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 [1]. 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 [1], [2]. While many unsourced code constructions have been proposed [2]–[8], 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 [9], [10] 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 [11] 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 [12], [13]. 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 [14], [15]. For the code construction presented in [15] 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 [14], 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 program, the codes in [14] are constrained such that variable nodes (VNs) that are connected through the MAC have the

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 (NSF) 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)

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. [16], 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 [17]. 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 give the following result, which shows that random linear codes can achieve the two-user BAC capacity if a frame delay of just one symbol is introduced.

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 Appendix A. ■

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 in Appendix A 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 $\mathbf{H}\mathbf{m}_1=\mathbf{H}\mathbf{m}_2=\mathbf{0}$ 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., [18]. 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 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.

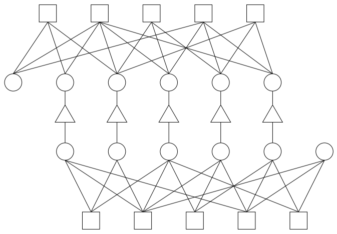

Fig. 1: Factor Graph for a UBAC with $\tau=1$. Triangles denote MAC nodes, squares are CNs, circles are VNs.

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 $\mathbf{d}\in\{\pm1\}^n$. 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 $\mathbf{H}\mathbf{m}=\mathbf{H}\tilde{\mathbf{d}}$.

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.

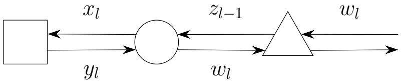

Fig. 2: Fraction of erased messages between VNs, CNs and MAC nodes.

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:

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

Likewise, we can obtain the following recursion on y_l:

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

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

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

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

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

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 $\bar{P}_b(\mathbf{d}, l) = \frac{1}{|C|^2} \sum_{\mathbf{c}} P_b(\mathbf{d}, \mathbf{c}, l)$ denote the average BER. Then the following holds:

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

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

for any $\lambda > 0$. □

Proof: The proof is given in Appendix B. ■

VI. Optimization

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

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

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

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

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 Appendix C. 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 [19, 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 $n \to \infty$. As we shall see, this 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. In the following theorem we provide a bound on the probability that a randomly chosen graph with a fraction L_1 of degree one VNs has a $4K$-sized stopping set, consisting of just degree one VNs. The case $K = 1$ is depicted in Fig. 3.

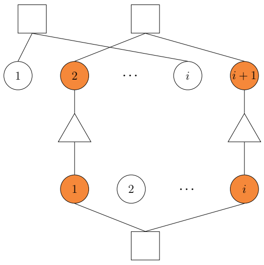

Fig. 3: Stopping set of size 4 in a joint graph for $\tau = 1$

Theorem 3: The probability that a random code from the ensemble LDPC(λ, ρ) results in a joint graph that has no stopping sets of size $\leq 4K$ created by just degree one VNs for all $\tau \in [1 : \tau_{\text{max}}]$ can be bounded from below by

$$1 - \frac{\tau_{\text{max}}}{2} \sum_{k=1}^{K} \left(\frac{L_1^2}{1-R} \right)^k \frac{1}{2k} - \mathcal{O}\left(\frac{K}{n^2} \right) \tag{15}$$

□

Proof: See Appendix D. ■

The above theorem also implies the following result on the BLER.

Theorem 4: If L_1 is sufficiently small compared to τ_{max} such that (15) is strictly larger than zero, there exists a constant fraction of codes in the ensemble with a vanishing BLER. □

Proof: See Appendix D. ■

Theorem 4 shows that error floors can be avoided by resampling the code until one is found where the joint graph contains no $4K$-sized stopping sets for a desired range τ_{max}. It is necessary that τ_{max} is small compared to $L_1^2/(1-R)$.[1] Note that even if $4K$-sized stopping sets of degree one VNs exist, they only result in bit-errors if all VNs in the set are erased (i.e., users transmit different symbols), which happens with probability 2^{-4K}. Therefore it may not be necessary to expurgate these sets for large K, depending on the desired BLERs. Besides, fixed length stopping sets result in a number of bit-errors which does not scale with n. As such, they could also be corrected by adding an outer code with rate approaching 1 as $n \to \infty$. See also the discussion in [20].

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 $\tau \in [1 : \tau_{\text{max}}]$ is explored in Fig. 6. We choose $\tau_{\text{max}} = 100$ for Code 1 and $\tau_{\text{max}} = 500$ for Codes 2 and 3. The reason for choosing a smaller τ_{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 behavior of the BER for Code 2. Note that both BLER and BER are limited by $1/\tau_{\text{max}}$ 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.

	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.

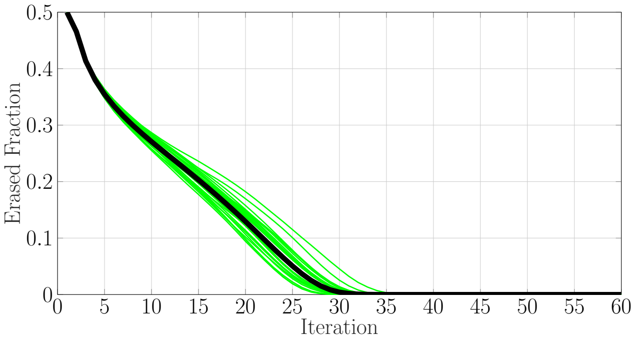

Fig. 4: Erased fraction of VNs as a function of the number of iterations for Code 2. The black thick line represents the erasure probability from DE. The thin lines are sample paths for $n = 5 \cdot 10^4$.

[1] This can be the case in applications where synchronization is possible but a small shift τ is deliberately introduced at the transmitters.

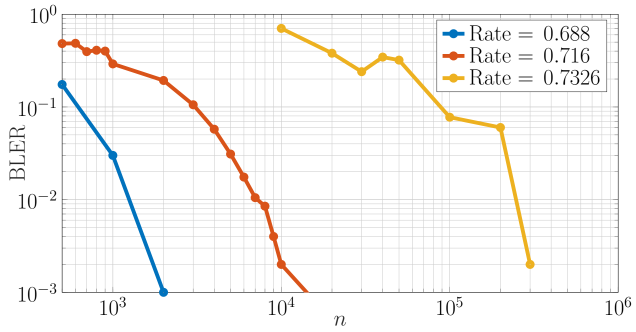

Fig. 5: BLER as a function of n for $\tau = 1$.

