On Coding Techniques for Unsourced Multiple-Access

Gianluigi Liva
German Aerospace Center (DLR)
gianluigi.liva@dlr.de

Yury Polyanskiy
Massachusetts Institute of Technology (MIT)
yp@mit.edu

***Abstract*—In this paper, we attempt to gain insights on designing codes for unsourced multiple access by investigating an important special case of a two-user unsourced binary adder channel (2-UBAC). In 2-UBAC the receiver observes a noiseless real sum of two binary vectors. We show several results. First, for a linear code the per-user probability of error (PUPE) equals the fraction of nonminimal codewords, implying that such codes can at most achieve rate-$1/2$ while capacity of 2-UBAC is $3/4$. Second, for sparse-graph codes to jump start an iterative peeling decoder we need to reveal ("pivot") one of the ambiguous symbols. If the pivot is selected randomly then any irregular LDPC code ensemble has a non-vanishing error probability. If the pivot is selected optimally then we show that three regular LDPC code ensembles attain vanishing PUPE: $(3,4)$, $(4,5)$ and $(5,6)$. Our proof does not apply to any other regular LDPC code ensembles, but we believe that they should all have a non-vanishing error probability. Finally, we discuss ideas about (nonlinear) coding to break through the rate-$1/2$ bottleneck.**

I. Introduction

The rising importance of large-scale Internet of Things (IoT) networks and massive machine-type communication (mMTC) systems has recently produced an intense research effort in the domain of multiple access (MAC) protocols (see, e.g., [1]–[10]). While it appears evident that random access protocols are the most suitable choice to handle the sporadic and unpredictable transmissions of large populations of IoT terminals, many implemented systems [11], [12] still employ simple modifications of the basic ALOHA protocol [13]. To overcome the limitations that are inherent to (slotted) ALOHA, many advanced random access schemes were proposed during the past two decades. Some of them stem from the introduction of advanced signal processing capabilities at the receiver end, such as the use of successive interference cancellation (SIC), applied to judiciously-designed ALOHA-like protocols [2]–[6]. More recently, another class of random access schemes was suggested in [1], where the salient features of the random multiple access problem have been cast in an information theoretic setup. In a nutshell, [1] regards the random access problem as a MAC coding problem where all transmitters employ the same code, and where the receiver is only interested in obtaining the list of transmitted codewords. Since the use of a unique code does not allow to distinguish the transmitters identity, the setting of [1] is usually referred to as unsourced multiple access (UMAC). The definition of the UMAC coding problem lead to the introduction of schemes inspired by compressive sensing techniques [1], [7]–[10].

This material is based upon work supported by the NSF grants CCF-1717842 and CCF-2131115.

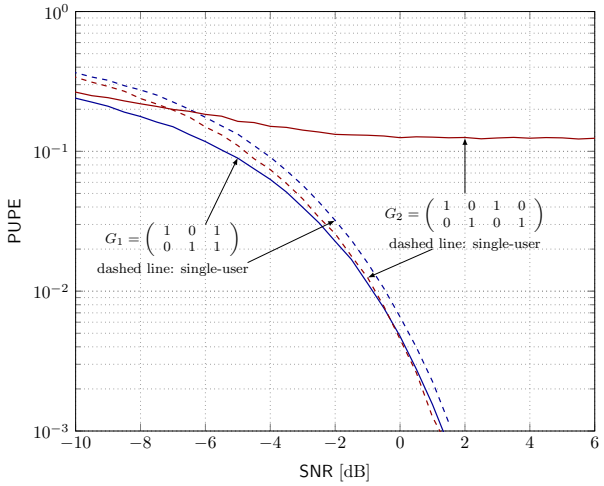

Fig. 1. Per-user probability of error vs. per-user signal-to-noise ratio (in dB) for two binary linear block codes over a binary adder Gaussian noise channel. Denoting by $A, B \in \{\pm 1\}$ the symbols transmitted by the two users, the channel output is $Y = A + B + N$ where N is Gaussian with zero mean and variance $\sigma^2 = 1/\mathsf{SNR}$. The decoder outputs the unordered part of codewords $\{\hat{v}^n, \hat{w}^n\}$ maximizing $p(y^n | v^n + w^n)$. The block error probability of the codes in absence of interference (i.e., single-user transmission) is depicted as reference with dashed lines. The estimates of the error probabilities are obtained through Monte Carlo simulations by collecting at 10^5 errors.

To gain insights on the coding problem underpinned by the UMAC setup, in this paper we address one of the simplest UMAC channels, namely the two-user unsourced binary adder channel (2-UBAC). In the 2-UBAC, two users transmit over the binary adder channel (BAC) with the constraint of using the same binary block code. Despite of its simplicity, this setting turns to be particularly rich from a coding theory viewpoint. Furthermore, the setting bears some relation with more realistic MAC channel modes. To support this statement, an example is provided in Figure 1, where the per-user probability of error (PUPE) is provided for the case of a real-valued Gaussian MAC channel where two users attempt a transmission under the constraint of using the same binary linear block code. The performance is measured as a function the per-user signal-to-noise ratio (SNR), defined as ratio between the power of the user signal and the noise variance. On the same plot, the block error rate is given as reference for the single-user case (i.e., in absence of interference). Two short codes are used for the simulation: a $(3,2)$ single-parity-check (SPC) code and a $(4,2)$ binary linear code. Both codes have minimum distance equal to two, and the $(4,2)$ code slightly outperforms the $(3,2)$ code in the single-user setting. However,

when transmission takes place over the two-user Gaussian MAC channel, the $(3,2)$ code yields a dramatically better performance, with the $(4,2)$ code suffering for a high error floor. The reason of this behavior can be found by analyzing the two codes in the 2-UBAC setting. As we will see in Section III, the $(3,2)$ code delivers a zero PUPE, whereas the $(4,2)$ code is limited to a PUPE of $1/8$ – which is the value at which the code performance floors over the Gaussian MAC.

In this paper, we are going to discuss the performance of binary linear block codes over the 2-UBAC, with emphasis on low-density parity-check (LDPC) code ensembles. The analysis addresses both maximum likelihood (ML) and iterative (peeling) erasure decoding. We are going to see how the performance of binary linear block codes is tightly related to a number of minimal codewords in the code, yielding rates that are remarkably lower than the capacity of the 2-UBAC channel with nonlinear coding. In particular, for binary linear codes the maximum symmetric rate is shown to be $1/2$, whereas the use of nonlinear codes allows achieving a symmetric rate equal to $3/4$. For the case of LDPC codes, the performance under peeling decoding will be put in relation with structure of the stopping sets that are present in certain sub-graphs of the code Tanner graph. Finally, we discuss ideas about (nonlinear) coding to break through the rate-$1/2$ bottleneck.

