LDPC Codes for Random Access Channel

Yuxin Liu, *Student Member, IEEE,* and Michelle Effros, *Fellow, IEEE*

***Abstract*—This paper derives finite-blocklength achievability bounds for low-density parity-check (LDPC) codes for the random access channel (RAC), in which neither the encoders nor the decoder perceive which and how many of the total K transmitters are active. The LDPC code design is comprised of K stages, corresponding to each of the $k \in \{1, \dots, K\}$ potential decoding times n_k. The results demonstrate that the proposed RAC LDPC codes achieve first- and second-order performance that is identical to the best-prior results for the corresponding MAC.**

I. Introduction

THIS paper extends the error-exponent and finite-blocklength analyses of low-density parity-check (LDPC) code from Point-to-Point Channels (PPC) and Multiple Access Channels (MAC) in [1] to Random Access Channels (RAC), in which neither the transmitters nor the decoder have access to the information about the set of active transmitter(s).

Random access techniques have a pivotal role in modern communication systems, as the set of transmitters may be unknown or time-varying in many applications like Internet of Things (IoT) and sensor networks. Whilst network information theory, which aims to establish the fundamental limits of information flow and corresponding coding schemes in networks, has been attracting increasingly interests in the past decades, majority of the studies assume the set of transmitters is fixed and known.

Current random access techniques focus on collision avoidance or orthogonalization, which facilitate many users to share the network medium without interference. Slotted-ALOHA (S-ALOHA) introduces discrete time slots to synchronize the start of transmission. A transmission is acknowledged to be successful if only one transmitter is active for that slot, otherwise a collision is declared. Collided packets are re-transmitted after a random delay to avoid repeated collision. Under a set of assumptions, S-ALOHA is shown to achieve 36.8% of the single-transmitter capacity [2]. Past improvements on S-ALOHA include adaptive S-ALOHA [3], [4], which addresses the instability issue of S-ALOHA, and fast adaptive S-ALOHA [5], which seeks to accelerate and algorithm and decrease the access delay. Orthogonalization methods partition available resources and allocate a fixed fraction to each user, typical orthogonalization protocols include time-division multiple access (TDMA), frequency-division multiple access (FDMA), and code-division multiple access (CDMA). The cost associated with simplifying coding using orthogonalization protocols is significant, as the best sum-rate (throughput) achievable is equal to the single-transmitter capacity of the channel, which could be considerably smaller than the maximum multi-transmitter sum-rate of the channel. Some prominent non-orthogonal multiple access (NOMA) advancements are reviewed and discussed in [6].

This paper investigates the coding schemes for communication scenarios in which no one knows the set of active transmitters. With the intention of matching the varying capacity region of a MAC with different transmitters, the designed code must be able to tailor its codebook size to modify the code rate. With the enlightenment of the definition of the code rate, $R = \frac{\log M}{n}$, we are able to adjust the code rate by tuning the decoding time, which effectively changes the blocklength of the codewords in, but not the size of, the codebook.

Rateless codes, first analyzed by Burnashev in 1976 [7], are codes that enable various decoding times. Traditionally, all transmitters in a rateless code listen for a single feedback bit from the receiver that indicates whether to continue or stop their transmissions at every time step. Error exponents of independent and identically distributed (i.i.d.) rateless codes over binary symmetric channels are analyzed in [8]. Draper et al. generalize the strategy in [8] to arbitrary discrete memoryless (DM) PPCs (DM-PPC) [9]. The dispersion-style analysis of i.i.d. rateless codes over DM-PPC is derived in [10]. The results of this study reveal that feedback can dramatically improve the maximal achievable rate in the finite-blocklength regime. Instead of allowing arbitrary decoding times and thus requiring constant feedback, recent developments in rateless codes have lifted the heavy burden on all users by employing a limited set of decoding times [11]–[13]. The set of decoding times $\{n_1, n_2, \dots, n_K\}$ is finite and predetermined, where $n_i, i \in \{1, \dots, K\}$ refers to the decoding time at which the decoder believes there are i active transmitters and it is able to decode.

Despite the challenge that the set of active transmitters is unknown and the feedback rate is asymptotically vanishing, Effros et al. [11] prove that their proposed RAC code can achieve the first- (capacity) and second-order (dispersion) terms of the corresponding MAC. For Gaussian RAC model, Yavas et al. [12] design Gaussian RAC code by concatenating K partial codewords, each of blocklength $n_1, n_2 - n_1, \dots, n_K - n_{K-1}$, and demonstrate the proposed code is able to attain the same first- (capacity), second- (dispersion), and third-order performance as the best known result for the Gaussian MAC.

While i.i.d. random codes enjoy the property of independence between symbols within and across codewords, making them theoretically simple to analyze, the mutual independence also induces difficulties in practical implementations. In contrast, linear codes, such as LDPC codes, are more structured and thus more realistic to implement.

LDPC codes, introduced by Gallager in 1962 [14], are equipped with sparse parity-check matrices for the purpose

Yuxin Liu and Michelle Effros are with the Department of Electrical Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: yuxinl@caltech.edu; effros@caltech.edu).

TABLE I
SUMMARY OF NOTATIONS

K	maximum number of active transmitters
n_i	blocklength/number of LDPC variable nodes after stage $i, i \in [K]$ design
r_i	number of LDPC check nodes after stage $i, i \in [K]$ design
$n_{\Delta i}$	increment blocklength/number of LDPC variable nodes for stage $i, i \in [K]$ design
$r_{\Delta i}$	increment number of LDPC check nodes for stage $i, i \in [K]$ design
λ_i	variable node degree of stage $i, i \in [K]$ regular LDPC code
ρ_i	check node degree of stage $i, i \in [K]$ regular LDPC code
$\mathcal{Q}$	$\mathrm{GF}(q)$
$\mathcal{G}$	bipartite LDPC graph
$\mathcal{V}$	vertex set of a graph $\mathcal{G}$
$\mathcal{E}$	edge set of a graph $\mathcal{G}$
$i(x;y)$	information density
C	channel capacity
V	channel dispersion
T	third-order centered moment of information density
Q	complementary Gaussian CDF
$\mathcal{T}_{\mathcal{Q}}^n$	set of all possible types for n elements from $\mathcal{Q}$
$\boldsymbol{v}$	LDPC coset vector
δ	LDPC quantizer
$\overline{S}^n(\boldsymbol{t})$	ensemble-average number of type-$\boldsymbol{t}$ codewords/codematrices
$\overline{S}^n$	ensemble-average spectrum
$B(n, \boldsymbol{t})$	multinomial coefficient
$E_p(R)$	Gallager's error exponent for distribution p

of enabling low complexity decoding strategies. There has been renewed interest in LDPC codes since the introduction of turbo codes in 1990s. LDPC codes are now in widespread use, playing a role in commercial standards like 10 Gb/s Ethernet (IEEE 803.3an), WiFi (IEEE 802.11n), WiMAX (IEEE 802.16e), and the 5G standard.