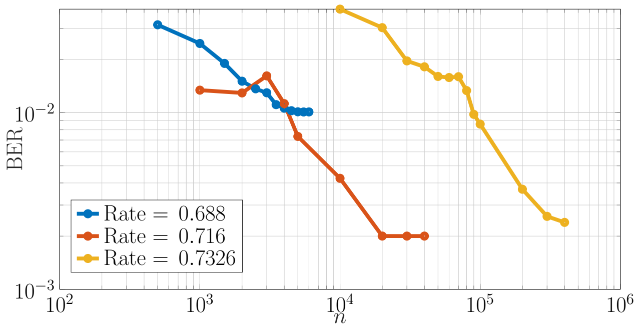

Fig. 6: BER as a function of n for random $\tau \in [0, \tau_{\max}]$, with $\tau_{\max} = 100$ for Code 1, and $\tau_{\max} = 500$ for Code 2 and 3.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

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

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

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

[20] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001. DOI: 10.1109/18.910577.

[21] W. Hoeffding, "Probability Inequalities for Sums of Bounded Random Variables," *J. Am. Stat. Assoc.*, vol. 58, no. 301, pp. 13–30, 1963. DOI: 10.2307/2282952.

[22] K. Azuma, "Weighted sums of certain dependent random variables," *Tohoku Math. J. (2)*, vol. 19, no. 3, Jan. 1967. DOI: 10.2748/tmj/1178243286.

[23] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," in *Proc. Thirtieth Annu. ACM Symp. Theory Comput. - STOC 98*, Dallas, Texas, United States: ACM Press, 1998, pp. 249–258. DOI: 10.1145/276698.276756.

Appendix A
Proof of Theorem 1

Proof: We start by reformulating the decoding problem on the frame-asynchronous UBAC in terms of the parity check matrix $\mathbf{H} \in \mathbb{F}_2^{n-k\times n}$ of a linear code. Let $\mathbf{m}_1, \mathbf{m}_2$ be two codewords, i.e., $\mathbf{H}\mathbf{m}_1 = \mathbf{H}\mathbf{m}_2 = 0$, and let both codewords be transmitted through the BAC (1) for some fixed τ. Let $\mathcal{E} = \{i : y_i = 0\}$ and denote the shifted set by $\mathcal{E} - \tau := \{i : y_{i+\tau} = 0\}$. For $i \in \mathcal{E}$ we have $m_{1,i} = 1 - m_{2,i-\tau}$. Let $\mathbf{H}_\mathcal{E}$ denote the sub-matrix of $\mathbf{H}$ with column indices in $\mathcal{E}$ and, analogously, $\mathbf{m}_{u,\mathcal{E}}$ be the restriction of $\mathbf{m}_u$ to the set $\mathcal{E}$. Note that $\underline{\text{on } \overline{\mathcal{E}}} = [n] \setminus \mathcal{E}$ the entries of $\mathbf{m}_1$ are known and similarly on $\overline{\mathcal{E} - \tau}$ the entries of $\mathbf{m}_2$ are known. Therefore, we can compute the two syndromes:

$$\begin{aligned} \mathbf{s}_1 &= \mathbf{H}_{\overline{\mathcal{E}}}\mathbf{m}_{1,\overline{\mathcal{E}}} \\ \mathbf{s}_2 &= \mathbf{H}_{\overline{\mathcal{E}-\tau}}\mathbf{m}_{2,\overline{\mathcal{E}-\tau}} \end{aligned} \tag{16}$$

and $\mathbf{m}_1$ satisfies the two constraints:

$$\begin{aligned} \mathbf{H}_\mathcal{E}\mathbf{m}_{1,\mathcal{E}} &= \mathbf{s}_1 \\ \mathbf{H}_{\mathcal{E}-\tau}\mathbf{m}_{1,\mathcal{E}} &= \mathbf{H}_{\mathcal{E}-\tau}(\mathbf{1} - \mathbf{m}_{2,\mathcal{E}-\tau}) \\ &= \mathbf{H}_{\mathcal{E}-\tau}\mathbf{1} - \mathbf{s}_2 =: \tilde{\mathbf{s}}_2 \end{aligned} \tag{17}$$

We can define $\tilde{\mathbf{H}} = [\mathbf{H}\ \mathbf{0}_\tau; \mathbf{0}_\tau\ \mathbf{H}] \in \mathbb{F}_2^{2(n-k)\times(n+\tau)}$. With this (17) can be written as $\tilde{\mathbf{H}}_\mathcal{E}\mathbf{m}_{1,\mathcal{E}} = [\mathbf{s}_1; \tilde{\mathbf{s}}_2]$. This equations can be solved if the rank of $\text{rank}(\tilde{\mathbf{H}}_\mathcal{E}) > |\mathcal{E}|$.

Now let the entries of the parity check matrix $\mathbf{H}$ be Bernoulli(1/2) i.i.d. and define $r = n - k$. For some arbitrary erasure set $\mathcal{E} \subset [\tau : n]$ of size d we compute the probability P_d that a sub-matrix $\tilde{\mathbf{H}}_\mathcal{E}$ of $\tilde{\mathbf{H}}$ of size $2r \times d$ has rank d. Note that this probability is well defined since it does only depend on the size of $\mathcal{E}$ but not on the actual set. The complications in the proof, compared to standard techniques, arise from the fact that $\tilde{\mathbf{H}}$ may contain the same vectors in top and bottom half, in which case we cannot assume anymore that they are independent. We can bound P_d as follows:

$$P_d \geq \prod_{k=1}^{d}\left(1 - \frac{2^{k+1}}{2^{2r}}\right) \tag{18}$$

To get the bound we compute the smaller probability $\tilde{P}_d$ that $\mathbf{H}_\mathcal{E}$ has rank d *and* the following condition is fullfilled:

i) Non of the top half vector from $\mathbf{H}_\mathcal{E}$ are in the column span of the bottom half $\mathbf{H}_{\mathcal{E}-\tau}$.