II. Preliminaries

We denote length-n vectors as $x^n = (x_1, x_2, \ldots, x_n)$. The vectors 0^n and 1^n are the length-n all-zero and all-one vectors, respectively. We use capital letters for random variables, e.g., X, and lower case letters for their realizations, e.g., x. We use the shorthand notation $[n] = \{1, 2, \ldots, n\}$. The support set of a vector is defined as $\mathrm{supp}(x^n) = \{i \in [n] : i \neq 0\}$. The Hamming weight of a vector x^n is $w_H(x^n) = |\mathrm{supp}(x^n)|$. We use $+$ and $-$ to denote integer addition and subtraction, whereas $\oplus$ is used for addition/subtraction in $\mathbb{F}_2$ (the order-2 finite field). For (n,k) binary linear block code C the rate is $R = k/n$, with k being the code dimension and n its blocklength. Finally, we use the shorthand $[x]^+ = \max(0, x)$ and we denote by H_b the binary entropy function.

A. Two-User Multiple-Access Channels

In this paper we consider the following two channels. First, a two-user BAC: the channel output is obtained by adding in $\mathbb{N}$ the binary symbols $A, B \in \{0,1\}$ transmitted by two users. The input-output relation at time i is hence

$$Y_i = A_i + B_i.$$

A generalization of this channel model is the *A-channel* of Chang and Wolf [14]. This channel has q-ary input alphabet $A, B \in [q]$ and the channel output Y belongs to $\left\{\binom{[q]}{1}\right\} \cup \left\{\binom{[q]}{2}\right\}$ with

$$Y_i = \{A_i, B_i\}.$$

That is, the receiver noiselessly observes the set of transmitted symbols (but not who transmitted them and also not the multiplicity, although for 2 users, the multiplicity can be inferred from the cardinality of Y). For $q = 2$ the A-channel is just the BAC.

The setting we are interested in is the one where the two users transmit over the BAC using the same (n, M) binary block code C. Here, n is the blocklength, and $M = |C|$ and the per-user rate is $R = n^{-1} \log_2 M$. Upon observing the channel output $Y^n = (Y_1, Y_2, \ldots, Y_n)$, the decoder produced a list $\mathcal{L}$ of two codewords. The PUPE is the probability that the codeword transmitted by a user does not appear in the decoded list, i.e.,

$$\mathsf{PUPE} = \frac{1}{2}\mathsf{P}(A^n \notin \mathcal{L}) + \frac{1}{2}\mathsf{P}(B^n \notin \mathcal{L}).$$

We define also the decoding failure probability (DFP) as the probability that there exists multiple distinct unordered codeword pairs, where the sum of the two codewords forming each pair yields Y^n. It is trivial to see that the DFP provides an upper bound on the PUPE.

B. Minimal Codewords

Consider two length-n binary vectors v^n and w^n. We say that v^n *covers* w^n if the support of w^n is a strict subset of the support of v^n. A codeword x^n of a binary linear code $C(n,k)$ is said to be minimal if it does not cover any non-zero codeword [15], [16]. A code whose codewords are all minimal is referred to as a minimal code. We denote in the following $\mathcal{M}(C)$ the subset of minimal codewords of the binary linear block code C. Properties of minimal codewords have been analyzed in [17]. One property, in particular, will turn to be useful in Section IV. We recall it next.

Lemma 1. *In a binary linear block code $C(n,k)$, any codeword with Hamming weight larger than $n - k + 1$ is nonminimal.*

C. LDPC Code Ensembles

We consider unstructured LDPC code ensembles. We denote the code bipartite graph (Tanner graph) as $\mathcal{G}$. For regular LDPC codes, the variable node (VN) degree (left degree) is $\mathtt{l}$ and the check node (CN) degree (right degree) is $\mathtt{r}$. A $(\mathtt{l},\mathtt{r})$-regular LDPC code ensemble is denoted as $\mathcal{C}^n_{\mathtt{l},\mathtt{r}}$ where n is the blocklength of the codes in the ensemble. For irregular LDPC code ensembles, the node oriented degree distributions are denoted as Λ and P, where Λ_d is the fraction of VNs with degree d and P_d is the fraction of CNs with degree d. An irregular LDPC code ensemble is denoted as $\mathcal{C}^n_{\Lambda,\mathrm{P}}$ where n is the blocklength of the codes in the ensemble. The code bipartite graph contains n VNs $\mathsf{v}_1, \mathsf{v}_2 \ldots, \mathsf{v}_n$ and m CNs $\mathsf{c}_1, \mathsf{c}_2, \ldots, \mathsf{c}_m$. The rate of a code C is $R(C)$, while the ensemble design rate is $R_0 = 1 - m/n$, where in general $R_0 \leq R(C)$. We denote the neighborhood of a VN v (CN c) as $\mathcal{N}(\mathsf{v})$ ($\mathcal{N}(\mathsf{c})$). The definition extends to set of nodes, e.g., the set of neighbors of a subset $\mathcal{V}$ of VNs is denoted as $\mathcal{N}(\mathcal{V})$. A stopping set $\mathcal{S}$ is a subset of VNs such that every CN in $\mathcal{N}(\mathcal{S})$ is connected through at least two VNs in $\mathcal{S}$.

The bipartite graph induced by a (non-zero) codeword x^n is denoted as $\mathcal{G}(x^n)$ and it is the subgraph of $\mathcal{G}$ formed by the VNs with indexes in $\mathrm{supp}(x^n)$, their adjacent edges and their

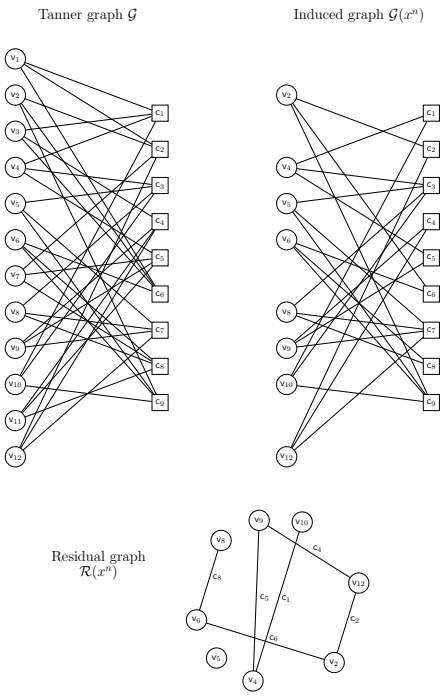

Fig. 2. Tanner graph of a length-12 $(3, 4)$ regular LDPC code (top-left). The graph induced by the codeword $x^n = (0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1)$ is provided in the top-right. Below, the residual graph, with edge labels showing the associated degree-2 check nodes in the induced graph.

neighboring CNs. We refer to $\mathcal{G}(x^n)$ as the *induced graph*. Note that the degrees of the CNs in $\mathcal{G}(x^n)$ must be even. We denote the CN degree distribution of $\mathcal{G}(x^n)$ as R, where R_d is the fraction of CNs with degree d, and the VN degree distribution of $\mathcal{G}(x^n)$ as L, where L_d is the fraction of VNs with degree d. Note that for $(\mathtt{l}, \mathtt{r})$ regular LDPC codes the degree of the VNs in $\mathcal{G}(x^n)$ is still $\mathtt{l}$.