This paper proposes a new rateless RAC LDPC code and analyzes its performance using both the error-exponent approach and the dispersion-style approach.

The paper is organized as follows. In Section II, we define notations, followed by the channel models and code construction in Section III.

II. NOTATIONS

For non-negative integers a and b, let $[a] \triangleq \{1, 2, \ldots, a\}$, and $[a:b] \triangleq \{a, a+1, \ldots, b\}$, where $[a:b] = \emptyset$ when $a > b$. We employ uppercase letters (e.g., X and Y) to denote random variables, lowercase letters to denote their realizations (e.g., x and y), and calligraphic uppercase letters (e.g., $\mathcal{X}$ nad $\mathcal{Y}$) to indicate their sample spaces. To represent vectors, we use both superscripts (e.g., X^n) and bold fonts (e.g., $\boldsymbol{X}$, $\mathbf{1}$ and $\mathbf{0}$) when the length of the vector is clear in the context. The subscript i in X_i indicates the ith element of the vector $\boldsymbol{X}$. The inequality symbols (e.g., $>$ and $\leq$) operate element-wise when used for comparing vectors (e.g., $X^n > Y^n$ if $X_i > Y_i$ for all $i \in [n]$). Likewise, applying a scalar function $f(\cdot)$ to a vector $\boldsymbol{X} \in \mathbb{R}^n$ gives the vector of function values $f(\boldsymbol{X}) \triangleq (f(X_i), i \in [n])$. Given a scalar $a \in \mathbb{R}$, a vector $\boldsymbol{v} \in \mathbb{R}^n$, and a set $\mathcal{S} \subseteq \mathbb{R}^n$, the set $a\mathcal{S} + \boldsymbol{v}$ is defined as $a\mathcal{S} + \boldsymbol{v} \triangleq \{a\boldsymbol{s} + \boldsymbol{v}, \boldsymbol{s} \in \mathcal{S}\}$.

Throughout the paper, unless otherwise indicated, all logarithms are evaluated with base q, where the prime power q specifies the alphabet size of our proposed LDPC code. We apply standard asymptotic notations $O(\cdot)$ and $o(\cdot)$ to describe the limiting behavior of functions, writing $f(n) = O(g(n))$ if there exists constants a and n' such that $|f(n)| \leq a|g(n)|$ for all $n > n'$ and $f(n) = o(g(n))$ if $\lim_{n\to\infty} \left|\frac{f(n)}{g(n)}\right| = 0$.

For a joint distribution P_{XY} on discrete alphabet $\mathcal{X} \times \mathcal{Y}$ and any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, we denote the information density by

$$i(x;y) \triangleq \log \frac{P_{Y|X}(y|x)}{P_Y(y)}. \tag{1}$$

Given a set of alphabets $\mathcal{X}_i, i \in [n]$, a joint distribution P_{X^kY} for some integer k, and ordered sets $\mathcal{A}, \mathcal{B} \subseteq [k]$ such that $\mathcal{A} \cap \mathcal{B} = \emptyset$, we define $\mathcal{X}_{\mathcal{A}} \triangleq \prod_{i\in\mathcal{A}} \mathcal{X}_i$ and information density and conditional information density as

$$i(x_{\mathcal{A}};y) \triangleq \log \frac{P_{Y|X_{\mathcal{A}}}(y|x_{\mathcal{A}})}{P_Y(y)} \tag{2}$$

$$i(x_{\mathcal{A}};y|x_{\mathcal{B}}) \triangleq \log \frac{P_{Y|X_{\mathcal{A}},X_{\mathcal{B}}}(y|x_{\mathcal{A}},x_{\mathcal{B}})}{P_{Y|X_{\mathcal{B}}}(y|x_{\mathcal{B}})}, \tag{3}$$

for any $x_{\mathcal{A}} \in \mathcal{X}_{\mathcal{A}}, x_{\mathcal{B}} \in \mathcal{X}_{\mathcal{B}}$, and $y \in \mathcal{Y}$.

The corresponding mutual informations, dispersions, conditional dispersions, third centered moments, conditional third centered moments of the information density (2) and conditional information density (3) are

$$I(P_{X_{\mathcal{A}}}) \triangleq \mathbb{E}[i(X_{\mathcal{A}};Y)] \tag{4}$$

$$I(P_{X_{\mathcal{A}}}|P_{X_{\mathcal{B}}}) \triangleq \mathbb{E}[i(X_{\mathcal{A}};Y|X_{\mathcal{B}}))] \tag{5}$$

$$V(P_{X_{\mathcal{A}}}) \triangleq \mathrm{Var}[i(X_{\mathcal{A}};Y)] \tag{6}$$

$$V(P_{X_{\mathcal{A}}}|P_{X_{\mathcal{B}}}) \triangleq \mathrm{Var}[i(X_{\mathcal{A}};Y|X_{\mathcal{B}})] \tag{7}$$

$$V^Y(P_{X_{\mathcal{A}}}) \triangleq \mathrm{Var}[i(X_{\mathcal{A}};Y)|Y] \tag{8}$$

$$V^Y(P_{X_{\mathcal{A}}}|P_{X_{\mathcal{B}}}) \triangleq \mathrm{Var}[i(X_{\mathcal{A}};Y|X_{\mathcal{B}})|Y] \tag{9}$$

$$T(P_{X_{\mathcal{A}}}) \triangleq \mathbb{E}[|i(X_{\mathcal{A}};Y) - I(P_{X_{\mathcal{A}}})|^3] \tag{10}$$

$$T(P_{X_{\mathcal{A}}}|P_{X_{\mathcal{B}}}) \triangleq \mathbb{E}[|i(X_{\mathcal{A}};Y|X_{\mathcal{B}}) - I(P_{X_{\mathcal{A}}}|P_{X_{\mathcal{B}}})]|^3] \tag{11}$$

$$T^Y(P_{X_{\mathcal{A}}}) \triangleq \mathbb{E}[|i(X_{\mathcal{A}};Y) - I(P_{X_{\mathcal{A}}})|^3|Y] \tag{12}$$

$$T^Y(P_{X_{\mathcal{A}}}|P_{X_{\mathcal{B}}}) \triangleq \mathbb{E}[|i(X_{\mathcal{A}};Y|X_{\mathcal{B}}) - I(P_{X_{\mathcal{A}}}|P_{X_{\mathcal{B}}})]|^3|Y]. \tag{13}$$

The (probability density function) PDF and cumulative distribution function (CDF) for standard Gaussian distribution $\mathcal{N}(0,1)$ are denoted by

$$\phi(x) \triangleq \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{14}$$

$$\Phi(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{u^2}{2}} du, \tag{15}$$

respectively. The function $Q(\cdot)$ denotes the standard Gaussian complementary CDF

$$Q(x) \triangleq 1 - \Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{\infty} e^{-\frac{u^2}{2}} du, \tag{16}$$

and $Q^{-1}(\cdot)$ is the inverse function of $Q(\cdot)$.