We compute $\tilde{P}_d$ recursively by adding the indices in $\mathcal{E}$ in increasing order: Let $\mathcal{E}_k$ denote the sub-set of $\mathcal{E}$ with only the first k indices. Assume the columns of $\tilde{\mathbf{H}}_{\mathcal{E}_{k-1}}$ are linearly independent and condition i) is satisfied. If one column $\tilde{\mathbf{h}}_i := [\mathbf{h}_i; \mathbf{h}_{i-\tau}]$ is added, the resulting set will be linearly dependent if $\{\tilde{\mathbf{h}}_i \in \text{span}(\tilde{\mathbf{H}}_{\mathcal{E}_{k-1}-\tau})\} := \mathcal{I}_1$. In addition, condition i) will be broken if $\{\mathbf{h}_i \in \text{span}(\mathbf{H}_{\mathcal{E}_{k-1}-\tau})\} := \mathcal{I}_2$ happens. $\mathcal{I}_1$ can be further decomposed into the two disjoint events $\mathcal{I}_{1,1} = \mathcal{I}_1 \cap \{i - \tau \in \mathcal{E}_{k-1}\}$ and $\mathcal{I}_{1,2} = \mathcal{I}_1 \cap \{i - \tau \notin \mathcal{E}_{k-1}\}$. The conditional probability of $\mathcal{I}_{1,1}$ is zero due to the assumption that condition i) is fullfilled. On the other hand, if $i - \tau \notin \mathcal{E}_{k-1}$ then $\mathbf{h}_{i-\tau}$ is independent of $\tilde{\mathbf{H}}_{\mathcal{E}_{k-1}}$ and the probability that $\mathcal{I}_{1,2} \cap \mathcal{I}_2$ happens is $(1 - (2^{k-1} + 2^k)/2^{2r})$ since there are 2^{k-1} binary vectors in the span of $\mathbf{H}_{\mathcal{E}_{k-1}}$ and 2^k vectors in the span of $[\mathbf{H}_{\mathcal{E}_{k-1}-\tau}, \mathbf{h}_{i-\tau}]$. So we can bound $\tilde{P}_d$ as

$$\tilde{P}_d \geq \left(1 - \frac{2^{k+1}}{2^{2r}}\right)\tilde{P}_{d-1} \tag{19}$$

which proves (18).

For a fixed parity check matrix $\mathbf{H}$, the probability of decoding the first codeword wrong, averaged over all codeword pairs, is given by

$$P_{e,\mathbf{H}} = 1 - \sum_{d=1}^{n-\tau}\mathbb{P}(|\mathcal{E}| = d)\mathbb{1}(\text{rank}(\tilde{\mathbf{H}}_\mathcal{E}) = d) \tag{20}$$

Let $n' := n - \tau$ and $d_{\text{max}} = n'/2 + \delta n'$. We can write $|\mathcal{E}| = \sum_{i=1}^{n'} X_i$ where we define $X_i := \mathbb{1}(c_{1,i} = -c_{2,i-\tau})$. It holds that $\mathbb{E}[X_i] = 1/2$, $\text{Var}[X_i] = 1/4$, [2] and the X_i are *pairwise* independent [3]. Therefore $\text{Var}(|\mathcal{E}|) = n/4$ and Chebychev's inequality shows that for any $\delta > 0$

$$P\left(\left||\mathcal{E}| - \frac{n}{2}\right| > \delta n\right) \leq \frac{1}{4\delta n}. \tag{21}$$

Since $\mathbb{1}(\text{rank}(\tilde{\mathbf{H}}_\mathcal{E}) \geq d)$ is non-increasing in d we have

$$\begin{aligned} P_{e,\mathbf{H}} &\leq 1 - \mathbb{1}(\text{rank}(\tilde{\mathbf{H}}_\mathcal{E}) = d_{\text{max}})P(|\mathcal{E}| \leq d_{\text{max}}) \\ &\leq 1 - \mathbb{1}(\text{rank}(\tilde{\mathbf{H}}_\mathcal{E}) = d_{\text{max}}) + o_n(1). \end{aligned} \tag{22}$$

Note that the channel is noiseless. Therefore, if one codeword is correctly recovered the second one can be obtained by subtracting the first. Also, it is irrelevant which codeword we attempt to decode since both users share the same $\tilde{H}_\mathcal{E}$. For simplicity we assume that $d_{\text{max}} = (n - \tau + \delta)/2$ is an integer. Averaging (22) over the code ensemble we get

$$\begin{aligned} P_e &\leq 1 - \tilde{P}_{(n-\tau+\delta)/2} + o_n(1) \\ &= \frac{n - \tau + \delta}{2}2^{n(1/2-2(1-R))} + o_n(1) \end{aligned} \tag{23}$$

■

An alternative proof for the slightly different linear code ensemble of iid random generator matrices $\mathbf{G} \in \mathbb{F}_2^{k\times n}$ can be sketched as follows: Let $(x_i)' = x_{i-\tau}$ for $i > \tau$ denote the left-shift of a vector entry by some fixed τ and let $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{F}_2^k$ be the transmitted bit sequences. Then the channel output reads as

$$\mathbf{y} = (\mathbf{u}_1\mathbf{G}) + (\mathbf{u}_2\mathbf{G})' \tag{24}$$

W.l.o.g. we can choose $\mathbf{u}_1 = \mathbf{e}_1$ and $\mathbf{u}_2 = \mathbf{e}_2$. For arbitrary $\mathbf{u}_1, \mathbf{u}_2$ we can find a basis which has $\mathbf{u}_1, \mathbf{u}_2$ as first two basis vectors and work in the new basis. Since the distribution of $\mathbf{G}$

[2] This holds for all typical codes, that is those which have a marginal bit distributions close to Bernoulli(1/2). It can be shown easily that all but an exponentially small fraction of random parity check codes are typical. So we can restrict parity check matrices to be typical.

[3] Again, this is true for all but an exponentially small fraction of parity check matrices. This can be seen by bringing $\mathbf{H}$ into systematic form. Then the first k data bits are clearly independent and two of the $n - k$ parity check bits are dependent if and only if they are sums of the exact same set of bits.

is invariant under basis change this does not affect the error probability.

