes discussed thus far, we have assumed all users have the same channel gain. In practice, of course, this is never the case. Nevertheless, if each user i knows its gain h_i to the receiver (which can be attained by, e.g., exploiting reciprocity), it can scale its codeword by $1/h_i$, to create an effective channel gain of $\tilde{h}_i = 1$. When this strategy is taken by all users, we get the symmetric channel model that was treated above, where the required E_b/N_0 for user i is the one for the symmetric model, multiplied by $1/h_i^2$.

Often, the magnitudes of the channel gains, and consequently the required E_b/N_0, significantly vary between the users. Below, we describe a modification of our scheme that enables to reduce the required E_b/N_0 of the weak users at the expense of increasing the E_b/N_0 for the strong users, and therefore create a somewhat more balanced distribution of the resources among users.

The main idea in the proposed modification is that instead of multiplying its codeword by $1/h_1$, in order to equalize its channel gain to $\tilde{h}_i = 1$, the ith user equalizes its gain h_i to some number in the grid $\mathcal{G} \triangleq \{1 = 2^0, 2^1, 2^2, \cdots, 2^b\}$, for some natural b, where the mapping between values of h_i and points in the grid is according to some predetermined monotonically increasing quantization function $q : \mathbb{R}_+ \mapsto \mathcal{G}$ whose input is $|h_i|$.

Let L be the number of users that transmitted during the first sub-block, and let $L_m \leq L$ be the number of users that transmitted in the first sub-block whose effective gain is $\tilde{h}_i = 2^m$, such that $L = \sum_m L_m$. We compute

$$\mathbf{y}_{\text{CoF},1} = \frac{1}{2\sqrt{V \cdot P}} \mathbf{y}_1 + \sum_{m=0}^{b} 2^m \frac{L_m}{2} = \sum_{m=0}^{b} 2^m \sum_{j=1}^{L_m} \mathbf{c}_{i_{m_j}} + \tilde{\mathbf{z}}_1,$$

where $\tilde{\mathbf{z}}_1 \sim \mathcal{N}(\mathbf{0}, \sigma^2)$, $\sigma^2 = \frac{1}{4V \cdot P}$. The decoding can now be performed in a successive cancellation manner. First we compute $\mathbf{y}^0_{\text{CoF},1} = [\mathbf{y}_{\text{CoF},1}] \bmod 2 = [\sum_{j=1}^{L_1} \mathbf{c}_{i_{1_j}} + \tilde{\mathbf{z}}_1] \bmod 2$,

from which we can decode the messages of the weakest users. Next, we subtract the sum of corresponding codewords from $\mathbf{y}_{\text{CoF},1}$, divide by 2 and reduce modulo 2 to get $\mathbf{y}^1_{\text{CoF},1} = [\sum_{j=1}^{L_2} \mathbf{c}_{i_{1_j}} + \frac{\tilde{\mathbf{z}}_1}{2}] \bmod 2$, from which we decode the group of messages transmitted by the users whose gains satisfy $q^{-1}(|h_i|) = 2^1$, and so on.

The advantage of this approach is that now the decoding can succeed if $L_m \leq T$ for all m, which is a weaker constraint than $L = \sum_m L_m \leq T$. When the distribution of channel magnitudes $|h_i|$ among the K_{tot} users is known in advance, the function $q(\cdot)$ can be chosen to induce a favorable distribution on $\{L_m\}$, which in turn leads to the possibility of decreasing T without increasing the "forbidden collisions" probability ϵ_1. Note that potentially, we can use a different codebook for each group of users, where the users are grouped according to their equalized channel gain. Indeed, the signal-to-noise ratio (SNR) for the channel $\mathbf{y}^m_{\text{CoF},1}$ is $6m$ dBs better than that of $\mathbf{y}^0_{\text{CoF},1}$. Thus, the rate for $\mathcal{C}^m_{\text{lin}}$, the inner linear code for the mth group of users, can increase with m. Since each user only has a fixed number of k bits to send, and that the transmission power of each user is fixed according to its group, the increase in R_{lin} can only be exploited for enabling to deal with a larger number of collisions. Specifically, increasing R^m_{lin} can allow to use codes $\mathcal{C}^m_{\text{BAC}}$, with lower rates $R^m_{\text{BAC}} = \frac{1}{T_m}$, that can decode whenever at most T_m users from group m simultaneously transmitted within the same sub-block.