We further introduce the notion of *residual graph* $\mathcal{R}(x^n)$ of an induced graph $\mathcal{G}(x^n)$, that is a graph consisting of $w_H(x^n)$ vertexes labeled as the corresponding $w_H(x^n)$ VNs in $G(x^n)$, where vertexes v_i and v_j are connected by an edge if and only if v_i and v_j in $\mathcal{G}(x^n)$ are connected to the same degree-2 CN. The degree distribution of the vertexes of $\mathcal{R}(x^n)$ is λ, with λ_d being the fraction of degree-d vertexes. An example of Tanner graph for a regular LDPC code, an induced graph $\mathcal{G}(x^n)$ and the corresponding residual graph is given in Figure 2.

The weight (stopping set) enumerator of a code is A_w^{CW} (A_w^{SS}), while the average weight (stopping set) enumerator of a random code from an ensemble is $\bar{A}_w^{\mathrm{CW}}$ ($\bar{A}_w^{\mathrm{SS}}$). Given an ensemble sequence $\mathcal{C}^n$, we denote the growth rate of the codeword weight distribution and of the stopping set weight distribution as

$$\mathsf{G}^{\mathrm{CW}}(\omega) = \lim_{n\to\infty} \frac{1}{n} \log_2 \bar{A}_{\omega n}^{\mathrm{CW}} \quad \text{and} \quad \mathsf{G}^{\mathrm{SS}}(\omega) = \lim_{n\to\infty} \frac{1}{n} \log_2 \bar{A}_{\omega n}^{\mathrm{SS}}.$$

Finally, for a given LDPC code Tanner graph, we introduce $A_{w,\mathrm{L},\mathrm{R}}^{\mathrm{IG}}$ as the number of graphs with VN degree distribution L and CN degree distribution R, induced by weight-w codewords. We refer to the $(w, \mathrm{L}, \mathrm{R})$ triplet as the *induced graph type*. The ensemble average is $\bar{A}_{w,\mathrm{L},\mathrm{R}}^{\mathrm{IG}}$, whereas the growth rate of the type-$(\omega n, \mathrm{L}, \mathrm{R})$ induced graph multiplicity is

$$\mathsf{G}^{\mathrm{IG}}(\omega, \mathrm{L}, \mathrm{R}) = \lim_{n\to\infty} \frac{1}{n} \log_2 \bar{A}_{\omega n,\mathrm{L},\mathrm{R}}^{\mathrm{IG}}.$$

III. Maximum-Likelihood Decoding over the Binary Adder Channel: Linear Codes

ML decoding over the BAC can be formulated as an erasure decoding problem. Given the observation y^n, we may construct a vector $\tilde{y}^n$ by setting $\tilde{y}_i = 0$ if $y_i = 0$, $\tilde{y}_i = 1$ if $y_i = 2$, and $\tilde{y}_i =?$ if $y_i = 1$. Here, the symbol "?" represents an erasure. The set of coordinates in which $\tilde{y}^n$ is set to "?" is

$$\mathcal{E} = \{i | \tilde{y}_i =?\}$$

and its complementary set is $\bar{\mathcal{E}} = [n] \setminus \mathcal{E}$. Decoding proceeds by creating a list $\tilde{\mathcal{L}}$ of all codewords that are compatible with the modified observation $\tilde{y}^n$, i.e.

$$\tilde{\mathcal{L}} = \left\{ x^n | x_i = \tilde{y}_i, \, \forall i \in \bar{\mathcal{E}} \right\}.$$

The final step consists of searching all unordered codeword pairs $\{v^n, w^n\} \in \tilde{\mathcal{L}} \times \tilde{\mathcal{L}}$ whose (integer) sum yields y^n. We refer to a pair $\{v^n, w^n\}$ satisfying $y^n = v^n + w^n$ as a *valid (codeword) pair*. If the solution is unique, then the final list is $\mathcal{L} = \{v^n, w^n\}$ and decoding succeeds. In case of multiple valid pairs, a pair is picked at random. We refer to the event where decoding yields multiple valid pairs as a decoding failure. According to the definition provided in Section II-A the probability of such event is the DFP.

The construction of the list of compatible codewords is largely simplified in the case of an (n, k) binary linear block code C. Denote by H the $(n-k)\times n$ code parity-check matrix, by $x_{\mathcal{E}}$ ($x_{\bar{\mathcal{E}}}$) the sub-vector of x^n with coordinates in $\mathcal{E}$ ($\bar{\mathcal{E}}$). Similarly, we denote by $H_{\mathcal{E}}$ ($H_{\bar{\mathcal{E}}}$) the matrix formed by the columns of H with coordinates in $\mathcal{E}$ ($\bar{\mathcal{E}}$). The list of compatible codewords is formed by all vectors x^n where $x_{\bar{\mathcal{E}}} = \tilde{y}_{\bar{\mathcal{E}}}$ and where $x_{\mathcal{E}}$ is a solution of

$$H_{\mathcal{E}} x_{\mathcal{E}}^T = H_{\bar{\mathcal{E}}} x_{\bar{\mathcal{E}}}^T. \tag{1}$$

Observe that (1) must admit at least two solutions, i.e., $\operatorname{rank} H_{\mathcal{E}} < |\mathcal{E}|$. The solutions can be found by (i) picking any element of $x_{\mathcal{E}}$ and (ii) setting it to an arbitrary value in $\{0, 1\}$, and solving the residual system of equations. Consider a generic solution v^n: a valid codeword pair $\{v^n, w^n\}$ is then obtained by setting $w^n = y^n - v^n$. We refer to the codeword bit selected in step (i) as *pivot*, to the action of selecting a pivot as *pivoting*, and to step (ii) as *guessing*. If the solution is unique, i.e., if $\operatorname{rank} H_{\mathcal{E}} = |\mathcal{E}| - 1$, then decoding stops. In this case, the solution of (1) is attained with a complexity $O(n^3)$. If $\operatorname{rank} H_{\mathcal{E}} = |\mathcal{E}| - s$ with $s > 1$, then there are 2^{s-1} valid codeword pairs. Some observations follow.

1) Any solution of (1) participates in exactly one valid pair.
2) We have that $A^n \notin \mathcal{L}$ if and only if $B^n \notin \mathcal{L}$.