Conditioned on the two rows $\mathbf{g}_1, \mathbf{g}_2$ the error probability is given by

$$P_{e,\mathbf{G}} = \mathbb{P}(\bigcup_{\mathbf{v}_1,\mathbf{v}_2} \{(\mathbf{v}_1\mathbf{G}) + (\mathbf{v}_2\mathbf{G})' = \mathbf{g}_1 + \mathbf{g}_2'\}|\mathbf{g}_1, \mathbf{g}_2) \quad (25)$$

We partition the space of possible sequences $\mathbf{v}_1, \mathbf{v}_2$ into two sets $\mathcal{A}$ and $\mathcal{A}^c$ such that sequences in $\mathcal{A}$ are zero in the first two positions. Within $\mathcal{A}$, $\mathbf{v}_1\mathbf{G}, \mathbf{v}_2\mathbf{G}$ are independent of $\mathbf{g}_1, \mathbf{g}_2$. Furthermore, we distinguish two cases. First, that $\mathbf{v}_1$ and $\mathbf{v}_2$ both contain at least one unique bit. In this case we can treat them as independent vectors and bound the error probability, averaged over $\mathbf{G} \setminus [\mathbf{g}_1, \mathbf{g}_2]$ as

$$\begin{aligned}P_e \leq\ & 2^{2(k-2)}\mathbb{P}(b_1 + b_2 = 0)^{|\mathcal{Y}_0|}\mathbb{P}(b_1 + b_2 = 1)^{|\mathcal{Y}_1|} \\ & \mathbb{P}(b_1 + b_2 = 2)^{|\mathcal{Y}_2|}\end{aligned} \quad (26)$$

where b_1, b_2 are independent Bernoulli(1/2) distributed bits and $\mathcal{Y}_l = \{i : y_i = l\}$. Averaging over $\mathbf{g}_1, \mathbf{g}_2$ gives

$$\begin{aligned}P_e &\leq 2^{2(k-2)} \left(\frac{1}{4}\right)^{n/4} \left(\frac{1}{2}\right)^{n/2} \left(\frac{1}{4}\right)^{n/4} + o_n(1) \\ &= 2^{n(2R-1.5)} + o_n(1)\end{aligned} \quad (27)$$

In the second case, where $\mathbf{v}_2$ is of the form $\mathbf{v}_2 = \mathbf{v}_1 \oplus \mathbf{u}$ for some vector $\mathbf{u}$ that is independent of $\mathbf{v}_1$. We get probabilities of the form $\mathbb{P}(b_1 + (b_1 \oplus b_2) = j)$ for $j \in \{0, 1, 2\}$, leading to (we skip the intermediate steps):

$$P_e \leq 2^{2(k-2)} \left(\frac{1}{4}\right)^n + o_n(1) = 2^{n(2R-2)} + o_n(1) \quad (28)$$

It remains to estimate the error probabilities in $\mathcal{A}^c$. First note that whenever $\mathbf{v}_1, \mathbf{v}_2$ both contain at least one unique bit they can be treated as independent and we get the same bound as (27). Also, if only one of them contains a random vector, e.g., $\mathbf{v}_1 = \mathbf{g}_1, \mathbf{v}_2 = \mathbf{g}_2 \oplus \mathbf{u}$, then there is at most one values of u_i for each i which replicates the channel values. Resulting in

$$P_e \leq 2^{k-2} \left(\frac{1}{2}\right)^n + o_n(1) = 2^{n(R-1)} + o_n(1) \quad (29)$$

This leaves only 15 possible cases, most of which are trivial or can be reduced by symmetry to one of the following 4 non-trivial cases:

- Case 1: $\mathbf{v}_1 = \mathbf{g}_1 \oplus \mathbf{u}, \mathbf{v}_2 = \mathbf{g}_2 \oplus \mathbf{u}$
  We explicitly write down the equations that need to be satisfied for an error to occur (wlog for $\tau = 1$):
  $$g_{1,i} \oplus u_i + g_{2,i-1} \oplus u_{i-1} = g_{1,i} + g_{2,i-1} \quad (30)$$
  This can only be satisfied if $u_i = u_{i-1}$ which happens with probability 1/2. Therefore we can bound the error probability
  $$P_e \leq 2^{k-2} \left(\frac{1}{2}\right)^n + o_n(1) = 2^{n(R-1)} + o_n(1) \quad (31)$$
- Case 2: $\mathbf{v}_1 = \mathbf{g}_1 \oplus \mathbf{g}_2 \oplus \mathbf{u}, \mathbf{v}_2 = \mathbf{u}$
  In this case there are some channel values that cannot be replicated. E.g. $g_{1,i} = g_{2,i} = 0$ and $g_{2,i-1} = 1$, then $y_i = 1$, but $g_{1,i} \oplus g_{2,i} = 0$. So neither values if u_i can replicate the channel output. Therefore $P_e = 0 + o_n(1)$.
- Case 3: $\mathbf{v}_1 = \mathbf{g}_1 \oplus \mathbf{g}_2 \oplus \mathbf{u}, \mathbf{v}_2 = \mathbf{g}_1 \oplus \mathbf{u}$
  Similar to Case 2, giving $P_e = 0 + o_n(1)$ similar to Case 2.
- Case 4: $\mathbf{v}_1 = \mathbf{g}_1 \oplus \mathbf{u}, \mathbf{v}_2 = \mathbf{u}$ If $y_i = 1$ both $(u_i, u_{i-1}) = (0, 1)$ and $(u_i, u_{i-1}) = (1, 0)$ result in the correct channel output for both $g_{1,i} = 0$ and $g_{1,i} = 1$
  $$P_e \leq 2^{k-2} \left(\frac{1}{4}\right)^{n/2} \left(\frac{1}{2}\right)^{n/2} + o_n(1) = 2^{n(R-1.5)} + o_n(1) \quad (32)$$

The most restricting constraint is (27), allowing for any $R < 3/4$.

Appendix B
Proof of Theorem 2

The outline of the proof is as follows:

Lemma 1 shows that P_b with fixed dither concentrates around the dither average. Corollary 1 shows that P_b for a fixed codeword pair concentrates around the average over all codeword pairs. Lemma 2 establishes that the dither average is independent of the transmitted codewords. Lemma 3 states that the computation tree for each VN is with high probability tree-like for a fixed depth l as $n \to \infty$. Finally, we argue that that P_b for any fixed random graph concentrates around the ensemble average, which concludes the proof of Theorem 2.

The proof will make repeated use of Azuma-Hoeffding's inequality [21], [22] applied to so called Doob martingales, which are conditional expectations of the form

$$Y_i = \mathbb{E}[f(X_1, ..., X_n)|X_1 = x_1, .., , X_i = x_i] \quad (33)$$

for some function f and a (not necessarily iid) sequence of RVs $(X_i)_{i=1,...,n}$. It holds that $Y_0 = \mathbb{E}[f]$, $Y_n = f(x_1, ..., x_n)$, and

Theorem 5 (Azuma-Hoeffding for Doob Martingales): Suppose that $|Y_k - Y_{k-1}| \leq d_k$ for a sequence $(d_k)_{k=1,...,n}$ of non-negative reals. Then for $\lambda > 0$ it holds

$$\mathbb{P}(|Y_n - Y_0| > \lambda) \leq 2 \exp\left(-\frac{\lambda^2}{2\sum_{k=1}^n d_k^2}\right) \quad (34)$$

□

The next lemma shows that for two fixed transmitted codewords any randomly chosen dither sequence, with high probability, will result in a bit-error rate that is close to the bit-error rate averaged over all dither sequences.