The generalization of inverse Gaussian complementary CDF $Q^{-1}(\cdot)$ to higher dimension is denoted by $Q_{\mathrm{inv}}(\cdot,\cdot)$. Let $\boldsymbol{Z}$

be a Gaussian random vector in $\mathbb{R}^d$ with mean zero and covariance matrix K, the set $Q_{\rm inv}(\mathsf{K},\epsilon)$ is given by

$$Q_{\rm inv}(\mathsf{K},\epsilon) \stackrel{\triangle}{=} \{\boldsymbol{z}\in\mathbb{R}^d : \Pr[\boldsymbol{Z}\le\boldsymbol{z}]\ge 1-\epsilon\}. \tag{17}$$

Given a positive integer n, a prime power q, and a PDF $\boldsymbol{\theta}=(\theta_0,\theta_1,\ldots,\theta_{q-1})$ such that $n\theta_i\in\mathbb{Z},\forall i\in\{0\}\cup[q-1]$, the multinomial coefficient $B(n,n\boldsymbol{\theta})$ is defined as

$$B(n,n\boldsymbol{\theta}) \stackrel{\triangle}{=} \frac{n!}{\prod i=0^{q-1}(n\theta_i)!}. \tag{18}$$

III. System Model and Design

A. Random Access Channel Model

We begin with the definition of a K-user DM-MAC, denoted by DM-K-MAC.

Definition 1. *(DM-K-MAC) A DM-K-MAC is defined by*

$$\left(\prod_{i=1}^{K}\mathcal{X}_i, P_{Y|\boldsymbol{X}}, \mathcal{Y}\right)$$

where $\mathcal{X}_i, i\in[K]$, and $\mathcal{Y}$ are the discrete channel input and output alphabets, respectively, and $P_{Y|\boldsymbol{X}}=P_{Y|X_1,X_2,\ldots,X_K}$ is the channel transition probability..

Note that a DM-K-MAC is called **symmetric** if all transmitters share the same input alphabet $\mathcal{X}_i=\mathcal{X}$ for all $i\in[K]$ and

$$P_{Y|\boldsymbol{X}}(y|\boldsymbol{x}) = P_{Y|\boldsymbol{X}}(y|\pi(\boldsymbol{x}))$$

for all $y\in\mathcal{Y}$, $\boldsymbol{x}\in\mathcal{X}^K$, and permutations π on $[K]$.

All the MAC models in this study, unless otherwise specified, are symmetric. We regard the RAC model as a family of all possible MACs resulting from different transmitter patterns.

Definition 2. *A DM-RAC is described by a collection of symmetric DM-MACs*

$$\left\{\left(\mathcal{X}^k, P_{Y|X^k}, \mathcal{Y}_k\right)\right\}_{k=0}^{K}, \tag{19}$$

where $\mathcal{X}^k, P_{Y|X^k}$, and $\mathcal{Y}_k$ are the channel input alphabet, channel transition probability, and channel output alphabet when there are k active transmitters, respectively.

B. Quantized Coset RAC LDPC Code Design

The RAC LDPC code design is an extension of the MAC design from our previous work. For any prime power q and finite field $\mathrm{GF}(q)$, a quantized coset RAC $\mathrm{GF}(q)$-LDPC code consists of five components: a RAC LDPC encoder, a random codeword selector (fixing operational rate to design rate), an expurgator (removing codes with small minimum distances) a coset vector $\boldsymbol{v}$, and a quantizer δ, defined below and illustrated in Figure 1.

Definition 3. *(RAC $\mathrm{GF}(q)$-LDPC code) For a K-user RAC, a* **RAC $\mathrm{GF}(q)$-LDPC code** *is defined on a bipartite Tanner graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$ with n_K variable nodes, r_K check nodes, and edge set $\mathcal{E}\subseteq[n_K]\times[r_K]$ using a K-stage design.*

For each stage $i\in[K]$, the code is constructed as follows:

- *When $i=1$, the LDPC code is a length-n_1 regular LDPC code with variable node degree (left degree) λ_1 and check node degree (right degree) ρ_1. The number of check nodes r_1 satisfies $n_1\lambda_1=r_1\rho_1$.*
- *When $i=2$, there are $n_{\Delta 2}\stackrel{\triangle}{=}n_2-n_1$ additional variable nodes and $r_{\Delta 2}\stackrel{\triangle}{=}r_2-r_1$ additional check nodes. All the n_2 variable nodes are added ρ_2 sockets each, which are connected to the additional $r_{\Delta 2}$ check nodes of degree λ_2. The number of check nodes $r_{\Delta 2}$ satisfies $n_2\lambda_2=r_{\Delta 2}\rho_2$.*
- *When $3\le i<K$, $n_{\Delta i}\stackrel{\triangle}{=}n_i-n_{i-1}$ additional variable nodes and $r_{\Delta i}\stackrel{\triangle}{=}r_i-r_{i-1}$ are added. λ_i additional sockets are added to each of the n_i variable node, which are connected with a random permutation to the $r_{\Delta i}$ check nodes of degree ρ_i.The number of check nodes $r_{\Delta i}$ satisfies $n_i\lambda_i=r_{\Delta i}\rho_i$.*

For each $(i,j)\in\mathcal{E}$, (i,j) represents an undirected edge connecting the ith variable node and the jth check node; each edge value $g_{i,j}$ is chosen uniformly at random from $\mathrm{GF}(q)\setminus\{0\}$. The notation

$$\mathcal{N}(j) \stackrel{\triangle}{=} \{i:(i,j)\in\mathcal{E}\},$$

captures the neighborhood of check node $j\in[r]$ resulting from edge set $\mathcal{E}$.

The n_K variable nodes hold a length-n_K vector $\boldsymbol{u}$ from $\mathrm{GF}(q)^{n_K}$. Vector $\boldsymbol{u}$ is a **codeword** *if it satisfies all the r_K check nodes, giving*

$$\sum_{i\in\mathcal{N}(j)} g_{i,j}u_i = 0 \quad \forall j\in[r_K]; \tag{20}$$

the linear equation operates in $\mathrm{GF}(q)$. The set of all $M=|\boldsymbol{c}|$ codewords constitute the **codebook**

$$\boldsymbol{c}=\{\boldsymbol{c}_1,\ldots,\boldsymbol{c}_M\}\subseteq\mathrm{GF}(q)^{n_K}$$

for the given Tanner graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$. All the K transmitters employ the same RAC LDPC codebook.

We consider a random ensemble of RAC $\mathrm{GF}(q)$-LDPC codes, where each code in the ensemble is chosen uniformly at random from all possible permutations of connecting the $\lambda_i n_i, i\in[K]$ sockets, and all possible edge values from $\mathrm{GF}(q)\setminus\{0\}$ for each socket during the stage-i design.