3) The PUPE is given by

$$\begin{aligned}\mathsf{PUPE} &= 1 - \sum_{s=1}^{|\mathcal{E}|} 2^{-(s-1)} \mathsf{P}(|\mathcal{E}| - \operatorname{rank} H_{\mathcal{E}} = s) \\ &\geq \frac{1}{2}\mathsf{DFP}.\end{aligned}$$

4) The number of valid codeword pairs computed by the decoder depends on the transmitted pair $\{a^n, b^n\}$ only through the support set of their difference.

Observation 1 follows by noting that if $v_{\mathcal{E}}$ is a solution of (1), then also $w_{\mathcal{E}} = v_{\mathcal{E}} \oplus 1^{|\mathcal{E}|}$ is a valid solution, and $\{v^n, w^n\}$ is a valid codeword pair. From this, observation 2 follows directly. The expression of the PUPE provided at point 3 stems from the fact that there are 2^{s-1} valid codeword pairs, and from observation 2. It also true that where there exist multiple valid codeword pairs, we have a probability $\geq 1/2$ to pick an erroneous pair. The last observation is a direct consequence of s being the dimension of the linear subspace spanned by the columns of $H_{\mathcal{E}}$, noting that $\mathcal{E} = \operatorname{supp}(a^n \oplus b^n)$.

The last observation yields a simplification in the analysis of the PUPE and of the DFP for linear codes.

Lemma 2 (All-zero codeword)**.** *Consider two users transmitting over the BAC with a (n, k) binary linear block code C, and denote as A^n, B^n the transmitted codewords. Then, the PUPE and the DFP under ML decoding are conditionally independent on A^n.*

Proof. We provide first a sketch of the proof for the PUPE. Denote by E the event $\{A^n \notin \mathcal{L}\}$. Due to observation 2, $\mathsf{PUPE} = \mathsf{P}(E)$. We have that

$$\begin{aligned}\mathsf{P}(E|A^n{=}a^n) &= \sum_{b^n \in C} \mathsf{P}(E|A^n{=}a^n, B^n{=}b^n)\, \mathsf{P}(B^n{=}b^n) \\ &\overset{(a)}{=} \sum_{b^n \in C} \mathsf{P}(E|A^n{=}a^n \oplus a^n, B^n{=}b^n \oplus a^n)\, \mathsf{P}(B^n{=}b^n \oplus a^n) \\ &= \sum_{c^n \in C} \mathsf{P}(E|A^n{=}0^n, B^n{=}c^n)\, \mathsf{P}(B^n{=}c^n) \\ &= \mathsf{P}(E|A^n{=}0^n)\end{aligned}$$

where (a) follows by observation 4. The proof for the DFP, upon re-defining E as the event $\{|\mathcal{E}| - \operatorname{rank} H_{\mathcal{E}} > 1\}$, is obtained by going through the same steps. □

Note that the statement holds is we exchange A^n and B^n. Lemma 2 implies that $\mathsf{PUPE} = \mathsf{P}(E|A^n{=}0^n)$, i.e., we can evaluate the PUPE of a binary linear block code under ML decoding by restricting to the case where one of the two users transmits the all-zero codeword. The same applies to the DFP. The principle can be extended to binary linear block codes under any decoding algorithm whose outcome depends on the transmitted codewords only through the support of their difference. An example is given by erasure peeling decoding of LDPC codes applied to the BAC.

Consider now the case where $A^n = 0^n$. In this case, $\mathcal{E} = \operatorname{supp}(B^n)$ and (1) yields only two solutions if and only if B^n is minimal. Owing to this fact and to Lemma 2, we have that the DFP is given by

$$\mathsf{DFP} = \mathsf{P}(B^n \notin \mathcal{M}(C))\,.$$

As a result, minimal codes achieve a zero DFP (and, hence, a zero PUPE) over a two-user BAC with same-codebook constraint.

Example 1. *Consider the two binary linear block codes C_1 and C_2 with generator matrices*

$$G_1 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \quad \textit{and} \quad G_2 = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}.$$

The first code is a $(3, 2)$ SPC code, and the second code is a $(4, 2)$ code. The minimum distance of both is 2. By listing all codewords, we see that C_1 is a minimal code, whereas C_2 possesses one nonminimal codeword (the all-one vector). It results that C_1 yields a zero DFP (zero PUPE), while for C_2 we have $\mathsf{DFP} = 1/4$ and $\mathsf{PUPE} = 1/8$.

It is important to stress the nature of the erasure channel that stems from the transmission of two codewords of a linear code, over the BAC: By restricting A^n to be the all-zero vector, we see that the erasure pattern at the decoder input erases the support of B^n. We are hence facing the problem of decoding a linear code over an erasure channel that produces erasure patterns that match codewords.

IV. Linear Codes: Maximum Symmetric Rate

The following results show that transmission over the 2-UBAC with linear block codes is fundamentally limited to a symmetric rate not larger than $\mathsf{C}_{\text{LIN}} = 1/2$, in contrast to the maximum symmetric rate $\mathsf{C} = 3/4$ of the 2-UBAC achievable by nonlinear block codes (to be discussed in Section VI).

Theorem 1 (Converse)**.** *Consider transmission over the the 2-UBAC with a (n, k) binary linear block code. We have that*

$$\mathsf{PUPE} \geq \frac{1}{2}\left(1 - \frac{n}{2k-2}\right).$$

Thus, the maximal achievable symmetric rate is $R \leq 1/2$.

Proof. We fix $A^n = 0$. We have that

$$\begin{aligned}\mathsf{PUPE} &\overset{(a)}{\geq} \frac{1}{2}\mathsf{P}(B^n \notin \mathcal{M}(C)) \\ &\overset{(b)}{\geq} \frac{1}{2}\mathsf{P}(w_H(B^n) > n - k + 1) \\ &\overset{(c)}{=} \frac{1}{2}\left[1 - \mathsf{P}(n - w_H(B^n) \geq k - 1)\right] \\ &\overset{(d)}{\geq} \frac{1}{2}\left(1 - \frac{n}{2k-2}\right)\end{aligned}$$

where (a) follows by observation 3) in Section III and by the definition of minimal codewords, (b) is due to Lemma 1, (c) is the result of a simple manipulation, and (d) is due to Markov's inequality and to the fact that the average codeword weight is $n/2$ for any binary linear block code without idle coordinates. By taking the limit $n \to \infty$ of the RHS of (e), we see that the PUPE is bounded away from zero for any $R > 1/2$. □

Note that the result stated in Theorem 1 extends to codes that are $\mathbb{F}_2$-affine. This can be checked by observing that any fixed offset added to the codewords of an (n, k) binary linear block code C does not modify the set of coordinates at which A^n and B^n differ, and the offset can be removed from the sum $A^n + B^n$ at the decoder input.

The following result shows that random linear codes allow operating arbitrarily close to the $1/2$ symmetric rate limit.