Lemma 1:

$$\mathbb{P}(|P_b(\mathbf{d}, \mathbf{c}_1, \mathbf{c}_2) - \mathbb{E}[P_b(\mathbf{d}, \mathbf{c}_1, \mathbf{c}_2)]| > \lambda) \leq \exp(-C\lambda n) \quad (35)$$

for some constant $C > 0$ and any $\lambda > 0$.

Proof: Define the Doob martingale $Y_i = \mathbb{E}[P_b(\mathbf{d})|d_1, ..., d_i]$. Since any dither value d_i affects at most two VNs and all VNs included in their depth l

computation graphs, the number of affected VNs is upper bounded by a constant that does not scale with n. This constant can be bounded by the maximal VN and CN degrees in the graph as we will show later as part of the proof of Lemma 3. Therefore Y_i has bounded increments and the concentration inequality (35) follows from (34). ■

In fact, also the stronger statement holds, that a randomly chosen dither can be used for all codeword pairs $(\mathbf{c}_1, \mathbf{c}_2)$.

Corollary 1:

$$\mathbb{P}\left(\left|\frac{1}{|\mathcal{C}|^2}\sum_{\mathbf{c}_1,\mathbf{c}_2} P_b(\mathbf{d},\mathbf{c}_1,\mathbf{c}_2) - \mathbb{E}[P_b(\mathbf{d})]\right| > \lambda\right) \leq \exp(-C'\lambda n) \tag{36}$$

for some constant $C' > 0$ and any $\lambda > 0$.

Proof: First, it holds that

$$\mathbb{P}_{\mathbf{c}_1',\mathbf{c}_2'}(|\bar{P}_b(\mathbf{d}) - P_b(\mathbf{d},\mathbf{c}_1',\mathbf{c}_2')| > \lambda) \leq \exp(-C''\lambda n) \tag{37}$$

for some constant $C'' > 0$, any $\lambda > 0$ and any fixed $\mathbf{d}$. This can be shown by applying Azuma-Hoeffding's inequality to the martingale that reveals $\mathbf{c}_1'$ and $\mathbf{c}_2'$ component by component. Since each component affects at most a finite number of VNs in the depth l neighborhood, the martingale has bounded increments. Furthermore,

$$\begin{aligned}
&\mathbb{P}_{\mathbf{d}}(|\bar{P}_b(\mathbf{d}) - \mathbb{E}[P_b(\mathbf{d})| > \lambda) \\
&= \mathbb{P}_{\mathbf{c}_1',\mathbf{c}_2',\mathbf{d}}(|\bar{P}_b(\mathbf{d}) - P_b(\mathbf{d},\mathbf{c}_1',\mathbf{c}_2') \\
&+ P_b(\mathbf{d},\mathbf{c}_1',\mathbf{c}_2') - \mathbb{E}[P_b(\mathbf{d})| > \lambda) \\
&\leq \mathbb{P}_{\mathbf{c}_1',\mathbf{c}_2',\mathbf{d}}\left(|\bar{P}_b(\mathbf{d}) - P_b(\mathbf{d},\mathbf{c}_1',\mathbf{c}_2')| > \frac{\lambda}{2}\right) \\
&+ \mathbb{P}_{\mathbf{c}_1',\mathbf{c}_2',\mathbf{d}}\left(|P_b(\mathbf{d},\mathbf{c}_1',\mathbf{c}_2') - \mathbb{E}[P_b(\mathbf{d})| > \frac{\lambda}{2}\right) \\
&\leq 2\exp(-C'\lambda n)
\end{aligned} \tag{38}$$

where the last inequality follows by applying (35) and (37), integrating, and setting $C' = \max\{C, C''\}/2$. ■

The next lemma will show that the channel output, when averaged over the distribution of the dither, is iid and does not depend on the transmitted codewords $\mathbf{c}_1, \mathbf{c}_2$. Therefore, when evaluating $\mathbb{E}[P_b(\mathbf{d})]$, we can assume that both users transmit the all-ones codeword.

Lemma 2: For any two transmitted codewords $\mathbf{c}_1, \mathbf{c}_2$ and any set $\mathcal{S} \subset [\tau + 1 : n]$, each symbol in the channel output is erased independently with probability $1/2$, i.e.,

$$\mathbb{P}_{\mathbf{d}}(\mathbf{y}_{\mathcal{S}} = \mathbf{0}) = \left(\frac{1}{2}\right)^{|\mathcal{S}|}. \tag{39}$$

□

Proof: Since d_i, d_j are independent if $i \neq j$, we have $p(y_i = 0) = p(d_i c_{1,i} + d_{i-\tau}c_{2,i-\tau} = 0) = \frac{1}{2}$ since $d_i c_{1,i}$ and $d_{i-\tau}c_{2,i-\tau}$ are independent and uniform over $\{-1, 1\}$. Dependencies may occur only if d_i is shared in multiple channel outputs. Note that only y_i and $y_{i+\tau}$ include d_i. We can compute

$$\begin{aligned}
&p(y_i = 0, y_{i+\tau} = 0) \\
&= p(y_{i+\tau} = 0 | y_i = 0) p(y_i = 0) \\
&= \frac{1}{2} p(d_{i+\tau}c_{1,i+\tau} + d_i c_{2,i} = 0 | d_i c_{1,i} + d_{i-\tau}c_{2,i-\tau} = 0) \\
&= \frac{1}{2} p(d_{i+\tau}c_{1,i+\tau} = d_{i-\tau}c_{2,i-\tau}) \\
&= \frac{1}{4}
\end{aligned} \tag{40}$$

where the last inequality follows because $d_{i+\tau}$ and $d_{i-\tau}$ are independent. An arbitrary set $\mathcal{S}$ can be handled by using (40) repeatedly. ■

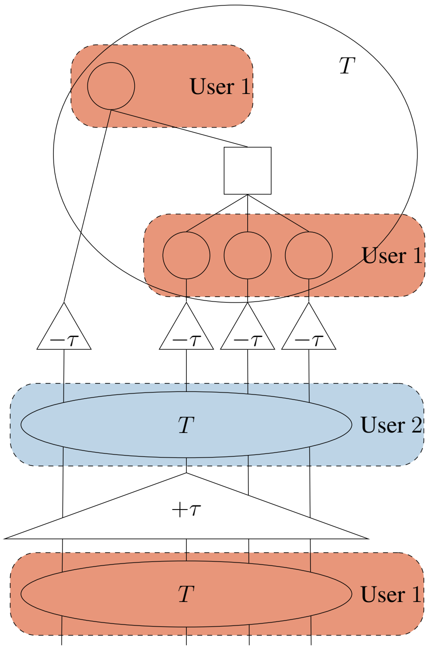

Fig. 7: Computation Graph, T denotes the basic LDPC computation tree with one VN connected to its adjacent CNs which in turn connected to their adjacent VNs.