The design is illustrated in Figure[reference] below

We have the following relationships for the design parameters:

$$n_k=\sum_{i=1}^{k}n_{\Delta i} \tag{21}$$

$$r_k=\sum_{i=1}^{k}r_{\Delta i} \tag{22}$$

Note that after we choose parameters n_k,λ_k,ρ_k, that will automatically determine the corresponding number of check nodes r_k. For example, $r_1=\frac{n_1\lambda_1}{\rho_1}$, $r_{\Delta 2}=\frac{n_2\lambda_2}{\rho_2}$, and in general

$$r_{\Delta k}=\frac{n_k\lambda_k}{\rho_k}. \tag{23}$$

Therefore, the design rates, in q-ary symbols per channel use, for each blocklength $n_i, i\in[K]$ is

$$R_1=\frac{n_1-r_1}{n_1}=1-\frac{\lambda_1}{\rho_1} \tag{24}$$

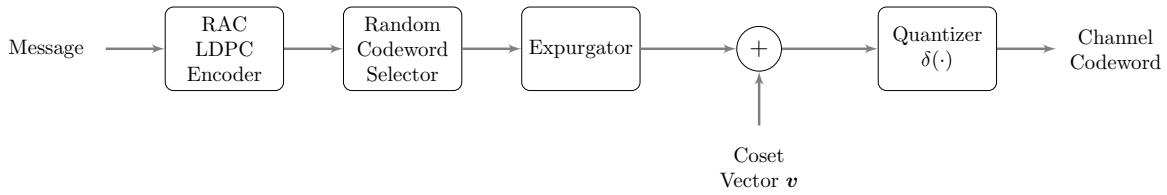

Fig. 1. Encoding of Quantized Coset LDPC Code

Fig. 2. Place holder for RAC LDPC design

$$R_2 = \frac{n_2 - r_2}{n_2} = 1 - \frac{r_2}{n_2} = 1 - \frac{\overline{\lambda}}{\overline{\rho}} = 1 - \frac{\frac{n_1\lambda_1 + n_2\lambda_2}{n_2}}{\frac{r_1\rho_1 + r_{\Delta 2}\rho_2}{r_2}} \quad (25)$$

$$= 1 - \frac{\frac{n_1}{n_2}\lambda_1 + \lambda_2}{\frac{r_1}{r_2}\rho_1 + \frac{r_{\Delta 2}}{r_2}\rho_2} \quad (26)$$

$$\vdots$$

$$R_K = \frac{n_K - r_K}{n_K} = 1 - \frac{r_K}{n_K} \quad (27)$$

$$= 1 - \frac{\sum_{i=1}^{K} \frac{n_i}{n_K}\lambda_i}{\sum_{i=1}^{K} \frac{r_{\Delta i}}{r_K}\rho_i}. \quad (28)$$

Remark 1. *The fact that $R_i, i \in [K]$ are named "design rates" is deliberate, as each of the first $r_i, i \in [K]$ check node equations in* (20) *might be linearly dependent, making the actual rate larger. The vast majority of LDPC literature assumes that the operational rate is equal to the design rate, which may not be precise for short blocklengths. Measson et al. proved in [15, Lemma 7] that the probability of the actual rate of an irregular PPC LDPC code deviating from the design rate decays exponentially in the blocklength, and this proof was generalized to regular MAC LDPC codes in [1, Theorem 3].*

To fix the operational rate to the design rate, we apply the random codeword selector.

Definition 4. *(Random Codeword Selector) Given the ensemble of RAC $\mathrm{GF}(q)$-LDPC codes with design rates vector $\boldsymbol{R} = (R_1, \ldots, R_K)$ from the RAC LDPC encoder, the random codeword selector generates an ensemble by dividing the probability of each code in the original ensemble equally among all code(s) corresponding to a distinct combination of q^{nR_i} codewords of length-n_i for all $i \in [K]$ from the original code.*

Since the performance of an ensemble of LDPC codes is negatively affected by a small number of codes with small minimum distance [16, Observation below (7)], removing (expurgating) these codes produces an ensemble with improved properties [16, Lemma 1], [1, Appendix C]. Therefore an expurgator is employed to remove codes with small minimum distances in the ensemble from the random codeword selector.

Definition 5. *(Expurgator) Let $P_L(\boldsymbol{C})$ denotes the probability of a randomly chosen code from the ensemble resulting from the random codeword selector. The expurgator generates an ensemble, by placing probability zero on all codes whose minimum distances for the first $n_i, i \in [K]$ symbols (denoted by $d_{\min}^{n_i}(\boldsymbol{C})$ is less than or equal to $\sigma_i n_i$, and probability $P_L(\boldsymbol{C}|d_{\min}^{n_i}(\boldsymbol{C}) > \sigma_i n_i, \forall i \in [K])$. The values of $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_K)$ can be chosen such that $\Pr[d_{\min}^{n_i}(\boldsymbol{C})] = O(n_i^{\bar{\lambda}_i/2-1})$, where $\bar{\lambda}_i \triangleq \sum_{j=1}^{i} \frac{n_j}{n_i}\lambda_j$ for all $i \in [K]$ [1, Appendix C].*

Similar to the MAC LDPC codes in [1], we apply quantized coset coding [17] before transmitting the codewords over the channel.

Definition 6. *(Coset $\mathrm{GF}(q)$-LDPC Code) Given the Tanner graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of a RAC LDPC encoder and the corresponding LDPC codebook $\boldsymbol{c}$ from the expurgator, the coset LDPC code is constructed by adding a constant vector $\boldsymbol{v}$ chosen uniformly at random from $\mathrm{GF}(q)^{n_K}$, called the* **coset vector***, to each codeword $\boldsymbol{c}_i \in \boldsymbol{c}$. The addition*

$$\boldsymbol{c}_i + \boldsymbol{v}, i \in [M]$$

is performed component-wise in $\mathrm{GF}(q)$. The set $\{\boldsymbol{c}_i + \boldsymbol{v}, i \in [M]\}$ is the codebook for the **coset $\mathrm{GF}(q)$-LDPC code***. Each of the K transmitters is given an independently generated coset vector $\boldsymbol{v}_i, i \in [K]$.*

Definition 7. *(Quantized Coset $\mathrm{GF}(q)$-LDPC Code) Given a coset LDPC codebook $\{\boldsymbol{c}_i + \boldsymbol{v}, i \in [M]\}$, we map each symbol from each codeword $\boldsymbol{c}_i + \boldsymbol{v}$ to a symbol from the channel input alphabet $\mathcal{X}$ using a* **quantizer** *δ:*

$$\delta : \mathrm{GF}(q) \to \mathcal{X}. \quad (29)$$

Mapping δ is applied component-wise; we therefore employ notation

$$\delta(\boldsymbol{c}_i + \boldsymbol{v}) \triangleq [\delta((\boldsymbol{c}_i + \boldsymbol{v})_j)]_{j \in [n]}$$

for coset codeword $\boldsymbol{c}_i + \boldsymbol{v}$. The set $\{\delta(\boldsymbol{c}_i + \boldsymbol{v}), i \in [M]\}$ is the codebook for the **quantized coset $\mathrm{GF}(q)$-LDPC code***. All the K transmitters adopt the same quantizer $\delta(\cdot)$.*

Remark 2. *Even all the k-MAC, $k \in [K]$, in the RAC model (Def. 2) are assumed to be symmetric, the capacity-achieving distribution P_X^* might vary with the number of active transmitters [11, Example 2]. In such scenarios, [11, Sec. V.B] discusses how to choose the optimal distributions using dominant rate points, which in turn guides the design of the quantizer $\delta(\cdot)$.*

We label the random ensemble resulting from the application of random RAC LDPC encoder, random codeword selector, and expurgator as $\mathrm{LDPC} - \{(\lambda_i, \rho_i; n_i)\}_{i=1}^{K}$. Before

transmission starts, the chosen RAC LDPC codebook, coset vector, and quantizer are disclosed to all parties.