Theorem 2 (Achievability)**.** *Consider the random linear code ensemble $\mathcal{C}(n, R_0)$ defined by all parity-check matrices with n columns and $m = n(1-R_0)$ rows, where $R_0 = 1-m/n$ is the nominal ensemble rate, containing all binary linear block codes with blocklength n and rate $R \geq R_0$. The average PUPE of a random code from $\mathcal{C}(n, R_0)$ satisfies*

$$\mathsf{E}\left[\mathsf{PUPE}(C)\right] < \sum_{d=1}^{n} \binom{n}{d} 2^{-n-[n(1-R_0)-d+1]^+}. \qquad (2)$$

Moreover, the by denoting $\delta = d/n$, a lower bound on the error exponent of rate-R random linear block codes is given by

$$E(R) = \inf_{0<\delta\leq 1} \left(1 - \mathsf{H}_\mathrm{b}(\delta) + \left[(1-R)-\delta\right]^+\right) \qquad (3)$$

which is strictly positive for any $R < 1/2$.

The proof of Theorem 2 requires first to prove (2). The result is achieved by applying standard results on the rank properties of random matrices with independent coefficients that are uniformly distributed in $\mathbb{F}_2$. By analyzing the $\lim_{n\to\infty} \frac{1}{n} \log_2 \mathsf{E}\left[\mathsf{PUPE}(C)\right]$, the error exponent can be lower bounded through (3). An additional result on the properties of the random linear code ensemble $\mathcal{C}(n, R_0)$ is given in [17, Corr. 2.5], which we recall next.

Theorem 3 (Growth rate of the number of minimal codewords, [17])**.** *Consider the random linear code ensemble $\mathcal{C}(n, R_0)$ defined by all parity-check matrices with n columns and $m = n(1-R_0)$ rows, where $R_0 = 1-m/n$ is the nominal ensemble rate, containing all binary linear block codes with blocklength n and rate $R \geq R_0$. We have that the growth rate of the expected number of minimal codewords is*

$$\lim_{n\to\infty} \frac{1}{n} \log_2 \mathsf{E}\left[|\mathcal{M}(C)|\right] = \begin{cases} \mathsf{H}_\mathrm{b}(1-R_0) - (1-R_0), & R_0 > \frac{1}{2} \\ R_0, & R_0 \leq \frac{1}{2}. \end{cases}$$

In particular for $R_0 > 1/2$ the growth rate of the expected number of minimal codewords is lower than R_0, meaning that $\mathsf{PUPE} \to 1$, strengthening Theorem 2.

V. Asymptotic Analysis of LDPC Code Ensembles

For the analysis of LDPC codes over the 2-UBAC, we analyze two decoding strategies, namely ML decoding and peeling decoding. The former has $O(n^3)$ complexity, although efficient decoders that exploit the sparse nature of the system of equations in (1) do exist (see e.g. [18]). As we will see, while under ML decoding the choice of the pivot is irrelevant with respect to a successful outcome of the decoding process, under peeling decoding the choice of the pivot plays a crucial role, i.e., the decoding may succeed or fail depending on the selected pivot. In the following, we will provide only the statement of the main theorems with a brief outlook to the proofing technique.

A. Maximum-likelihood Decoding: A Negative Result

We discuss next a negative result that holds for any unstructured LDPC code ensemble, under ML decoding. Specifically, the following theorem shows that certain LDPC code ensembles are not suitable for communication over the 2-UBAC.

Theorem 4. *Consider an LDPC code ensemble $\mathcal{C}^n_{\Lambda,\mathrm{P}}$. If CNs admit only degrees 2 and 3, then the ensemble average ML decoding PUPE cannot be made arbitrarily small as $n \to \infty$.*

The result stated in Theorem 4 can be proved by focusing on codewords with weight linear in n. The associated induced graph possesses only CNs of degree zero or two. We have that (1) admits two solutions if and only if the residual graph is a cycle touching all vertexes (VNs). This implies that all VNs must have degree two. It can be shown that the probability that a random 2-regular graph is unicyclic goes to zero polynomially fast in the order of the graph.

B. Peeling Decoding and Pivoting

We assume in the following that the reader is familiar with erasure peeling decoding [19] [20, Chapter 3] of LDPC codes. We distinguish two pivoting strategies:

1) *Random pivoting.* A pivot is picked uniformly at random within $\mathcal{E}$. In this case, the decoding complexity is $O(n)$;
2) *Optimum pivoting.* The decoder attempts decoding by trying one-by-one all pivots in $\mathcal{E}$. Pivoting stops when a pivot that yields a solution of all erasures is found, or all pivots have been tried. In this case, the decoding complexity is $O(n^2)$.

We refer to pivots whose guessing allows the recovery of all erasures as *good pivots*, whereas pivots that do not yield a success are called *bad pivots*. Among bad pivots, there might exist pivots whose guessing does not allow solving any of the remaining erasures. We refer to such pivots as *locked pivots*. The existence of good and bad pivots is illustrated through the following example.

Example 2. *Figure 3 depicts a Tanner graph representing the constraints imposed on the transmitted codewords by a $(5, 2)$ LDPC code. Additional nodes are included, which represent the channel observations. Figure 4 illustrates the case where $y^n = (2, 1, 1, 1, 1)$. Decoding proceeds by operating over a single-user graph. After observing the channel output, four erasures remain to be solved. The first step consists in selecting a pivot. Note that the peeling decoder is doomed to fail if we choose x_2 as pivot. On the contrary, by selecting any pivot among x_3, x_4 and x_5 allows recovering all the erasures.*

The following lemma allows simplifying the analysis of peeling decoding.

Lemma 3 (All-zero codeword)**.** *Consider two users transmitting over the BAC with a (n, k) LDPC code C, and denote as*

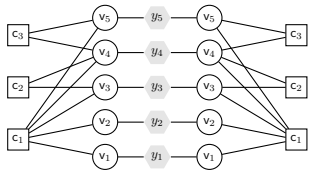

Fig. 3. Example of transmission over the 2-UBAC with a length-5 LDPC code.

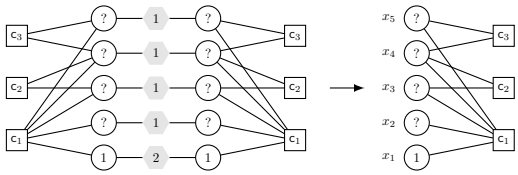

Fig. 4. Decoding by turning the problem into an erasure decoding problem.