IV. Choice of Code Parameters and Numerical Evaluation

In this section we evaluate the E_b/N_0 required by our scheme, first for the basic setup described in Section II, and then with the extension discussed in Section III-B. Fix k, n, and K_a, and assume a moderate target error probability P_e is required, say between 10^{-3} and 10^{-1}. For the basic scheme, the error probability is upper bounded by $P_e \leq \epsilon_1 + \epsilon_2 + \epsilon_3$, where ϵ_1 is as defined in (14), ϵ_2 is the error probability of the CoF phase, and ϵ_3 corresponds to the BAC phase, and can be neglected when the BCH-based code $\mathcal{C}_{\text{BAC}}$ is used, as seen from by (18).

We fix target probabilities ϵ_1, ϵ_2 such that $\epsilon_1 + \epsilon_2 = \epsilon$, and assume temporarily that T is also fixed. Let $\alpha^*(\epsilon_1)$ be the solution of the equation (14) in α. By the monotonicity of $\Pr(\text{Binomial}(K_a - 1, \frac{\alpha T}{K_a}) < T)$ in α, we have that all $\alpha \in [0, \alpha^*(\epsilon_1)]$, would lead to "T-collision" probability smaller than ϵ_1. Choosing some α from this interval and 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 of the linear code $\mathcal{C}_{\text{lin}}$ must satisfy

$$R_{\text{lin}} = \frac{Tk}{\bar{n}} = \frac{Tk}{\alpha T n / K_a} = \frac{\rho}{\alpha},$$

whereas the blocklength for this code is $\bar{n} = \alpha T n / K_a$. Let $\tilde{P} = V \cdot P$ be the average transmission power of an active user within its sub-block and let $\tilde{P}(r, n, \epsilon)$ be the smallest $\tilde{P}$ for which there exists a rate r linear code of blocklength n, that achieves error probability ϵ over the channel (10). Note that $\tilde{P}(r, n, \epsilon)$ can be found using (15). Now, recalling that $V = K_a/(\alpha T)$ and using the definition of E_b/N_0 from (4), we see that our basic scheme requires

$$\frac{E_b}{N_0} = \frac{n}{2k} \frac{\tilde{P}\left(\frac{\rho}{\alpha}, \frac{\alpha T n}{K_a}, \epsilon_2\right)}{K_a/(\alpha T)} = T \cdot \frac{\tilde{P}\left(\frac{\rho}{\alpha}, \frac{\alpha T n}{K_a}, \epsilon_2\right)}{2\frac{\rho}{\alpha}}. \quad (26)$$

Recall that the infinite blocklength fundamental limit for transmitting ρ bits per channel use for the AWGN channel is $(E_b/N_0)^* = (2^{2\rho} - 1)/2\rho$. In a coordinated MAC (unlike our random access one), $(E_b/N_0)^*$ can be achieved asymptotically for $n \to \infty$, K_a fixed and $k = n\rho/K_a$, by, e.g., TDMA. Contrasting the performance of our scheme (26) with $(E_b/N_0)^*$, and ignoring finite blocklength effects for the sake of the discussion, we identify three different losses:

- Our E_b/N_0 scales with T. This is due to the fact that we decode the modulo-2 sum of the codeword rather than their real sum, and therefore our scheme does not exploit the fact that the received constellation has average power $T\tilde{P}$ rather than $\tilde{P}$;
- The effective spectral efficiency for our scheme is $1/\alpha$ higher than that required by the TDMA scheme. This is due to the fact that our scheme is designed to avoid more than T collisions, which in turn leads to a less efficient use of the channel resources.
- Our scheme reduces the communication channel to a modulo-2 AWGN channel (10), rather than an AWGN one. The capacity of this channel can be roughly approximated by $\frac{1}{2}\log_2^+(\frac{12}{2\pi e}\tilde{P})$. Contrasting this with the AWGN channel capacity $\frac{1}{2}\log(1+\tilde{P})$, we identify a shaping loss of $10\log_{10}(2\pi e/12) \approx 1.53$ dB and an additional power loss of $-10\log_{10}(1-2^{-2\rho})$ dB, due to the loss of $+1$ in the capacity expression for the modulo-2 AWGN channel. The latter loss becomes negligible as the spectral efficiency increases, whereas the former is independent of ρ.

