On the Advantages of Asynchrony in the Unsourced MAC

Alexander Fengler, Alejandro Lancho, Krishna Narayanan, and Yury Polyanskiy

***Abstract*—In this work we demonstrate how a lack of synchronization can in fact be advantageous in the problem of random access. Specifically, we consider a multiple-access problem over a frame-asynchronous 2-user binary-input adder channel in the unsourced setup (2-UBAC). Previous work has shown that under perfect synchronization the per-user rates achievable with linear codes over the 2-UBAC are limited by 0.5 bit per channel use (compared to the capacity of 0.75). In this paper, we first demonstrate that arbitrary small (even single-bit) shift between the user's frames enables (random) linear codes to attain full capacity of 0.75 bit/user. Furthermore, we derive density evolution equations for irregular LDPC codes, and prove (via concentration arguments) that they correctly track the asymptotic bit-error rate of a BP decoder. Optimizing the degree distributions we construct LDPC codes achieving per-user rates of 0.73 bit per channel use.**

***Index Terms*—Multiple-Access, Low-density parity check (LDPC), Unsourced, massive machine-type communication**

I. Introduction

A recent line of work, termed unsourced random access (URA or UMAC), exploits the idea of same-codebook communication [2]. This approach allows to separate the different messages in a multiple-access channel (MAC) based purely on the structure of the codebook, i.e., the set of allowed messages. It was shown that good unsourced code designs can approach the capacity of the additive white Gaussian noise (AWGN) adder channel without the need for coordination [2], [3]. While many unsourced code constructions have been proposed [3]–[9], most of them lack analytic understanding and it is not well understood what properties make a good unsourced codebook. Furthermore, many proposed schemes have a high decoding complexity. Recent works [10], [11] have constructed LDPC codes specifically for two-user communication on the unsourced binary input adder channel (UBAC). It was found that linear codes in general suffer a rate loss in the UBAC and cannot achieve sum rates higher than 1 bit/channel use, which is still far from the sum-rate capacity of 1.5 bits/channel use.

Another concern for the practical applicability of unsourced codes is the assumption of perfect synchronization, present in many works. In low-power low-cost transmitters perfect synchronization is hard to achieve. Classic results [12] show that frame-asynchrony does not change the capacity of a discrete MAC, as long as the allowed delay is smaller than the blocklength. Recent solutions for uncoordinated multiple-access schemes that can deal with asynchronism were proposed in [13], [14]. Both of these works present schemes specifically for orthogonal frequency-division multiplexing (OFDM) modulation with timing offsets within the cyclic prefix. Such timing offsets can be efficiently handled in the frequency domain. Nonetheless, OFDM is not necessarily the best choice for the mMTC scenario since it requires a high level of frequency synchronization, which is hard to achieve with low-cost transmitters.

In this work, we first show that random linear codes achieve the BAC capacity of 1.5 bits/ch. use as soon as a frame delay of at least one symbol is introduced. As such, it enables same-codebook communication with linear codes and linear decoding complexity that does not suffer from the rate 1 bottleneck, which limits unsourced linear codes in the frame-synchronous case. Although the channel model is idealistic, it is also quite general and does not rely on any specific modulation method. Further, we design LDPC codes with linear decoding complexity for the two-user frame-asynchronous UBAC. We find codes that achieve sum-rates of 1.46 bits/ch. use. The decoding can be done by two copies of a conventional single-user belief propagation (BP) decoder that periodically exchange information. We also show that our design works if the delay is a random integer with a maximum value that scales at most sub-linearly with the blocklength.

Randomized LDPC code designs for the two-user multiple-access channel with AWGN have been presented in [15], [16]. For the code construction presented in [16] it is crucial that the two code ensembles are optimized independently, resulting in two different ensembles. If one check node (CN) distribution is fixed, the CN distribution of the other user can be optimized by a linear program. In [15], one common code ensemble is designed, but the two users pick a different random code from the same ensemble. In addition, to obtain a linear optimization

A. Fengler, A. Lancho, and Y. Polyanskiy are with the Massachusetts Institute of Technology. (Email: {fengler,lancho,yp}@mit.edu)

Research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. Alejandro Lancho has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 101024432. Alexander Fengler was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Grant 471512611. This work is also supported by the National Science Foundation under Grant No CCF-2131115.