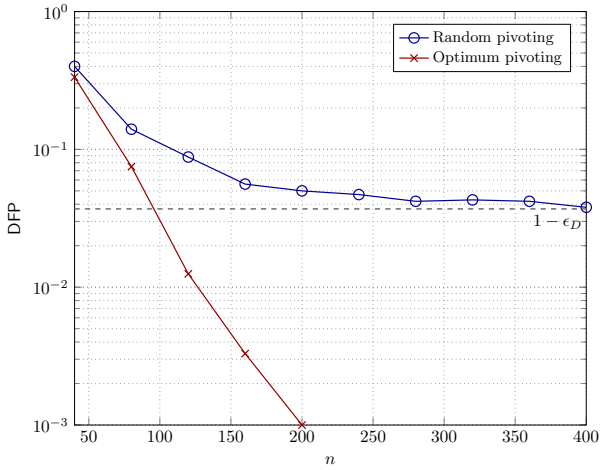

Fig. 5. DFP vs. blocklength n for $(3,4)$ regular LDPC codes under random and optimum pivoting.

A^n, B^n the transmitted codewords. Then, the PUPE and the DFP under peeling decoding are conditionally independent on A^n.

Lemma 3 extends the Lemma 2 to peeling decoding of LDPC codes under random and optimal pivoting. We omit the proof, since it follows directly by the fact that under either pivoting approach, the outcome of decoding depends only on the erasure set $\mathcal{E}$. Lemma 3 allows assuming the transmission of the all-zero codeword by one of the two users. Assume next x_i, $i \in \mathcal{E}$, to be the chosen pivot. Peeling decoding succeeds if and only if the set of VNs with indexes in $\mathcal{E} \setminus \{i\}$ does not contain stopping sets. If $a^n = 0^n$, this is equivalent to require that $\mathrm{supp}(b^n) \setminus \{i\}$ does not contain stopping sets.

We discuss first a negative result holding for any (regular or irregular) unstructured LDPC code ensemble, under peeling decoding.

Theorem 5. *Consider transmission with a random code C from an LDPC code ensemble $\mathcal{C}^n_{\Lambda,\mathrm{P}}$. Assume furthermore $A^n = 0^n$. Denote by σ_n the expected fraction of locked pivots, where the expectation is over the code C and the choice of the codeword B^n. If the maximum check node degree is larger than 3, then*

$$\lim_{n\to\infty} \sigma_n = \sigma > 0.$$

The proof of Theorem 5 requires the introduction of the notion of *dominant graph type* for a (Λ, P) LDPC code ensemble. The dominant graph type $(\omega^\star, \mathrm{L}^\star, \mathrm{R}^\star)$ is the triplet for which $\mathsf{G}^{\mathrm{IG}}(\omega, \mathrm{L}, \mathrm{R})$ attains its global maximum. Qualitatively speaking, by picking a random code C in $\mathcal{C}^n_{\Lambda,\mathrm{P}}$ and by selecting a random codeword B^n in C, we have that $\mathcal{G}(B^n)$ will possess a number of VNs close to $w^\star = \omega^\star n$ and a degree distributions close to $(\mathrm{L}^\star, \mathrm{R}^\star)$, with high probability, as n grows large. Assume now that $\mathrm{R}^\star$ admits a positive fraction of CNs with degree larger than two. It is then sufficient to prove that there is a constant expected fraction of VNs in $\mathcal{G}(B^n)$ which are connected only to CNs with degree larger than two in $\mathcal{G}(B^n)$. In fact, by selecting a VN at random in $\mathcal{G}(B^n)$, and denoting its degree as d, there is a positive probability of having all d edges connected to the subset of CNs with degree larger than two. An obvious consequence of Theorem 4 and of Theorem 5 is provided by the following lemma.

Corollary 1. *Under peeling decoding and random pivoting, the average of the DFP of any LDPC code ensemble is non-vanishing as $n \to \infty$.*

In fact, the result stated by Theorem 4 is inherited by any decoder, including peeling decoding under random pivoting. Moreover, Theorem 5 implies that there is fixed probability of selecting as pivot a locked pivot. Figure 5 provides empirical evidence of the result stated by Corollary 1. In particular, the DFP under random pivoting and under optimum pivoting is provided for codes from the $(3,4)$ regular ensemble, and various values of the blocklength n. The result are obtained by Monte Carlo simulations, where for each blocklength a code has been drawn randomly from the ensemble. The DFP appears to approach a floor at $\approx 3.7 \times 10^{-2}$ as n grows.

Theorem 6. *Consider transmission with a random code C from an LDPC code ensemble $\mathcal{C}^n_{\Lambda,\mathrm{P}}$. Assume furthermore $A^n = 0^n$. Denote by $\gamma_n(\mathtt{l},\mathtt{r})$ the expected fraction of good pivots, where the expectation is over the code C and the choice of the codeword B^n, and by*

$$\gamma(\mathtt{l},\mathtt{r}) = \lim_{n\to\infty} \gamma_n(\mathtt{l},\mathtt{r}).$$

As $n \to \infty$ the expected fractions of good pivots for the $(3,4)$, $(4,5)$ and $(5,6)$ regular LDPC code ensembles are lower bounded respectively by $\gamma(3,4) \geq 0.963$, $\gamma(4,5) \geq 0.770$ and $\gamma(5,6) \geq 0.618$.

The proof of Theorem 6 involves several steps. First, given an $(\mathtt{l},\mathtt{r})$ regular LDPC code ensemble, one should determine the dominant induced graph type $(\omega^\star, \mathrm{L}^\star, \mathrm{R}^\star)$. We know that for n large the graph induced by a random codeword B^n in a random code $C \in \mathcal{C}^n_{\mathtt{l},\mathtt{r}}$ will possess about $w^\star = \omega^\star n$ VNs, where for the ensembles under analysis $\omega^\star = 1/2$. Moreover, $\mathrm{L}^\star_d = 1$ for $d = \mathtt{l}$ and $\mathrm{L}^\star_d = 0$ for $d \neq \mathtt{l}$ (since the code is regular). It is possible to show that if $\omega^\star = 1/2$, then $\mathrm{R}^\star_d = 0$

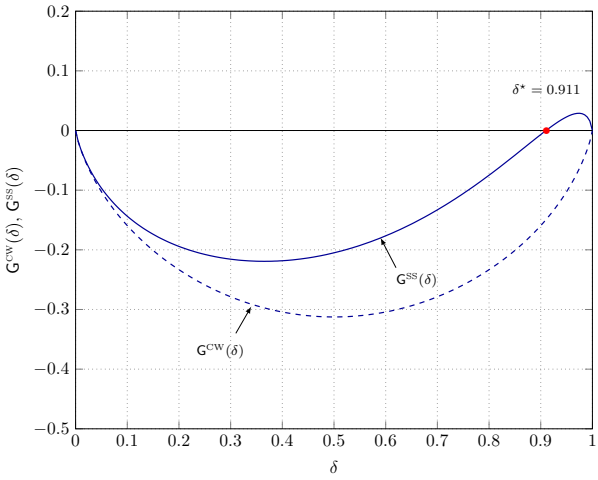

Fig. 6. Growth rate of the codeword weight distribution and growth rate of the stopping set weight distribution for the ensemble defined by the degree distribution pair $(\mathrm{L}^\star, \mathrm{R}^\star)$, being $(\mathrm{L}^\star, \mathrm{R}^\star)$ the degree distribution of the dominant induced graph type of the $(3, 4)$ regular LDPC code ensemble. The fact that $\mathsf{G}^{\mathrm{CW}} < 0$ simply means that for this ensemble the typical codeword is minimal with high probability.