Thus, a rough estimate on the loss of our scheme w.r.t. hypothetical TDMA is

$$\begin{aligned}\Delta &= \left(\frac{E_b}{N_0}\right) \text{dB} - \left(\frac{E_b}{N_0}\right)^* \text{dB} \\ &\approx 10\log_{10}\left(T \cdot \frac{2^{2\rho/\alpha}}{2\rho/\alpha} \cdot \frac{2\rho}{2^{2\rho}(1-2^{-2\rho})} \cdot \frac{2\pi e}{12}\right) \\ &\approx 1.53 - 10\log_{10}(1-2^{-2\rho}) + 6\rho\frac{1-\alpha}{\alpha} + 10\log_{10}(\alpha T)\ \text{dB}, \end{aligned} \quad (27)$$

which can be minimized w.r.t. $\alpha \in [0, \alpha^*(\epsilon_1)]$.[3]

Recall Ungerboeck's rule of thumb [24] that states that the information rates for communicating over an AWGN channel with binary inputs are close to capacity when capacity is below $1/2$ a bit per channel use, but significantly diverge

[3]The minimizing value is typically the extreme one $\alpha^*(\epsilon_1)$.

$\mathbf{K_a}$	**20**	**50**	**100**	**150**	**200**	**250**	**300**
$\mathbf{E_b/N_0}$**[dB]**	7.38	8.83	11.89	15.00	17.32	20.65	23.02
T	1	3	5	5	9	13	12
α	0.048	0.269	0.389	0.385	0.513	0.584	0.573
τ	2	1	1	2	2	2	3
Rate for Linear Code	1.38	0.62	0.86	1.61	1.66	1.85	2.53

Fig. 2. Optimized parameters

from the capacity when it grows above this value. The same behavior is also true for communication over the modulo-2 AWGN channel (10). Thus, when α and ρ are such that $\rho/\alpha > 1/2$, binary codes are insufficient and we use the multilevel construction described in Section III-B.

The performance analysis remains quite similar to that of the basic scheme, where the main difference is that we have an additional error event E_4, which corresponds to the "pairing" of message vectors decoded in the different layers. Consequently, we need to add an m-bit prefix to each message, where $m = \lceil \log_2(T(T-1)/\epsilon_4) \rceil - 1$ and ϵ_4 is the target error probability for the event E_4. This in turn, increases the required spectral efficiency for the linear code from $\frac{\rho}{\alpha}$ to $\frac{\rho(1+\tau\gamma)}{\alpha}$, where $\gamma \triangleq m/k$ and τ is the number of layers in our code. Using similar calculations to those we performed above, lead to the following rough estimate on the asymptotic gap (in dB) between the E_b/N_0 required by our multi-layer scheme and the hypothetical TDMA

$$\Delta \approx 1.53 - 10\log_{10}(1-2^{-2\rho}) + 6\rho\frac{1+\tau\gamma-\alpha}{\alpha} + 10\log_{10}(\alpha T) \quad (28)$$

where according to Ungerboeck's rule of thumb, the approximation is valid if the number of layers satisfies $\tau > \frac{\rho(1+\tau\gamma)}{\alpha} + \frac{1}{2}$ which is equivalent to $\tau > \frac{2\rho+\alpha}{2(\alpha-\rho\gamma)}$.

We perform a numerical evaluation of the proposed scheme's performance, and compare it to relevant benchmarks in Figure 1. For the evaluation we took $k = 100$, $n = 30,000$, $P_e = 0.05$ and K_a varies from 20 to 300. This is the regime of interest for LP-WANs such as LoRaWAN and Weightless. We plot the E_b/N_0 required by our scheme with the parameters T, α, and τ optimized. For all values of $20 \leq K_a \leq 300$, the required values of T were between $T = 1$ and $T = 13$. We also plot the E_b/N_0 required by other related schemes, as discussed in Section I. In the calculations, we have always chosen $\epsilon_1 = 0.9\epsilon$, $\epsilon_4 = 0.05\epsilon$ and $\epsilon_2 = 0.05\epsilon$, where ϵ_2 was equally split between the τ levels when multilevel codes were used. The optimal values of T, α and τ, as well as the corresponding rate for the linear code (or sum of rates when $\tau > 1$), is given in Table IV for selected values of K_a.