K. Narayanan is with the Texas A&M University. This work is supported by NSF grant CCF-2131106. (Email: krn@tamu.edu)

An extended version of this paper is available online [1].

program, the codes in [15] are constrained such that variable nodes (VNs) that are connected through the MAC have the same degree. Such a constraint would be hard to enforce in a model with random delay. In contrast, in this work we design one LDPC ensemble from which one code is chosen at random and used by both users. The design of the ensemble relies on alternating optimization of CN and VN degree distributions. Surprisingly, we find that degree one VNs do not result in error floors, in contrast to LDPC codes for the single-user binary-erasure channel (BEC). A particular difficulty in proving the density evolution (DE) in the joint graph is that the channel transition probabilities for one user depends on the transmitted codeword of the other user. Since the codewords come from the same codebook the channel outputs may be correlated. To that end we employ the symmetrization technique of coset ensembles, cf. [17], although an additional subtlety in our case is that we need to show that both users can use the same coset. Thus, our design strictly adheres to the unsourced paradigm where both users use a common codebook. The symmetrization allows us to prove that DE describes the asymptotic bit-error rate (BER) and, furthermore, that it is independent of the transmitted codewords. This implies that we can assume that both users transmit the all-zero codeword plus a dither when analyzing the error probability. We provide a full proof that the asymptotic error probability is described by the DE and give an analysis of the probability of short-length stopping sets, which result in an error floor. The error floor analysis shows that we can expurgate short-length stopping sets created by the MAC nodes as long as the fraction of degree one VNs is below a certain threshold. Numerical simulations confirm that DE accurately predicts the error probability for large blocklengths. We use the DE to construct codes that approach the capacity of the two-user BAC. Our work shows that frame-asynchrony can be exploited to design efficient linear unsourced codes.

To summarize, our main intellectual contributions in this paper are:

- A random coding argument that shows that linear codes can achieve the full BAC capacity with a single symbol delay.
- The derivation of the DE equations under the same-codebook constraint and sub-linear frame delays.
- A rigorous proof that the BER of a random code from the ensemble will concentrate around the DE.
- The design of a codebook that enables two-user communication at rates close to the Shannon limit.

These findings imply that a non-zero frame delay enables two users to use the same LDPC encoder while still achieving rates close to the two-user BAC capacity. In addition, decoding can be done with linear complexity and a simplified decoder architecture that consists of two connected copies of the same single-user BP decoder.

II. Channel Model

We study the frame-asynchronous noiseless BAC:

$$y_i = c_{1,i} + c_{2,i-\tau} \tag{1}$$

where $\tau \in [0 : \tau_{\max}]$ and $c_{u,i} \in \{1, -1\}$ for $u \in \{1, 2\}, i \in [1 : n]$ and $c_{u,i} = 0$ for $i < 1$ or $i > n$. More specifically, each user transmits a binary-phase-shift keying (BPSK) modulated version of a binary codeword $\mathbf{c}_u = 2\mathbf{m}_u - 1, \mathbf{m}_u \in \{0, 1\}^n$. We will analyze the case where τ is random and uniformly distributed. Furthermore, we will study the asymptotic behavior of code constructions when $\tau_{\max} \in o(n)$, i.e., $\tau_{\max}/n \to 0$ as $n \to \infty$. This setting is also known as mild asynchrony in information theory [18]. Both users transmit a uniform i.i.d. sequence of nR bits, $\mathbf{b}_1, \mathbf{b}_2$, by picking the respective binary codewords $\mathbf{m}_1, \mathbf{m}_2$ independently, uniformly at random from a common codebook over the binary field $\mathcal{C} \in \mathbb{F}_2^{n \times 2^{nR}}$, where n denotes the blocklength and $0 < R < 1$ the per-user rate. The decoder outputs a list of two messages $g(\mathbf{y})$ and the per-user error probability is defined as $P_e = \frac{1}{2}(\mathbb{P}(\mathbf{b}_1 \notin g(\mathbf{y})) + \mathbb{P}(\mathbf{b}_2 \notin g(\mathbf{y})))$.