Next, we show that for a random code from LDPC(λ, ρ) the depth l computation graph rooted at a VN is a tree with high probability. Fig. 7 depicts the structure of the computation graph of an arbitrary VN of user 1. The root node is connected to one MAC node (triangle) and a variable number of check nodes, which in turns connect to other variable nodes, which connect to other check nodes. After this point, at each additional iteration the structure of parity checks followed by MAC nodes is recursively repeated l times. The number of leaves of each element, denoted by T, is an iid random variable whose distribution can be calculated from the left and right degree distributions. Here, we only need an upper bound on the number of leaves, which is given by $n_{\text{max}} = l_{\text{max}} r_{\text{max}} + 1$ where l_{max} and r_{max} are the maximal VN and CN degrees. We next show that for a randomly chosen code from the ensemble LDPC(λ, ρ) the probability that the nodes in a computation graph with root VN i, $i = 1, ..., 2n$, contains only distinct VNs, and is therefore a tree, can be bound as follows.

Let us represent the VNs as two vectors $\mathbf{v}_1, \mathbf{v}_2 \in \{0,1\}^{n+\tau}$ with zero padding, i.e. $v_{1,j} = 0$ for $j \in [n+1 : n+\tau]$ and $v_{2,j} = 0$ for $j \in [0, \tau]$. Let $\mathcal{V}_u$ denote the set of VNs for user u, $u \in \{1,2\}$. Due to the same-codebook constraint, the neighborhood of $v_{1,i}$ is the same as the neighborhood of $v_{2,i+\tau}$. The neighborhood of some fixed VN, without loss of generality in $\mathcal{V}_1$, at depth t can be recursively expressed as follows. Let $\mathcal{N}^{2t}(\tau)$ denote the neighborhood of root VN i (we drop the index i for readability) at depth t in the joint graph with offset τ. We also drop the dependence on τ when immaterial. We split the neighborhood as $\mathcal{N}^{2t} = N_1^{2t} \cup N_2^{2t}$ where N_1^{2t}, N_2^{2t} denote the neighbors in $\mathcal{V}_1$ and $\mathcal{V}_2$ respectively. Let $N_1^0 = i$ be the root. We can describe the evolution of N_u^{2t}, $u = \{1,2\}$, with increasing depth as follows. Define the shifted sets $\mathcal{N} \pm \tau := \{i : i = j \pm \tau, j \in \mathcal{N}\}$.

1) $N_1^{2t+1} = N_1^{2t} \cup V^{t,1}$ where $V^{t,1}$ is the set of nodes in $\mathcal{V}_1$ that connect to N_1^{2t} through a CN.
2) $N_2^{2t+1} = N_1^{2t+1} - \tau$. The right hand side (rhs) is the set of nodes in $\mathcal{V}_2$ that connect to N^{t+1} through MAC nodes.
3) $N_2^{2t+2} = N_2^{2t+1} \cup V^{t,2}$ where $V^{t,2}$ is the set of nodes in $\mathcal{V}_2$ that connect to N_2^{2t+1} through a CN.
4) $N_1^{2t+2} = N_2^{2t+2} + \tau$. The rhs is the set of nodes in $\mathcal{V}_1$ that connect to N_2^{2t+2} through MAC nodes.

Note that for a random code from the ensemble LDPC(λ, ρ) the sets $V^{t,u}$ are random.

Lemma 3:

$$\mathbb{P}(\mathcal{N}^{2T}(\tau) \text{ is not a tree for some } \tau \in [1 : \tau_{\text{max}}]) \leq \frac{\gamma}{n}$$

where γ depends on T, λ, ρ and τ_{max} but not on n.

Proof: The proof follows the structure of [20], [23]. Assume that the computation graph at iteration t, $t < T$, is a tree.[4] We need to compute the probability that any of the four steps in the construction of the neighborhood of a VN introduces a cycle. Note, that only the sets $V^{t,u}$ are random since the MAC connections are fixed. No cycle is introduced if $V^{t,1} \cap N_1^{2t} = V^{t,2} \cap N_2^{2t+1} = \emptyset$. In addition, since we need to take the same-codebook constraint into account, we also require that

$$V^{t,u} \cap \{N_u^{2t} + \tau\} = \emptyset \quad \forall \tau \in [1 : \tau_{\text{max}}] \tag{41}$$

for $u = \{1,2\}$. Note that (28) is necessary because otherwise there would be a τ for which N_2^{2t+1} contains a node which is a mirrored copy of a node in N_1^{2t+1}. This implies that the edges connected to it are fixed and cannot be considered random iid anymore. Even though the event (41) does no necessarily result in a cycle, we treat it as such to get an upper bound on the probability that a computation graph is cycle-free. This increases the number of VNs that results in a cycle by a factor $(1+\tau_{\text{max}})$ in each iteration compared to the case without MAC connections. Intuitively it is clear that this does not change the basic proof idea of [20] since the size of a neighborhood after t iterations does still not scale with n. Nonetheless, we give a formal proof for completeness. Let c_u^T and $v_u^T = |N_u^{2T}|$ denote the number of CNs and VNs in the computation graph of user u after T iterations in $\mathcal{V}_u$. Then, at iteration $t+1$, the number of newly added CNs is at most