C. Decoder Design

Follow the RAC communication protocol proposed in [11, Sec. II.B], transmissions happen in epochs, where each new epoch starts after a positive acknowledgement bit sent from the decoder at some time $n_i, i \in [K] \cup \{0\}$ of the previous epoch.

The decoder is a two-stage process and it sends a positive feedback bit (acknowledgement) to all transmitters if there is no active transmitter or after the second stage at the end of each epoch:

1) The first stage is a hypothesis test with $K+1$ testing functions $h_i(\cdot)$ with $K+1$ thresholds $\gamma_i, i \in [K] \cup \{0\}$ that determines whether there are any active transmitters at time n_0, and if so, how many active transmitters are there at time $n_i, i \in [K]$. If the hypothesis test threshold if fulfilled at time n_0, then the decoder declares there are no active transmitters for the current epoch and a positive acknowledgement is sent to signal the start of next epoch. Otherwise, the decoder enters the second stage upon passing the first testing function $h_k(\cdot)$ at time n_k for some $k \in [K]$.
2) The second stage is a maximum likelihood (ML) decoder with K decoding functions $g_k(\cdot), k \in [K]$ that maps the received symbols $y^{n_k} \in \mathcal{Y}_k^{n_k}$ to $(\mathcal{X}^k)^{n_k}$. A positive feedback bit is sent at time n_k to conclude the current epoch, and a new epoch starts at the next time step. Each transmitter then decides whether to start a new transmission or remain silent for the new epoch, until the next positive feedback bit is received.

IV. Error Analyses

We present two types of analyses, error-exponent analysis and dispersion-style analysis, for the proposed Quantized Coset RAC LDPC code in this section. As discussed in [1, Sec. III-C], a comparison of both approaches reveals that the error-exponent analysis achieves a sub-optimal bound at small blocklength n but a superior performance when target error probability ϵ is small.

Before presenting the analyses, we define an important quantity, ensemble-average number of codewords, that is essential in both approaches.

Denote the **type** of a vector $\boldsymbol{u} \in \mathrm{GF}(q)^n$ as $\mathcal{T}_{\mathcal{Q}}^n(\boldsymbol{u})$. The set of all possible types of a length-n_i vector from $\mathrm{GF}(q)^{n_i}, i \in [K]$ is denoted by

$$\mathcal{T}_{\mathcal{Q}}^{n_i} \triangleq \{\mathcal{T}_{\mathcal{Q}}^{n_i}(\boldsymbol{u}) : \boldsymbol{u} \in \mathrm{GF}(q)^{n_i}\} \subset \mathbb{Z}_+^{|\mathcal{Q}|}.$$

For a given codebook $\boldsymbol{c}$ from the $\mathrm{LDPC}-\{(\lambda_i, \rho_i; n_i)\}_{i=1}^K$ ensemble, the number of type-$n_i\boldsymbol{\theta}_i$, $n_i\boldsymbol{\theta}_i \in \mathcal{T}_{\mathcal{Q}}^{n_i}$ codewords in $\boldsymbol{c}$ is denoted by $S_{\boldsymbol{c}}^{n_i}(\boldsymbol{\theta}_i)$

$$S_{\boldsymbol{c}}^{n_i}(\boldsymbol{\theta}_i) = \sum_{m=1}^{q^{n_i R_i}} \mathbb{1}\{\mathcal{T}_{\mathcal{Q}}^{n_i}(\boldsymbol{c}_m) = n_i\boldsymbol{\theta}_i\} \tag{30}$$

We represent the **ensemble-average spectrum** of the $\mathrm{LDPC}-\{(\lambda_i, \rho_i; n_i)\}_{i=1}^K$ ensemble at each blocklength n_i (for which the total number of check nodes is r_i) by

$$\overline{\boldsymbol{S}}_{L,r_i}^{n_i} \triangleq \mathbb{E}_{\boldsymbol{C}}[\boldsymbol{S}_{\boldsymbol{C}}^{n_i}] = \{\overline{S}_{L,r_i}^{n_i}(\boldsymbol{\theta}_i) : n_i\boldsymbol{\theta}_i \in \mathcal{T}_{\mathcal{Q}}^{n_i}\}, \forall i \in [K]. \tag{31}$$

A. Error-Exponent Analysis

Given a RAC model $\left\{\left(\mathcal{X}^k, P_{Y|X^k}, \mathcal{Y}_k\right)\right\}_{k=0}^K$, where a single distribution P_X^* achieves the capacity for different number of active transmitters (see Remark 2 for discussions on when such distribution does not exist), Theorem 1 bounds the ensemble-average probability of the proposed RAC LDPC ensemble.

Theorem 1. *Consider $\left\{\left(\mathcal{X}^k, P_{Y|X^k}, \mathcal{Y}_k\right)\right\}_{k=0}^K$. Let the RAC's maximal symmetrical rate vector be the K-vector $(C, \ldots, C)$, and fix any $\boldsymbol{R} = (R, \ldots, R)$ with $R < C$. Let P_U be a pmf on $\mathcal{U}$ for which $P_U(u) = N_u/q$ for some integer N_u for each $u \in \mathcal{U}$, and let $\delta : \mathrm{GF}(q) \to \mathcal{U}$ be a quantization matched to P_U. The ensemble-average error probability of the quantized coset-shifted of the $\mathrm{LDPC} - \{(\lambda_i, \rho_i; n_i)\}_{i=1}^K$ satisfies*

Similar to the error analysis presented in [1, Thm. 1], we obtain

$$\frac{\log \alpha_i}{n_i} = \max \boldsymbol{\theta}_i : \boldsymbol{\theta}_i n_i \in \mathcal{T}_{\mathcal{Q}}^{n_i} \frac{\overline{S}_L^{n_i}(\boldsymbol{\theta})}{(M_i - 1)B(n_i, n_i\boldsymbol{\theta})q^{-n_i}}, \quad i \in [K], \tag{32}$$

where $\overline{S}_L^{n_i}(\boldsymbol{\theta})$ is the expected number of codematrices of length n_i and type $n_i\boldsymbol{\theta}$, $M_i = q^{nR_i}$ is the number of codewords for each transmitter at length n_i, R_i is the design rate at length n_i, and $B(n_i, n_i\boldsymbol{\theta})$ is the multinomial coefficient.