Since the model includes no noise, the channel model reduces to an erasure channel where a received symbol can be considered as erased if $(c_{1,i}, c_{2,i-\tau}) \in \{(+1, -1), (-1, +1)\}$.

Remark 1: The coding construction in this paper also works for the synchronous model if users employ a randomly chosen *cyclic* shift of their codeword before transmission. However, in this case some mechanism needs to be added that allows to recover the shift of each user, e.g., adding a preamble to each codeword. For the model (1) this is not necessary since τ can be found easily from amplitude information in $\mathbf{y}$.

Remark 2: The BAC model can also be used to model on-off keying modulation. In that case there is some ambiguity since there is no dedicated idle symbol. Nonetheless, it is still possible to detect the start of a frame by introducing a preamble.

III. Random Linear Codes

We next show that a frame delay of just one symbol is enough for random linear codes to achieve the two-user BAC capacity

Theorem 1: There exist linear (n, k) codes for the two-user frame-asynchronous UBAC with $\tau = 1$ and

$$P_e \leq \frac{n-1}{2} 2^{n(2R-1.5)} + o_n(1). \tag{2}$$

□

Proof: The proof is given in [1]. ■

Theorem 1 shows that random linear codes can achieve a vanishing error probability if $R < 0.75 - \delta$ for any $\delta > 0$. It can be shown for both parity check and generator ensembles. We briefly describe the intuition behind the proof for parity check ensembles and why $\tau > 0$ is strictly necessary to get rates larger than 0.5. The idea is to treat the channel as erasure channel, as described in Section II. The erased symbols can, in principle, be recovered by solving the parity check equations $\mathbf{H}\mathbf{m}_1 = \mathbf{0}$ and $\mathbf{H}\mathbf{m}_2 = \mathbf{0}$. A key property of the BAC is that on the erased set the codewords from the two user have opposed bits, i.e. $c_{1,i} = -c_{2,i-\tau}$. This gives a second collection of parity equations for each codeword. For $\tau = 0$ the additional parity check equations would be linearly

dependent, and provide no new information. In that case, since the size of the erased set is around $n/2$, the parity check matrix needs to have $n/2+\delta$ linearly independent rows for correct recovery, resulting in $R < 1/2$. In contrast, for $\tau = 1$ we show that the collection of parity check equations arising from $c_{1,i} = -c_{2,i-\tau}$ for $i \in \mathcal{E}$ is linearly independent from the set of equations given by $\mathbf{Hm}_1 = \mathbf{Hm}_2 = \mathbf{0}$ with high probability. Therefore $n/4+\delta$ linearly independent equations for each user, resulting in a total of $n/2+2\delta$ linearly independent equations for each codeword, will be enough to ensure correct decoding, allowing for $R < 3/4$. In the following we will construct LDPC codes that approach this limit with linear decoding complexity.

IV. LDPC Code Design

A. LDPC Code Ensembles

LDPC codes are defined by a bipartite graph where the transmitted bits are represented by VNs which are subject to local parity checks, represented by CNs. We study random codes that are drawn uniformly at random from a given ensemble, defined by the degree distribution of VNs and CNs. Specifically, a random graph code from the ensemble is created by first assigning degrees to VN and CNs proportional to some degree distributions. Then the emanating stubs (half-edges) of VNs and CNs are connected through a uniform random permutation (multi-edges are not explicitly forbidden). Finally the VNs are also permuted uniformly at random. We would like to emphasize that it is important for our construction that the ensemble definition includes a random permutation of the VNs. For memoryless single-user channels this is usually not necessary since the error probability is invariant under permutation of VNs, and some works do not mention it for this reason, e.g., [19]. However, in the multiple-access case correlations between VN degrees of neighboring nodes may introduce unwanted correlations in the joint graph.

Let L_i denote the fraction of nodes with degree i, λ_i the fraction of edges that connect to degree i VNs, and ρ_i the fraction of edges that connect to degree i CNs. We also define the corresponding power series $L(x) := \sum L_i x^i, \lambda(x) := \sum \lambda_i x^{i-1}$, and $\rho(x) := \sum \rho_i x^{i-1}$, and we denote the corresponding ensemble as LDPC(λ, ρ).