$$c_u^{t+1} - c_u^t \leq v_u^t l_{\text{max}} \tag{42}$$

and the number of newly added VNs is at most

$$v_u^{t+1} - v_u^t \leq c_u^{t+1} r_{\text{max}}. \tag{43}$$

Both of these quantities can be upper-bounded independently of the index $u = \{1,2\}$, so we drop it. Furthermore, $v^T \geq v^t$ and $c^T \geq c^t$ for $T \geq t$. Conditioned on the event that $\mathcal{N}^{2(T-1)}(\tau)$ is a tree for all $\tau \in [1 : \tau_{\text{max}}]$, going one step deeper will result in no cycles if the edges from the new VNs, of which there are $v^T - v^{T-1}$, meet two conditions: First, they connect to distinct, not yet visited, CNs. And second, the resulting new CNs connect to distinct VNs that are neither in the set of c^{T-1} already visited VNs nor in the same set shifted by some τ. Both copies of the graph follow the same rules and have distinct sets of VNs and CNs, so we can bound them in the same way. The resulting probability is

$$\begin{aligned} &\mathbb{P}(\mathcal{N}^{2T}(\tau) \text{ is a tree } \forall\tau | \mathcal{N}^{2(T-1)}(\tau) \text{ is a tree } \forall\tau) \\ &\geq \left(1 - \frac{(1+\tau_{\text{max}})c^T}{m}\right)^{(1+\tau_{\text{max}})(c^T - c^{T-1})} \\ &\quad \cdot \left(1 - \frac{(1+\tau_{\text{max}})v^T}{n}\right)^{(1+\tau_{\text{max}})(v^T - v^{T-1})} \end{aligned} \tag{44}$$

So we obtain recursively that

$$\begin{aligned} &\mathbb{P}(\mathcal{N}^{2T} \text{ is a tree } \forall\tau) \\ &\geq \prod_{t=1}^{T} \mathbb{P}(\mathcal{N}^{2t} \text{ is a tree } \forall\tau | \mathcal{N}^{2(t-1)} \text{ is a tree } \forall\tau) \\ &\geq \left(1 - \frac{(1+\tau_{\text{max}})c^T}{m}\right)^{(1+\tau_{\text{max}})c^T} \\ &\quad \cdot \left(1 - \frac{(1+\tau_{\text{max}})v^T}{n}\right)^{(1+\tau_{\text{max}})v^T} \\ &\geq 1 - \frac{((1+\tau_{\text{max}})v^T)^2 + \frac{((1+\tau_{\text{max}})c^T)^2}{1-R}}{n} \end{aligned} \tag{45}$$

and therefore

$$\mathbb{P}(\mathcal{N}_i^{2T} \text{ is not a tree}) \leq \frac{((1+\tau_{\text{max}})v^T)^2 + \frac{((1+\tau_{\text{max}})c^T)^2}{1-R}}{n} \tag{46}$$

We conclude the proof by giving bounds on c^T and v^T

$$c^T \leq l_{\text{max}} \sum_{t=1}^{T-1} v^t \leq l_{\text{max}}(T-1)v^{T-1} \tag{47}$$

$$v^T \leq r_{\text{max}} T c^T \leq l_{\text{max}} r_{\text{max}} T^2 v^{T-1} \tag{48}$$

[4] The first iteration is special as it has one more connection than the others, as depicted in Fig. 7. It is apparent that this does not change the proof.

which gives

$$v^T \leq (l_{\max} r_{\max} T^2)^T \tag{49}$$

$$c^T \leq l_{\max}(T-1)(l_{\max} r_{\max} T^2)^{T-1}. \tag{50}$$

We conclude the proof by noting that both upper bounds on c^T and v^T are independent of n. ■

To conclude the proof of Theorem 2 it remains to show that $\mathbb{E}_{\mathbf{d}}[P_b(\mathbf{d})]$ converges to the ensemble average over $\mathcal{G} \in$ LDPC(λ, ρ) as $n \to \infty$. We omit a full proof and give only an outline since it follows, almost without modifications, the proof in [20]. By Lemma 2 assume that the channel output is iid in the computation of $\mathbb{E}_{\mathcal{G},\mathbf{d}}[P_b(\mathbf{d})]$. By Lemma 3 we can reduce the computation of $\mathbb{E}_{\mathcal{G},\mathbf{d}}[P_b(\mathbf{d})]$ to $\mathbb{E}_{\mathcal{G}}[\mathbb{1}(v_i^l == \epsilon)|\mathcal{N}_i^l$is a tree] where $v_i^l, i = 1, ..., 2n$, denotes the value of the i-th VN after l iterations. The convergence of the edge erasure probabilities to the ensemble average can be shown by constructing an edge exposure martingale. In our case each revealed edge affects both users' graphs so the number of edges affected in the depth l neighborhood doubles. It is apparent that the martingale still has bounded increments as the number of edges in the depth l neighborhood of a given edge does not scale with n. Together with Lemma 1 and Corollary 1 this concludes the proof.

Appendix C
Details on Degree Optimization

The optimization of λ for fixed ρ can be expressed in standard form as follows.

$$\begin{aligned} g_\lambda(x) &= x - \frac{1}{2} L(z(x))\lambda(z(x)) \\ &= x - \frac{1}{2(\sum \frac{\lambda_i}{i})} \boldsymbol{\lambda}^T \mathbf{H}_x \boldsymbol{\lambda} \end{aligned} \tag{51}$$

where $H_{x,ij} = \frac{z(x)^{ij-1}}{i}$ and $z(x) = 1 - \rho(1-x)$. We get the optimization problem:

$$\begin{aligned} \max_{\kappa,\lambda} \quad & \kappa \\ \text{s.t.} \quad & \sum \frac{\lambda_i}{i} - \kappa = 0; \lambda_i \geq 0; \sum \lambda_i = 1; \\ & \boldsymbol{\lambda}^T \mathbf{H}_x \boldsymbol{\lambda} - 2\kappa(x - \delta) < 0 \ \forall x \in (0,1) \end{aligned} \tag{52}$$

Appendix D
Error Floor Analysis

Throughout this section we use the term $4K$ stopping set ($4K$-SS) to denote stopping sets of size $4K$ consisting of just degree one VNs.