for d odd while for d even we have $\mathrm{R}^\star_d = K\binom{\mathtt{r}}{d}$ where K is a normalization constant. It follows that the residual graph $\mathcal{R}(B^n)$ will have about $w^\star = \omega^\star n$ vertexes and $\mathrm{R}^\star_2 m$ edges, with $m = n\mathtt{l}/\mathtt{r}$. Furthermore, the degree distribution of the residual graph will be close to a binomial distribution with parameters $(\mathtt{l}, p)$ where $p = 4\mathrm{R}^\star_2/\mathtt{r}$. By analyzing the degree distribution of the residual graph, it is possible to verify that the condition for the emergence of a giant component [21] is met by the $(3, 4)$, $(4, 5)$ and $(5, 6)$ regular LDPC code ensembles (these are the only regular LDPC code ensembles satisfying the condition). Moreover, the size of the giant component can be determined via [22, Theorem 1]. Following [22], let us denote by ϵ_D the fraction of the residual graph vertexes belonging to the giant component (i.e., the fraction of VNs in the induced graph connected with degree-2 CNs). We refer to ϵ_D as the *normalized giant component size*. It is evident that by guessing the value of a codeword bit associated to a VN (i.e., a vertex) belonging to the giant component, we expect to resolve at least $w^\star \epsilon_D$ erasures. What is left to be addressed, is what happens to the remaining $w^\star(1 - \epsilon_D)$ erasures. To do so, we consider an irregular LDPC code ensemble with blocklength $w^\star$ and degree distribution pair $(\mathrm{L}^\star, \mathrm{R}^\star)$. For such ensemble, we derive the growth rate of the stopping set weight distribution

$$\mathsf{G}^{\mathrm{SS}}(\delta) = \lim_{w^\star \to \infty} \frac{1}{w^\star} \log_2 \bar{A}^{\mathrm{SS}}_{\delta w^\star}.$$

Denote by $\delta^\star$ critical exponent stopping ratio, i.e., the first positive zero crossing of $\mathsf{G}^{\mathrm{SS}}(\delta)$. Owing to $\mathtt{l} > 2$, we have that the probability of having stopping sets of size $\leq \delta^\star w^\star$ goes to zero polynomially fast in n [23]. (This can also be shown by noticing tha the full Tanner graph is an expander, so that stopping sets $\leq \delta w^\star$ do not exist, cf. [20, Section 8.4].) It follows that, if

$$1 - \epsilon_D < \delta^\star \tag{4}$$

decoding will proceed by solving all erasures with high probability. We have that for the $(3, 4)$ regular ensemble $\epsilon_D = 0.963$ and $\delta^\star = 0.911$, for the $(4, 5)$ regular ensemble $\epsilon_D = 0.854$ and $\delta^\star = 0.770$, and for the $(5, 6)$ regular ensemble $\epsilon_D = 0.549$ and $\delta^\star = 0.618$. In all three cases the condition (4) is met. For illustration, the growth rate of the stopping set weight distribution (as well as the growth rate of the codeword weight distribution) for the ensemble with degree distribution pair $(\mathrm{L}^\star, \mathrm{R}^\star)$, being $(\mathrm{L}^\star, \mathrm{R}^\star)$ the degree distribution of the dominant induced graph type of the $(3, 4)$ regular LDPC code ensemble, is provided in Figure 6. Remarkably, by inspection of Figure 5, we see that under random pivoting the DFP floors approaching for large n the upper bound $1 - \epsilon_D$ on the fraction of bad pivots. A similar behavior have been observed for the $(4, 5)$ and $(5, 6)$ regular LDPC code ensembles. This fact is striking, since it seems to suggest that the analysis of the residual graph degree distribution yields a tight estimate of the true expected fraction of bad pivots (at least for regular LDPC code ensembles).

We remark that, although our method fails to give non-zero lower bound for any other regular ensemble, many open problems remain. First, it could be that the fraction of good pivots converges to zero, but their number stays positive nevertheless. Second, our method (that essentially only tracks the number of degree-2 check nodes in the residual graph but does not account for the possibility of creating new degree-2 check nodes as the peeling progresses) may give a sub-optimal bound. Finally, we have not attempted to extend our method to irregular ensembles.

VI. Remarks on Nonlinear Codes

First, we recall that a standard observation dating back to Lindström shows that using a nonlinear code (with codewords taken to be columns of the parity check matrix of a 2-error correcting BCH code) achieves rate $1/2$ and *exactly zero PUPE* over 2-UBAC, cf. [24, Section II.A], which also shows that decoder complexity is $O(n^3)$ or smaller (depending on the way $\mathbb{F}_{2^k}$-multiplication is implemented). It is widely believed that under zero-error PUPE, this rate bound is not improvable, with the currently best upper bound equal to 0.5753 in [25].

We have seen in Theorems 1 and 2 that the random linear code has vanishing (but non-zero) PUPE up to the critical rate of $\mathsf{C}_{\mathrm{LIN}} = 1/2$. Next, we observe yet another way rate-$1/2$ is special.

Theorem 7 (TIN capacity)**.** *Consider a code C and the following simple TIN decoder (for treat interference as noise). It prepares a list of all codewords $c^n \in C$ which are compatible with the received vector (c^n is compatible with y^n if for all i with $y_i \neq 1$ we have $c_i = y_i/2$). If the list is of size greater than two then error is declared, otherwise the list becomes a decoded output. If C is a random nonlinear code of rate R with codewords uniform on $\{0, 1\}^n$, then*

$$\lim_{n\to\infty} \mathsf{E}\left[\mathsf{PUPE}(C)\right] = \begin{cases} 0, & R < 1/2, \\ 1, & R > 1/2 \end{cases}.$$

We mention that the coincidence of $\mathsf{C}_{\text{LIN}} = \mathsf{C}_{\text{TIN}} < \mathsf{C}$ remains true for a general 2-user A-channel, namely we have:

$$\mathsf{C}_{\text{LIN}} = \mathsf{C}_{\text{TIN}} = \log_2 q - \frac{q-1}{q} < \mathsf{C} = \log_2 q - \frac{q-1}{2q}. \quad (5)$$

Next, we ask whether there exist *any* polynomial time/space coding scheme for attaining the capacity ($\mathsf{C} = 3/4$) of the 2-UBAC. The answer is positive. First, a simple random coding argument shows that for any $R < 3/4$ and for any integer L there exists a collection $C_1, \dots, C_L$ of linear codebooks such that any pair of them C_i, C_j ($i \neq j$) has a vanishing error when used on a (sourced!) BAC. Furthermore, the decoder has $O(n^3)$ complexity by running the Gaussian elimination.