B. Message Passing Decoding

We study the bit-error probability under BP decoding on the joint graph.The values of VNs $(v_{1,i}, v_{2,i})$ are initialized with their know values if $y_i \neq 0$ and are initialized with the erased symbol ϵ if $y_i = 0$. BP decoding on the joint graph can be realized by running two conventional single-user BP decoders on $(y_1, ..., y_n)$ and $(y_{1+\tau}, ..., y_{n+\tau})$ respectively and exchanging information between them on $(y_{1+\tau}, ..., y_n)$. The information exchange is particularly simple for the BAC since $c_{1,i}$ fully defines $c_{2,i-\tau}$ given y_i. We denote the function nodes that enforce the channel constraint (1) as *MAC nodes*. An example of a joint graph is depicted in Fig. 1 where triangles depict MAC nodes, squares are CNs, and circles are VNs. The single-user decoder can be run for multiple iterations

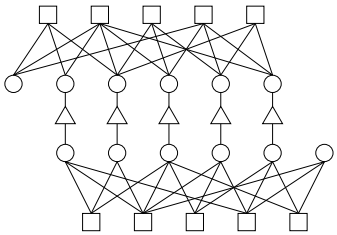

Fig. 1: Factor Graph for a UBAC with $\tau = 1$. Triangles denote MAC nodes, squares are CNs, circles are VNs.

before information exchange. Nonetheless, in this paper we only study the case where each iteration of the single-user decoders is followed by a message exchange through the MAC nodes. This decoder has $\mathcal{O}(n)$ complexity.

C. Coset Codes

To simplify the analysis we consider the ensemble of cosets of LDPC codes where each code in this ensemble is specified by a graph $\mathcal{G}$ and a 'dither' vector $\tilde{\mathbf{d}} \in \{0,1\}^n$ with its BPSK representation $\mathbf{d} \in \{\pm 1\}^n$. The ensemble is then specified by a degree distributions pair $(\lambda(x), \rho(x))$ and the dither vector. We consider the ensemble generated by randomly choosing VN and CN degrees according to the distribution pair $\lambda(x), \rho(x)$ followed by a random permutation between the left sockets and right sockets, and by choosing $\tilde{\mathbf{d}}$ uniformly from $\{0,1\}^n$. Let $\mathcal{C}_{\mathcal{G},\tilde{\mathbf{d}}}$ denote the coset code corresponding to a given $\mathcal{G}$ and $\tilde{\mathbf{d}}$. Let $\mathbf{G}$ and $\mathbf{H}$ denote the generator matrix and parity check matrix of the LDPC code, respectively, with a given $\mathcal{G}$ and $\tilde{\mathbf{d}} = \mathbf{0}$. Then, $\mathbf{m} \in \mathcal{C}_{\mathcal{G},\tilde{\mathbf{d}}}$ if and only if $\mathbf{Hm} = \mathbf{Hd}$.

At the encoders, the bit sequences $\mathbf{b}_1$ and $\mathbf{b}_2$ are encoded into codewords $\mathbf{m}_1$ and $\mathbf{m}_2$, respectively, according to

$$\mathbf{m}_u = \mathbf{G}\mathbf{b}_u + \tilde{\mathbf{d}}, \quad u \in \{1,2\}. \tag{3}$$

Note that both users share the same dither $\tilde{\mathbf{d}}$. Since the BPSK mapping is one-to-one, we can also express the addition of the dither as multiplication of $\mathbf{c}_1, \mathbf{c}_2$ with $\mathbf{d}$, resulting in the channel output

$$y_i = c_{1,i}d_i + c_{2,i-\tau}d_{i-\tau}.$$

Since $\mathbf{d}$ is chosen as part of the code design, it is known at the receiver and its effect can be easily incorporated into the message passing rules. The analysis in Section V will show that a randomly chosen dither will be good for any code and all codeword combinations with probability approaching 1 as $n \to \infty$.

Remark 3: Note that the constructed LDPC codes are not strictly linear but affine. Nonetheless, they can be encoded with a linear encoder followed by a common offset. Besides, numerical results suggest that the error probabilities stay unchanged when no dithering is used. As such, the dither is mainly used as an analytic tool here.