B. Calculation of $\overline{S}_L^{n_i}$

To calculate $\overline{S}_L^{n_i}$ for each type $n_i\boldsymbol{\theta}$

$$\overline{S}_L^{n_i}(\boldsymbol{\theta}) = B(n_i, n_i\boldsymbol{\theta}) \Pr[d_{n_i\boldsymbol{\theta}} \in \boldsymbol{D}], \tag{33}$$

where $\Pr[d_{n_i\boldsymbol{\theta}} \in \boldsymbol{D}]$ is the probability that a matrices of type $n_i\boldsymbol{\theta}$ is in the codebook of a randomly chosen code $\boldsymbol{D}$. We then use symmetry of code design and generating functions to evaluate $\Pr[d_{n_i\boldsymbol{\theta}} \in \boldsymbol{D}]$,

$$\Pr[d_{n_i\boldsymbol{\theta}} \in \boldsymbol{D}] = \frac{\lfloor((A_{<i>}(\boldsymbol{x}))^{r_i})\rfloor_{n_i\lambda_{<i>}\boldsymbol{\theta}}}{B(n_i\lambda_{<i>}, n_i\lambda_{<i>}\boldsymbol{\theta})(q-1)^{n_i\lambda_{<i>}}}, \tag{34}$$

where $n_i\lambda_{<i>} \triangleq \sum_{j=1}^i n_j\lambda_j$ is the total number of sockets at length n_i, $(A_{<i>}(\boldsymbol{x}))^{r_i}$ is the generating function of number of codematrices after stage-i design, and $\lfloor((A_{<i>}(\boldsymbol{x}))^{r_i})\rfloor_{n_{<i>}\lambda_{<i>}\boldsymbol{\theta}}$ denotes the coefficient of the term $\boldsymbol{x}^{n_i\lambda_{<i>}\boldsymbol{\theta}}$ in the generating function.

To illustrate how to find the expression of the generating function, we take $i = 2$ for notational simplicity. A generic expression for arbitrary $i \in [K]$ is also provided below.

$$A_{<2>}(\boldsymbol{x})^{r_2} \triangleq A_1(\boldsymbol{x}_1)^{r_1} A_2(\boldsymbol{x}_{2,1}, \boldsymbol{x}_{2,2})^{r_{\Delta 2}}, \tag{35}$$

$$\lfloor((A_{<i>}(\boldsymbol{x}))^{r_i})\rfloor_{n_{<i>}\lambda_{<i>}\boldsymbol{\theta}} \triangleq \lfloor A_1(\boldsymbol{x}_1)^{r_1} A_2(\boldsymbol{x}_{2,1}, \boldsymbol{x}_{2,2})^{r_{\Delta 2}}\rfloor_{n_1\lambda_1\boldsymbol{\theta}_1 | n_1\lambda_2\boldsymbol{\theta}_1} \tag{36}$$

where $A_1(\boldsymbol{x}_1)$ is the generating function for each of the first r_1 check nodes, $A_2(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is the generating function for each of $r_{\Delta 2}$ check nodes in the second group, $\boldsymbol{\theta}_1$ is the sub-type of the first n_1 variable nodes, $\boldsymbol{\theta}_{\Delta 2}$ is the sub-type of the $n_{\Delta 2}$ variable nodes in the second group, $n_1\lambda_1\boldsymbol{\theta}_1$ refers to the term $\boldsymbol{x}_1^{n_1\lambda_1\boldsymbol{\theta}_1}$ in $A_1(\boldsymbol{x}_1)$, and finally $n_1\lambda_2\boldsymbol{\theta}_1, n_{\Delta 2}\lambda_2\boldsymbol{\theta}_{\Delta 2}$ denotes the term $\boldsymbol{x}_{2,1}^{n_1\lambda_2\boldsymbol{\theta}_1}\boldsymbol{x}_{2,2}^{n_{\Delta 2}\lambda_2\boldsymbol{\theta}_{\Delta 2}}$ in $A_2(\boldsymbol{x}_{2,1}, \boldsymbol{x}_{2,2})$.

The calculation of $A_1(\boldsymbol{x}_1)$ is as follows:

$$A_1(\boldsymbol{x}_1) = \sum_{g_1,\ldots,g_{\rho_1}\in\mathcal{Q}_1^n} \sum_{e_1,\ldots,e_{\rho_1}\in\mathcal{Q}_1^n} \mathbb{1}\left\{\sum_{i=1}^{\rho_1} g_i e_i = 0\right\}\left\{\prod \boldsymbol{x}_{1,g_i}\right\}, \tag{37}$$

where g_i and e_i correspond to the variable node value and socket value, respectively.

However, the calculation of $A_2(\boldsymbol{x}_{2,1}, \boldsymbol{x}_{2,2})$ might be difficult as we need to separate it into different cases, depending on how many of the ρ_2 sockets are from the first n_1 variables nodes (corresponding to $\boldsymbol{x}_{2,1}$ term) and how many of them are from the $n_{\Delta 2}$ variable nodes (corresponding to $\boldsymbol{x}_{2,2}$ term). That is, in the last term $\{\prod \boldsymbol{x}_{1,g_i}\}$ for $A_1(\boldsymbol{x}_1)$, g_i may correspond to $\boldsymbol{x}_{2,1}$ or $\boldsymbol{x}_{2,2}$.

To solve this problem, we make two claims.

Claim 1. *The joint generating function* $\lfloor A_1(\boldsymbol{x}_1)^{r_1} A_2(\boldsymbol{x}_{2,1}, \boldsymbol{x}_{2,2})^{r_{\Delta 2}}\rfloor_{n_1\lambda_1\boldsymbol{\theta}_1|n_1\lambda_2\boldsymbol{\theta}_1, n_{\Delta 2}\lambda_2\boldsymbol{\theta}_{\Delta 2}}$ *can be broken into two independent generating functions as*

$$\begin{aligned}&\lfloor A_1(\boldsymbol{x}_1)^{r_1} A_2(\boldsymbol{x}_{2,1}, \boldsymbol{x}_{2,2})^{r_{\Delta 2}}\rfloor_{n_1\lambda_1\boldsymbol{\theta}_1|n_1\lambda_2\boldsymbol{\theta}_1, n_{\Delta 2}\lambda_2\boldsymbol{\theta}_{\Delta 2}}\\ &= \lfloor A_1(\boldsymbol{x}_1)^{r_1}\rfloor_{n_1\lambda_1\boldsymbol{\theta}_1}\lfloor A_2(\boldsymbol{x}_{2,1}, \boldsymbol{x}_{2,2})^{r_{\Delta 2}}\rfloor_{n_1\lambda_2\boldsymbol{\theta}_1, n_{\Delta 2}\lambda_2\boldsymbol{\theta}_{\Delta 2}}.\end{aligned} \tag{38}$$

The intuition behind Claim 1 is that the sockets for the first group of $r_{\Delta 1}$ check nodes are independently drawn of the sockets for the second group of $r_{\Delta 2}$ check nods. In other words, for any given sockets configuration that are satisfied with the first group of check nodes, we can choose $A_2(\boldsymbol{x}_{2,1}, \boldsymbol{x}_{2,2})^{r_{\Delta 2}}\rfloor_{n_1\lambda_2\boldsymbol{\theta}_1, n_{\Delta 2}\lambda_2\boldsymbol{\theta}_{\Delta 2}}$ different socket connections for the second group of check nodes, and all of them are satisfied with both group of check nodes and thus give valid codematrices.