Theorem 6: The probability that a random code from the ensemble LDPC(λ, ρ) results in a joint graph that has no 4-SS for all $\tau \in [1 : \tau_{\max}]$ can be bounded as

$$\begin{aligned} &\mathbb{P}\left(\mathcal{G}(\tau) \text{ has no 4-SS } \forall \tau \in [1 : \tau_{\max}]\right) \\ &\qquad \geq 1 - \tau_{\max} \frac{L_1^4}{2(1-R)^2} \end{aligned} \tag{53}$$

□

Proof: There are $\binom{L_1 n}{2} \leq n^2 L_1^2/2$ pairs of degree one VNs. Let $n_c = (1-R)n$ denote the number of CNs. The probability that a pair of VNs is connected to the same CN is $1/n_c$. Also let $\tilde{p}$ denote the probability that 4-stopping set appears that contains a given pair of VNs (v_1, v_2) in the joint graph with fixed τ. It is given by $\tilde{p} = L_1^2/n_c$, i.e., the probability that the nodes connected to (v_1, v_2) through MAC nodes are both of degree one and connect to the same CN. The degrees and edges of all $\tau_{\max}$ VNs to the right of (v_1, v_2) are independent and therefore the probability that at least one of the joint graphs with shift τ contains a 4-SS is given by $1-(1-\tilde{p})^{\tau_{\max}}$. Let N_4 denote the number of 4-SSs and I_p the event that a 4-SS goes through pair (v_1, v_2). Then the expected number of 4-SSs is given by

$$\begin{aligned} \mathbb{E}[N_4] &= \mathbb{E}\left[\sum_{p=1}^{\binom{L_1 n}{2}} I_p\right] \\ &\leq \frac{L_1^2 n_c^2}{2(1-R)^2} \frac{1}{n_c} \left(1 - \left(1 - \frac{L_1^2}{n_c}\right)^{\tau_{\max}}\right) \\ &\leq \tau_{\max} \frac{L_1^4}{2(1-R)^2} \end{aligned} \tag{54}$$

The last inequality follows because $(1-x)^\tau \geq 1 - \tau x$. If the expected number of 4-SSs is smaller than 1 there must be graphs in the ensemble that result in zero 4-SSs. Furthermore, for any non-negative random variable N it holds that $\mathbb{P}(N = 0) \geq 1 - \mathbb{E}[N]$. ■

Proof of Thm. 3: Let $k \leq K$. There are $\binom{L_1 n}{2k} \leq n^{2k} L_1^{2k}/(2k)!$ k-tuples of degree one VNs. Let $n_c = (1-R)n$ denote the number of CNs. For each $2k$-tuple there are $(2k-1)!!$ ways to partition them in pairs, where $(2k-1)!! = (2k-1)(2k-3)... \cdot 1$ denotes the double factorial. The probability that each pair is connected to the same CN is n_c^{-k}. Let $\tilde{p}_{2k}$ denote the probability that a $4k$-SS goes through a given $2k$-tuple in the joint graph with fixed τ. Note that the degrees and edges of the neighbor sequence are not independent if the original tuple of VNs contains a consecutive sequence of length at least three. We show later that their contribution to the expected number of $4k$-SSs is at most of order $\mathcal{O}(1/n^2)$ and results in the correction term in (15). For now we consider only $2k$-tuples which do not contain consecutive sequences. For those, the degrees and edges of the neighbor sequence are independent of the original tuple. A $4k$-SS is created if the $2k$-tuple connected by MAC nodes consist of only degree one VNs which connect to k CNs in a configuration that does not result in shorter SSs. With respect to random permutations of VNs and edges this happens with probability

$$\tilde{p}_{2k} \leq \frac{(2k-1)!! L_1^{2k}}{n_c^k} \tag{55}$$

Here we have trivially lower bound the configurations that result in SSs smaller than $2k$ by zero. The probability that at least one of the joint graphs with shift τ contains a $2k$-SS is given by $1-(1-\tilde{p}_{2k})^{\tau_{\max}}$. Let N_{4k} denote the number of $4k$-SSs and $I_{p,2k}$ the event that a $4k$-SS goes through the $2k$-tuple p. Then the expected number of $4k$-SSs is given by

$$\begin{aligned}\mathbb{E}[N_{4k}] &= \mathbb{E}\left[\sum_{p=1}^{\binom{L_1 n}{2k}} I_{p,2k}\right] \\ &\leq \frac{(2k-1)!!L_1^{2k}n_c^{2k}}{(2k)!(1-R)^{2k}}\frac{1}{n_c^k}\left(1-(1-\tilde{p}_{2k})^{\tau_{\max}}\right) \\ &+\mathcal{O}\left(\frac{1}{n^3}\right) \\ &\leq \tau_{\max}\left(\frac{L_1^2}{1-R}\right)^k\frac{((2k-1)!!)^2}{(2k)!}+\mathcal{O}\left(\frac{1}{n^3}\right) \\ &\leq \tau_{\max}\left(\frac{L_1^2}{1-R}\right)^k\frac{1}{2k}+\mathcal{O}\left(\frac{1}{n^2}\right)\end{aligned} \tag{56}$$

The second inequality follows because $(1-x)^\tau \geq 1-\tau x$. The expected number of SSs up to length $4K$ is $\mathbb{E}[N_{\leq 4K}] = \sum_{k=1}^{K}\mathbb{E}[N_{4k}]$. If $\mathbb{E}[N_{\leq 4K}]$ is smaller than 1 there must be graphs in the ensemble that result in zero SSs of size smaller than $4K$ because for any non-negative random variable N it holds that $\mathbb{P}(N=0)\geq 1-\mathbb{E}[N]$.

It remains to show that the number of $2k$-tuples that contain consecutive sequences is of order $\mathcal{O}(1/n^2)$. The number of length $2l+1$ sequences is at most linear in n while it reduces the probability that the neighbor sequence connects to k CNs by at most a factor of n_c^l. Therefore, the expected number of $4k$-SSs that go through at least $2l+1$ consecutive VNs can be bound loosely by a $\mathcal{O}(n/n^{2k-l})$ term. Since $l \leq k-1$ the term is maximized for $l=k-1$ and $k=2$ giving the desired result. ■

Proof of Thm. 4: The proof follows by noting that the probability of having stopping sets with VNs with degree larger than one connected to the same set of CNs will go to zero as $n\to\infty$. Indeed, the smallest possible stopping set containing degree two VNs is the one where two degree one VNs connect to the same CN, and two degree two VNs connected to the same pair of CNs. Their expected number can be upper-bounded by $\binom{L_2 n}{2}L_1^2/n_c^3 = \mathcal{O}(1/n)$ since n_c scales with n. Any larger stopping set containing degree two, or higher, VNs will have an even smaller expected number. Thus, as $n\to\infty$, we can have only stopping sets involving degree one VNs, which implies that expurgating the randomly generated graphs that contains these stopping sets guarantees a vanishing BLER as n grows. ■