V. Density Evolution Analysis

We next track the fraction of erased edges through the iterations averaged over the code and dither ensemble as $n \to \infty$. Let x_l be the probability that a message from a variable node to a check node is erased, y_l the probability that a message from a check node to a variable node is erased, w_l the probability that a message from a variable node to a MAC node is erased, and z_l the probability that a message from a MAC node to a variable node is erased. The subscript l refers to the l-th iteration. The passed messages are visualized in Fig. 2.

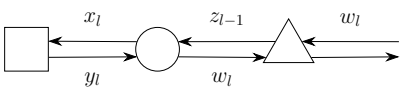

Fig. 2: Fraction of erased messages between VNs, CNs and MAC nodes.

Assuming that the depth l neighborhood of each node is a tree, we can derive a recursion for the evolution of the above parameters as follows. Begin with initial conditions $y_0 = 1, x_0 = 1, z_0 = 1/2$

$$\begin{aligned} x_{l+1} &= z_l \lambda(y_l) && (4)\\ y_{l+1} &= 1 - \rho\left(1 - x_{l+1}\right) && (5)\\ w_{l+1} &= L\left(y_{l+1}\right) && (6)\\ z_{l+1} &= \frac{1}{2} w_{l+1}. && (7) \end{aligned}$$

These equations are obtained by following the basic message passing rules. An edge from a degree i VN to a CN is erased if all incoming edges are erased. The VN has a total of $i-1$ incoming edges from other CNs which are independently erased with probability y_l and one incoming edge from a MAC node which is erased with probability z_l, resulting in an erasure probability $z_l y_l^{i-1}$. Averaging over all VN degrees gives the expression for x_{l+1}. The other equations are derived similarly. The factor $1/2$ in z_{l+1} arises since the value of each MAC node is independently erased with probability $1/2$. Note that this is only true because of the symmetrization by the dither.

By performing some standard substitutions, we end up with the following scalar recursion:

$$x_{l+1} = \frac{1}{2} L\left(1 - \rho\left(1 - x_l\right)\right) \lambda\left(1 - \rho\left(1 - x_l\right)\right). \quad (8)$$

Likewise, we can obtain the following recursion on y_l:

$$y_{l+1} = 1 - \rho\left(1 - \frac{1}{2} L(y_l)\lambda(y_l)\right). \quad (9)$$

The probability that a bit remains erased at the end of iteration $l+1$ is given by

$$p_{l+1} = z_l L(y_{l+1}), \quad (10)$$

where $(p_l)_{l=1,2,\ldots}$ is a deterministic sequence of numbers. Our main theorem below shows that the BER of a randomly chosen code with a random dither sequence after l decoding iterations concentrates tightly around p_l. Let

$$\begin{aligned} P_b(\mathbf{d}, \mathbf{c}, l) &:= P_b(\mathbf{c}, \mathcal{G}, n, l, \mathbf{d}, \tau) \\ &= \frac{1}{2n} \sum_{i=1}^{2n} \mathbb{E}[\mathbb{1}\{v_i^l = \epsilon\} | \mathcal{G}, \mathbf{d}] \end{aligned} \quad (11)$$

be the BER (fraction of erased VNs) at blocklength n after l iterations for a given code $\mathcal{G} \in \text{LDPC}(\lambda, \rho)$ and codeword pair $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2)$. Also let $\bar{P}_b(\mathbf{d}, l) = \frac{1}{|\mathcal{C}|^2} \sum_{\mathbf{c}} P_b(\mathbf{d}, \mathbf{c}, l)$ denote the average BER. Then the following holds:

Theorem 2: As $n \to \infty$, for any $\tau \in [1 : \tau_{\max}]$

$$\mathbb{P}_{\mathcal{G},\mathbf{d}}(|\bar{P}_b(\mathbf{d}, l) - p_l| > \lambda) \to 0 \quad (12)$$

for any $\lambda > 0$. □

Proof: The proof is given in [1]. ■

VI. Optimization

We can use the DE equations to optimize the degree distributions. Specifically, define

$$f_\rho(y) = y - 1 + \sum_{i=2}^{r_{\max}} \rho_i \left(1 - \frac{1}{2} L(z(y)\lambda(y)\right)^{i-1} \quad (13)$$

where $r_{\max}$ is the maximal CN degree. For fixed λ, (13) is linear in ρ_i and gives rise to the linear program:

$$\begin{aligned} \min_{\rho} \quad & \sum_i \frac{\rho_i}{i} \\ \text{s.t.} \quad & \rho_i \geq 0; \sum_i \rho_i = 1; f_\rho(y) > \delta \ \forall y \in (0,1) \end{aligned} \quad (14)$$

where $\delta \geq 0$ is a slack variable. For fixed ρ, (8) results in an optimization problem with linear objective and quadratic constraints. Details on the quadratic program are given in [1]. Unfortunately, it can be shown that the constraints are not positive semidefinite. Therefore, the problem is not convex in general and a solver is not guaranteed to converge to the optimal solution. Nonetheless, we find that general purpose quadratic solvers lead to good results and we are able to empirically find degree distributions that achieve rates close to the BAC capacity by alternating optimization of ρ and λ. To find distributions which can be decoded in a reasonable amount of iterations and are robust to finite length fluctuations we follow [20, Sec. VII] and set the slack variable to $\delta = c/\sqrt{n}$. The parameter c is set empirically. Higher c will result in lower rates but less required decoding iterations.

A. Error-Floor Analysis

In single-user LDPC ensemble constructions, degree one VNs are usually avoided because they prevent the BER (and the BLER) from going to zero. Indeed, when two degree one VNs connect to the same CN, they create a low-weight stopping set that cannot be recovered, even by an ML decoder. However, for the two-user frame-asynchronous case, under certain circumstances, the presence of degree one VNs does not prevent the BLER from going to zero as $n \to \infty$. This

	Code 1	Code 2	Code 3
L_1	0.376	0.560	0.444
L_2	0.594	0.371	0.445
L_5	0.014		
L_6	0.016		
L_7		0.061	
L_8		0.008	0.111
R_4	0.586	0.128	0.323
R_5	0.188	0.582	0.489
R_{10}	0.227	0.290	
R_{20}			0.188
Design Rate	0.689	0.716	0.733
Mean Iterations	30	30	100

TABLE I: Degree distributions for three codes at different rates.

implies that we can increase the rates in the finite-blocklength regime without introducing error floors by introducing a small fraction of degree one VNs.

In the joint graph, degree one VNs can be recovered through the MAC nodes, even if they connect to the same CN. An example of a cycle of length 4 is depicted in Fig. 3. More details on the appearance of cycles can be found in [1].

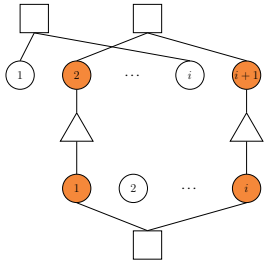

Fig. 3: Stopping set of size 4 in a joint graph for $\tau = 1$

VII. Numerical Results

Table I shows some degree distributions obtained using the optimization procedure given in Section VI. The slack variable δ was adjusted empirically to find codes that work with small blocklength and a reasonable number of required iterations. The erasure probability for Code 2 in Table I predicted from DE is shown in Fig. 4 together with some random decoding realizations with blocklength $n = 5 \cdot 10^4$. The empirical block error rate (BLER) of the codes in Table I is shown in Fig. 5 for a fixed delay $\tau = 1$. For the code construction we choose a random sample from the permutation ensemble and we check if it contains $4K$-stopping sets up to $K = 3$. If it does, we sample again. The number of required samples is typically less than 10 for Code 2 and between zero and two for Codes 1 and 3. We can see in Fig. 5 that the resulting codes do not show an error floor. The case with random delay $\tau \in [1 : \tau_{\max}]$ is explored in Fig. 6. We choose $\tau_{\max} = 100$ for Code 1 and $\tau_{\max} = 500$ for Codes 2 and 3. The reason for choosing a smaller $\tau_{\max}$ for Code 1 is that for $n < 1000$, a delay of several hundred symbols is a significant fraction of the blocklenght, in which case the number of symbols where both codewords collide is rather small and hence, the BER is small, too. This effect also explains the non-monotonic