Next, we use this collection of codebooks to build a 2-UBAC code by considering a two-phase scheme. In the first phase each of the users selects a random integer from $\{1, \dots, L\}$. These two messages are encoded via a zero-error 2-UBAC code in $O(\log_2 L)$ channel uses. Next, each user transmits its data via a respective codebook C_i. Note that asymptotically we have

$$\lim_{n\to\infty} \mathsf{E}\left[\mathsf{PUPE}(C)\right] = \frac{1}{L},$$

since errors happen only when both users select the same codebook. Thus by taking L large we can achieve arbitrary low error and rate $R \to 3/4$. The disadvantage of this scheme is that even if C_i's are linear codes[1] this requires about $\frac{n^2}{\mathsf{PUPE}}$ space. So while technically it is polynomial time/space algorithm, the dependence on PUPE is prohibitive.

Open question: Does there exist a coding scheme for 2-UBAC achieving $R > 1/2$ with time-space complexity $\mathrm{poly}(n, \log \mathsf{PUPE})$?

We mention one possible idea for achieving this. Notice that a simple search shows that there exists an $[5] \to \{0,1\}^3$ code for 2-UBAC with zero PUPE. Thus, using this code as an inner code we can convert every 3 uses of BAC into a single channel use of A-channel with $q = 5$. Then together with (5) we notice that concatenating our inner code with either TIN code or a random $\mathbb{F}_5$-linear code results in rate ≈ 0.507 and vanishing PUPE. This may appear to have solved the open question because (one guesses) the linear code can be decoded by some version of the Gaussian elimination. Unfortunately, this latter statement is wrong. In fact over the A-channel with $q > 2$ decoding a linear code is NP hard[2].

We conclude by mentioning two other interesting *open directions*. First, it is not clear how to implement LDPC codes for the 2-user A-channel with $q > 2$ since the peeling is no longer possible. Second, it is interesting whether it is possible to achieve $R = 1/2$ via iterative decoding. We conjecture that except for the three examples we have found all other *regular* LDPC code ensembles do not yield vanishing PUPE. However, it is possible that there exist *irregular* LDPC code ensembles (or other sparse-graph codes) attaining $R \to 1/2$ and $\mathsf{PUPE} \to 0$ as $n \to \infty$.

[1]Interestingly, it can be shown that taking C_i's to be random cosets of a fixed linear code C_0 does not work: the rate bottlenecks at $1/2$ again.

[2]Yuzhou Gu has communicated to us a reduction from q-NAE-SAT.

REFERENCES

[1] Y. Polyanskiy, "A perspective on massive random-access," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Jun. 2017.

[2] E. Casini, R. De Gaudenzi, and O. Del Rio Herrero, "Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Schemefor Satellite Access Packet Networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1408–1419, Apr. 2007.

[3] C. Stefanovic, P. Popovski, and D. Vukobratovic, "Frameless ALOHA protocol for wireless networks," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2087–2090, Dec. 2012.

[4] E. Paolini, G. Liva, and M. Chiani, "Coded Slotted ALOHA: A Graph-Based Method for Uncoordinated Multiple Access," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6815–6832, Dec. 2015.

[5] E. Sandgren, A. Graell i Amat, and F. Brännström, "On Frame Asynchronous Coded Slotted ALOHA: Asymptotic, Finite Length, and Delay Analysis," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 691–704, Feb. 2017.

[6] F. Clazzer, C. Kissling, and M. Marchese, "Enhancing Contention Resolution ALOHA Using Combining Techniques," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2576–2587, Jun. 2018.

[7] R. Calderbank and A. Thompson, "CHIRRUP: a practical algorithm for unsourced multiple access," *Information and Inference: A Journal of the IMA*, vol. 9, no. 4, pp. 875–897, 12 2019.

[8] S. S. Kowshik, K. Andreev, A. Frolov, and Y. Polyanskiy, "Energy efficient coded random access for the wireless uplink," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4694–4708, Aug. 2020.

[9] V. K. Amalladinne, J.-F. Chamberland, and K. R. Narayanan, "A coded compressed sensing scheme for unsourced multiple access," *IEEE Trans. Inf. Theory*, vol. 66, no. 10, pp. 6509–6533, Oct. 2020.

[10] A. Fengler, P. Jung, and G. Caire, "Sparcs for unsourced random access," *IEEE Trans. Inf. Theory*, vol. 67, no. 10, pp. 6894–6915, Oct. 2021.

[11] Sigfox, "SIGFOX: The Global Communications Service Provider for the Internet of Things," www.sigfox.com.

[12] LoRa Alliance, "The LoRa Alliance Wide Area Networks for Internet of Things," www.lora-alliance.org.

[13] N. Abramson, "The ALOHA System - Another Alternative for Computer Communications," in *Proc. 1970 Fall Joint Computer Conference*. AFIPS Press, Nov. 1970.

[14] S.-C. Chang and J. Wolf, "On the t-user m-frequency noiseless multiple-access channel with and without intensity information," *IEEE Trans. Inf. Theory*, vol. 27, no. 1, pp. 41–48, Jan. 1981.

[15] T.-Y. Hwang, "Decoding linear block codes for minimizing word error rate," *IEEE Trans. Inf. Theory*, vol. 25, no. 6, pp. 733–737, Nov. 1979.

[16] J. L. Massey, "Minimal codewords and secret sharing," in *Proc. 6th joint Swedish-Russian International Workshop on Information Theory*, 1993, pp. 276–279.

[17] A. Ashikhmin and A. Barg, "Minimal vectors in linear codes," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 2010–2017, May 1998.

[18] T. Richardson and R. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 638–656, Feb. 2001.

[19] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Efficient erasure correcting codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 569–584, Feb. 2001.

[20] T. Richardson and R. Urbanke, *Modern coding theory*. Cambridge University Press, 2008.

[21] M. Molloy and B. Reed, "A critical point for random graphs with a given degree sequence," *Random Structures and Algorithms*, vol. 6, pp. 161 – 180, 1995.

[22] ——, "The size of the giant component of a random graph with a given degree sequence," *Combinatorics, Probability and Computing*, vol. 7, pp. 295–305, 1998.

[23] A. Orlitsky, K. Viswanathan, and J. Zhang, "Stopping set distribution of LDPC code ensembles," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 929–953, Mar. 2005.

[24] O. Ordentlich and Y. Polyanskiy, "Low complexity schemes for the random access gaussian channel," in *IEEE International Symposium on Information Theory (ISIT)*, Jun. 2017.

[25] G. Cohen, S. Litsyn, and G. Zémor, "Binary b2-sequences: a new upper bound," *Journal of Combinatorial Theory, Series A*, vol. 94, no. 1, pp. 152–155, 2001.