Claim 2. *Given a sub-type pair $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_{\Delta 2})$, let $\boldsymbol{\theta}_2 = n_1\boldsymbol{\theta}_{\Delta 1} + n_{\Delta 2}\boldsymbol{\theta}_{\Delta 2}$, then*

$$\begin{aligned}&\lfloor A_2(\boldsymbol{x}_{2,1}, \boldsymbol{x}_{2,2})^{r_{\Delta 2}}\rfloor_{n_1\lambda_2\boldsymbol{\theta}_1, n_{\Delta 2}\lambda_2\boldsymbol{\theta}_{\Delta 2}}\\ &= \lfloor A_2(\boldsymbol{x}_2)^{r_{\Delta 2}}\rfloor_{n_2\lambda_2\boldsymbol{\theta}_2}\cdot\frac{\binom{n_1}{n_1\boldsymbol{\theta}_1}\binom{n_{\Delta 2}}{n_{\Delta 2}\boldsymbol{\theta}_{\Delta 2}}}{\binom{n_2}{n_2\boldsymbol{\theta}_2}},\end{aligned} \tag{39}$$

where $A_2(\boldsymbol{x}_2)$ is the generating function for one check node in a regular LDPCcode with parameter (λ_2, ρ_2), and we are taking the coefficient of the term $\boldsymbol{x}_2^{n_2\lambda_2\boldsymbol{\theta}_2}$ in the generating function $A_2(\boldsymbol{x}_2)^{r_{\Delta 2}}$.

The above claims can be generalized to arbitrary $k \geq 2$, as shown in the following corollaries.

Corollary 1. *The combined generating function $\lfloor A_1(\boldsymbol{x}_1)^{r_1}\ldots A_k(\boldsymbol{x}_{k,1},\ldots,\boldsymbol{x}_{k,k})^{r_{\Delta k}}\rfloor_{n_1\lambda_1\boldsymbol{\theta}_1|\ldots|n_1\lambda_k\boldsymbol{\theta}_1,\ldots,n_{\Delta k}\lambda_k\boldsymbol{\theta}_{\Delta k}}$ can be broken into k independent generating functions as*

$$\begin{aligned}&\lfloor A_1(\boldsymbol{x}_1)^{r_1}\ldots A_k(\boldsymbol{x}_{k,1},\ldots,\boldsymbol{x}_{k,k})^{r_{\Delta k}}\rfloor_{n_1\lambda_1\boldsymbol{\theta}_1|\ldots|n_1\lambda_k\boldsymbol{\theta}_1,\ldots,n_{\Delta k}\lambda_k\boldsymbol{\theta}_{\Delta k}}\\ &= \lfloor A_1(\boldsymbol{x}_1)^{r_1}\rfloor_{n_1\lambda_1\boldsymbol{\theta}_1}\cdot\prod_{i=2}^{k}\lfloor A_k(\boldsymbol{x}_{i,1},\ldots,\boldsymbol{x}_{i,i})^{r_{\Delta i}}\rfloor_{n_1\lambda_i\boldsymbol{\theta}_1,\ldots,n_{\Delta i}\lambda_i\boldsymbol{\theta}_{\Delta i}}.\end{aligned} \tag{40}$$

Corollary 2. *Given a sub-type pair $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_{\Delta 2},\ldots,\boldsymbol{\theta}_{\Delta_k})$, let $\boldsymbol{\theta}_k = \frac{\sum_{i=1}^{k} n_{\Delta i}\boldsymbol{\theta}_{\Delta i}}{n_k}$, then*

$$\begin{aligned}&\lfloor A_k(\boldsymbol{x}_{k,1},\ldots,\boldsymbol{x}_{k,k})^{r_{\Delta k}}\rfloor_{n_1\lambda_k\boldsymbol{\theta}_1,\ldots,n_{\Delta k}\lambda_k\boldsymbol{\theta}_{\Delta k}}\\ &= \lfloor A_k(\boldsymbol{x}_k)^{r_{\Delta k}}\rfloor_{n_k\lambda_k\boldsymbol{\theta}_k}\cdot\frac{\prod_{i=1}^{k}\binom{n_{\Delta i}}{n_{\Delta i}\boldsymbol{\theta}_{\Delta i}}}{\binom{n_k}{n_k\boldsymbol{\theta}_k}},\end{aligned} \tag{41}$$

where $A_k(\boldsymbol{x}_k)$ is the generating function for one check node in a regular LDPCcode with parameter (λ_k, ρ_k), and we are taking the coefficient of the term $\boldsymbol{x}_k^{n_k\lambda_k\boldsymbol{\theta}_k}$ in the generating function $A_k(\boldsymbol{x}_2)^{r_{\Delta k}}$.

For the ease of further analysis, the following analysis assumes there are number of $K = 2$ active transmitters. Given a type $\boldsymbol{\theta}_2 = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_{\Delta 2})$, denote the number of ensemble average codewords for the first group of r_1 and second group of $r_{\Delta 2}$ check nodes as:

$$\overline{S}_{L,r_{\Delta 1}}^{n_1}(\boldsymbol{\theta}_1) = B(n_1, n_1\boldsymbol{\theta}_1)\frac{\lfloor A_1(\boldsymbol{x}_1)^{r_1}\rfloor_{n_1\lambda_1\boldsymbol{\theta}_1}}{B(n_1\lambda_1, n_1\lambda_1\boldsymbol{\theta}_1)(q-1)^{n_1\lambda_1}} \tag{42}$$

$$\begin{aligned}\overline{S}_{L,r_{\Delta 2}}^{n_2}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_{\Delta 2}) &= B(n_2, n_2\boldsymbol{\theta}_2)\cdot\\ &\frac{\lfloor A_2(\boldsymbol{x}_{2,1}, \boldsymbol{x}_{2,2})^{r_{\Delta 2}}\rfloor_{n_1\lambda_1\boldsymbol{\theta}_1, n_{\Delta 2}\lambda_2\boldsymbol{\theta}_{\Delta 2}}}{B(n_2\lambda_2, n_2\lambda_2\boldsymbol{\theta}_2)(q-1)^{n_2\lambda_2}}.\end{aligned} \tag{43}$$