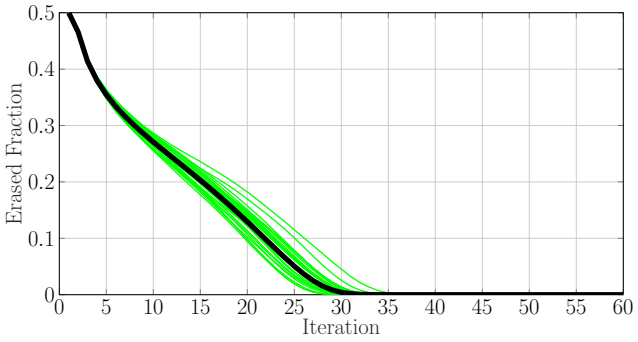

Fig. 4: Erased fraction of VNs as a function of the number of iterations for Code 2. The black thick line represents the erasure probability from DE. The thin lines are sample paths for $n = 5 \cdot 10^4$.

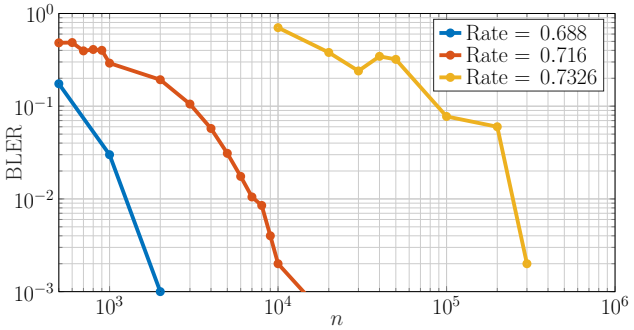

Fig. 5: BLER as a function of n for $\tau = 1$.

behavior of the BER for Code 2. Note that both BLER and BER are limited by $1/\tau_{\max}$ because $\tau = 0$ will always result in a block error. As expected from the analysis in Section VI-A, the codes exhibit an error floor due to short length stopping sets caused by degree one VNs and therefore the corresponding BLERs do not vanish. We can observe in the simulations that for large enough n, block errors are caused almost exclusively by 4 remaining bit-errors for Code 1 and 3, while Code 2 also occasionally exhibits 8 or 12 remaining bit-errors. Thus, a high-rate outer code would be sufficient to resolve the remaining bit-errors in this case. For example, a BCH code would suffice with minimum distance 8 or 24, respectively.

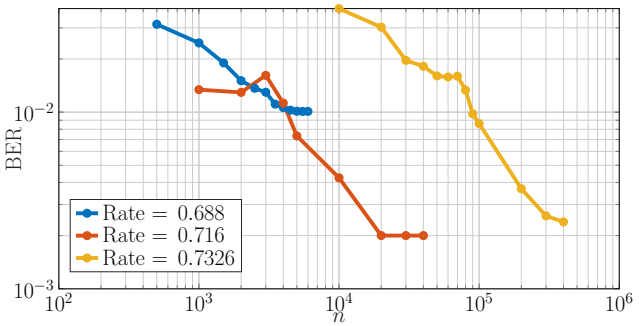

Fig. 6: BER as a function of n for random $\tau \in [0, \tau_{\max}]$, with $\tau_{\max} = 100$ for Code 1, and $\tau_{\max} = 500$ for Code 2 and 3.

REFERENCES

[1] A. Fengler, A. Lancho, K. Narayanan, and Y. Polyanskiy, *On the Advantages of Asynchrony in the Unsourced MAC*, May 2023. arXiv: 2305.06985.

[2] Y. Polyanskiy, "A perspective on massive random-access," in *2017 IEEE International Symposium on Information Theory (ISIT)*, Jun. 2017, pp. 2523–2527. DOI: 10.1109/ISIT.2017.8006984.

[3] 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. DOI: 10.1109/TIT.2021.3081189.

[4] A. K. Pradhan, V. K. Amalladinne, K. R. Narayanan, and J. Chamberland, "Polar Coding and Random Spreading for Unsourced Multiple Access," in *ICC 2020 - 2020 IEEE Int. Conf. Commun. ICC*, Jun. 2020, pp. 1–6. DOI: 10.1109/ICC40277.2020.9148687.