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 Transactions on Information Theory*, vol. 31, no. 2, pp. 124–142, Mar 1985.
[2] Y. Polyanskiy, "A perspective on massive random-access," in *Submitted to ISIT 17*, 2017.
[3] R. Ahlswede, "Multi-way communication channels," in *ISIT, Tsahkadsor, Armenia, USSR, Sept. 2-8, 1971*, Sept. 1973.
[4] H. Liao, "Multiple access channels," Ph.D. dissertation, University of Hawaii, Honolulu, 1972.
[5] A. El Gamal and Y.-H. Kim, *Network information theory*. Cambridge University Press, 2011.
[6] X. Chen, T. Chen, and D. Guo, "Capacity of Gaussian many-access channels," *CoRR*, vol. abs/1607.01048, 2016. [Online]. Available: http://arxiv.org/abs/1607.01048
[7] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted aloha," *IEEE Transactions on Communications*, vol. 59, no. 2, pp. 477–487, February 2011.
[8] K. R. Narayanan and H. D. Pfister, "Iterative collision resolution for slotted aloha: An optimal uncoordinated transmission policy," in *2012 7th International Symposium on Turbo Codes and Iterative Information Processing (ISTC)*, Aug 2012, pp. 136–139.
[9] B. Nazer and M. Gastpar, "Compute-and-forward: Harnessing interference through structured codes," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6463–6486, Oct. 2011.
[10] I. Bar-David, E. Plotnik, and R. Rom, "Forward collision resolution-a technique for random multiple-access to the adder channel," *IEEE Transactions on Information Theory*, vol. 39, no. 5, pp. 1671–1675, Sep 1993.
[11] B. Lindström, "Determination of two vectors from the sum," *Journal of Combinatorial Theory*, vol. 6, no. 4, pp. 402 – 407, 1969.
[12] T. Ericson and V. I. Levenshtein, "Superimposed codes in the hamming space," *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 1882–1893, Nov 1994.
[13] M. P. Wilson, K. Narayanan, H. Pfister, and A. Sprintson, "Joint physical layer coding and network coding for bidirectional relaying," *IEEE Transactions on Information Theory*, vol. 11, no. 56, pp. 5641–5654, November 2010.
[14] J. Zhan, B. Nazer, U. Erez, and M. Gastpar, "Integer-forcing linear receivers," *IEEE Transactions on Information Theory*, vol. 60, no. 12, pp. 7661–7685, Dec 2014.
[15] R. Zamir, *Lattice Coding for Signals and Networks*. Cambridge: Cambridge University Press, 2014.
[16] O. Ordentlich and U. Erez, "A simple proof for the existence of "good" pairs of nested lattices," *IEEE Transactions on Information Theory*, vol. 62, no. 8, pp. 4439–4453, Aug 2016.
[17] J. Körner and K. Marton, "How to encode the modulo-two sum of binary sources," *IEEE Transactions on Information Theory*, vol. 25, no. 2, pp. 219–221, March 1979.
[18] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Channel coding rate in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2307–2359, May 2010.
[19] S. Lin and D. J. Costello, *Error control coding*. Pearson Education, 2004.
[20] M. O. Rabin, "Probabilistic algorithms in finite fields," *SIAM Journal on Computing*, vol. 9, no. 2, pp. 273–280, 1980.
[21] M. Ben-Or, "Probabilistic algorithms in finite fields," in *22nd Annual Symposium on Foundations of Computer Science (sfcs 1981)*, Oct 1981, pp. 394–398.

[22] U. Erez and R. Zamir, "Achieving $\frac{1}{2}\log(1+\mathrm{SNR})$ on the AWGN channel with lattice encoding and decoding," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2293–2314, Oct. 2004.
[23] L. A. Shepp and J. Olkin, "Entropy of the sum of independent Bernoulli random variables and of the multidimensional distribution," in *Contributions to probability*. New York: Academic Press, 1981, pp. 201–206.
[24] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, no. 1, pp. 55–67, January 1982.