[5] E. Marshakov, G. Balitskiy, K. Andreev, and A. Frolov, "A Polar Code Based Unsourced Random Access for the Gaussian MAC," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, Sep. 2019, pp. 1–5. DOI: 10.1109/VTCFall.2019.8891583.

[6] S. S. Kowshik, K. Andreev, A. Frolov, and Y. Polyanskiy, "Energy efficient coded random access for the wireless uplink," *IEEE Trans. Commun.*, pp. 1–1, 2020. DOI: 10.1109/TCOMM.2020.3000635.

[7] V. K. Amalladinne, J.-F. Chamberland, and K. R. Narayanan, "A Coded Compressed Sensing Scheme for Unsourced Multiple Access," *IEEE Trans. Inf. Theory*, 2020. DOI: 10.1109/TIT.2020.3012948.

[8] D. Truhachev, M. Bashir, A. Karami, and E. Nassaji, "Low-Complexity Coding and Spreading for Unsourced Random Access," *IEEE Commun. Lett.*, vol. 25, no. 3, pp. 774–778, Mar. 2021. DOI: 10.1109/LCOMM.2020.3039436.

[9] A. Fengler, O. Musa, P. Jung, and G. Caire, "Pilot-Based Unsourced Random Access with a Massive MIMO Receiver, Interference Cancellation, and Power Control," *IEEE J. Sel. Areas Commun.*, pp. 1–1, 2022. DOI: 10.1109/JSAC.2022.3144748.

[10] G. Liva and Y. Polyanskiy, "On Coding Techniques for Unsourced Multiple-Access," in *2021 55th Asilomar Conf. Signals Syst. Comput.*, Oct. 2021, pp. 1507–1514. DOI: 10.1109/IEEECONF53345.2021.9723359.

[11] A. Fengler, G. Liva, and Y. Polyanskiy, "Sparse Graph Codes for the 2-User Unsourced MAC," in *2022 56th Asilomar Conf. Signals Syst. Comput.*, Nov. 2022.

[12] T. Cover, R. McEliece, and E. Posner, "Asynchronous multiple-access channel capacity," *IEEE Trans. Inform. Theory*, vol. 27, no. 4, pp. 409–413, Jul. 1981. DOI: 10.1109/TIT.1981.1056382. (visited on 12/28/2022).

[13] A. Decurninge, P. Ferrand, and M. Guillaud, "Massive Random Access with Tensor-based Modulation in the Presence of Timing Offsets," in *GLOBECOM 2022 - 2022 IEEE Glob. Commun. Conf.*, Dec. 2022, pp. 1061–1066. DOI: 10.1109/GLOBECOM48099.2022.10001729.

[14] X. Chen, L. Liu, D. Guo, and G. W. Wornell, "Asynchronous Massive Access and Neighbor Discovery Using OFDMA," *IEEE Trans. Inf. Theory*, pp. 1–1, 2022. DOI: 10.1109/TIT.2022.3224951.

[15] A. Roumy and D. Declercq, "Characterization and Optimization of LDPC Codes for the 2-User Gaussian Multiple Access Channel," *J Wireless Com Network*, vol. 2007, no. 1, p. 074 890, Dec. 2007. DOI: 10.1155/2007/74890. (visited on 11/03/2022).

[16] A. Balatsoukas-Stimming, S. Rini, and J. Kliewer, "LDPC Coded Multiuser Shaping for the Gaussian Multiple Access Channel," in *2019 IEEE Int. Symp. Inf. Theory ISIT*, Jul. 2019, pp. 2609–2613. DOI: 10.1109/ISIT.2019.8849785.

[17] R. G. Gallager, *Information Theory and Reliable Communication*. New York: Wiley, 1968.

[18] A. E. Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, Dec. 2011.

[19] T. Richardson and R. Urbanke, *Modern Coding Theory*, Illustrated edition. Cambridge ; New York: Cambridge University Press, Mar. 2008.

[20] A. Shokrollahi, "Raptor codes," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006. DOI: 10.1109/TIT.2006.874390.