Denote the number of codematrices for the combined $r_1 + r_{\Delta 2}$ check nodes as $\overline{S}_{L,r_2}^{n_2}(\boldsymbol{\theta}_2)$, then

$$\overline{S}_{L,r_2}^{n_2}(\boldsymbol{\theta}_2) = \sum_{\boldsymbol{\theta}_1\in\mathcal{C}(\boldsymbol{\theta}_1(\boldsymbol{\theta}_2, n_1))} \overline{S}_{L,r_{\Delta 1}}^{n_1}(\boldsymbol{\theta}_1)\cdot\overline{S}_{L,r_{\Delta 2}}^{n_2}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_{\Delta 2})\cdot\frac{\binom{n_1}{n_1\boldsymbol{\theta}_1}\binom{n_{\Delta 2}}{n_{\Delta 2}\boldsymbol{\theta}_{\Delta}}}{\binom{n_2}{n_2\boldsymbol{\theta}_2}} \tag{44}$$

where $\mathcal{C}(\boldsymbol{\theta}_1(\boldsymbol{\theta}_2, n_1))$ denotes the set of "compatible" $\boldsymbol{\theta}_1$ of length n_1 that can rise from the given $\boldsymbol{\theta}_2$.

Some attempts to evaluate $\overline{S}_{L,r_2}^{n_2}(\boldsymbol{\theta}_2)$ in (44):

1) Since $\overline{S}_{L,r_{\Delta 2}}^{n_2}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_{\Delta 2})$ is independent of $\boldsymbol{\theta}_1$ (note that $\boldsymbol{\theta}_1$ is in the $\boldsymbol{\theta}_2$-compatible set $\mathcal{C}(\boldsymbol{\theta}_1(\boldsymbol{\theta}_2, n_1))$), we can factor it out. (44) becomes

$$\overline{S}_{L,r_2}^{n_2}(\boldsymbol{\theta}_2) = \overline{S}_{L,r_{\Delta 2}}^{n_2}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_{\Delta 2})\cdot\left(\sum_{\boldsymbol{\theta}_1\in\mathcal{C}(\boldsymbol{\theta}_1(\boldsymbol{\theta}_2, n_1))} \overline{S}_{L,r_{\Delta 1}}^{n_1}(\boldsymbol{\theta}_1)\cdot\underbrace{\frac{\binom{n_1}{n_1\boldsymbol{\theta}_1}}{\binom{}{}}}\right. \tag{45}$$

Here $w(\boldsymbol{\theta}_1)$ can be considered as the weight for that sub-type $\boldsymbol{\theta}_1$. Using the property that the maximum is no less than the weighted average, we obtain

$$\overline{S}_{L,r_2}^{n_2}(\boldsymbol{\theta}_2) \leq \overline{S}_{L,r_{\Delta 2}}^{n_2}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_{\Delta 2})\cdot\max_{\boldsymbol{\theta}_1:\boldsymbol{\theta}_1\in\mathcal{C}(\boldsymbol{\theta}_1(\boldsymbol{\theta}_2, n_1))} \overline{S}_{L,r_{\Delta 1}}^{n_1}(\boldsymbol{\theta}_1). \tag{46}$$

Since the maximization is over a subset of $\mathcal{Q}_1$ (all possible types of length n_1 variable nodes), we have

$$\max_{\boldsymbol{\theta}_1:\boldsymbol{\theta}_1\in\mathcal{C}(\boldsymbol{\theta}_1(\boldsymbol{\theta}_2,n_1))} \overline{S}^{n_1}_{L,r_{\Delta 1}}(\boldsymbol{\theta}_1) \le \overline{S}^{n_1}_{L,r_{\Delta 1}}(\boldsymbol{\theta}^*_1), \quad (47)$$

where $\boldsymbol{\theta}^*_1$ is the type $\boldsymbol{\theta}_1$ that maximizes $\frac{\log \alpha_1}{n_1}$ (*and thus maximizes $\overline{S}^{n_1}$?*).

2) How to find the set of "compatible" $\boldsymbol{\theta}_1$ of length n_1 for a given $\boldsymbol{\theta}_2$, $\mathcal{C}(\boldsymbol{\theta}_1(\boldsymbol{\theta}_2, n_1))$. We know that without the limits of each element from 0 to q^2-1, then the number of ways (combinations) to choose n_1 objects is

$$\binom{n_1+q^2-1}{n_1}. \quad (48)$$

However, what if we limit the number of each element to $n_2\boldsymbol{\theta}_{2,0}$, $n_2\boldsymbol{\theta}_{2,1}$, $n_2\boldsymbol{\theta}_{2,2}$, …, $n_2\boldsymbol{\theta}_{2,q^2-1}$, then the problem becomes non-trivial.

After some searches, it seems that there are only formulas for unlimited repetitions. Problems with limited (but symmetric) repetitions may be solved case by case, but the calculation can get very messy. Our problem is even more complicated, as the repetition is irregular, i.e., different elements have different maximum repetitions. One solution is to use generating function as follows

$$f_{2\to\Delta 1}(x) = (1+x+\ldots+x^{n_2\boldsymbol{\theta}_{2,0}}). (1+x+\ldots+x^{n_2\boldsymbol{\theta}_{2,1}})\cdot\ldots\cdot(1+x+\ldots+x^{n_2\boldsymbol{\theta}_{2,q^2-1}}), \quad (49)$$

and then find the coefficient of the term x^{n_1}.

The generating function in (49) can be evaluated as follows (recall $\mathcal{Q}_2=\{0,1,\ldots,q^2-1\}$):

$$f_{2\to\Delta 1}(x) = \prod_{i\in\mathcal{Q}_2}(1+x+\ldots+x^{n_2,\boldsymbol{\theta}_{2,i}}) \quad (50)$$

$$= \prod_{i\in\mathcal{Q}_2}\frac{(1-x^{n_2\boldsymbol{\theta}_{2,i}+1})}{1-x} \quad (51)$$

$$= \frac{\prod_{i\in\mathcal{Q}_2}(1-x^{n_2\boldsymbol{\theta}_{2,i}+1})}{(1-x)^{q^2}}. \quad (52)$$

To obtain the coefficient of the term x^{n_1} in $f_{2\to\Delta 1}(x)$, $\lfloor f_{2\to\Delta 1}(x)\rfloor_{n_1}$, we use the standard coefficient extraction technique and note

$$\frac{1}{(1-x)^t} = \sum_{i=0}^{\infty}\binom{n+t-1}{t-1}x^n. \quad (53)$$

However, the calculation can still be tedious with generating function, as the number of ways to obtain x^{n_1} is enormous (or NOT?). First note that the numerator has q^2 terms, and therefore there are a total of

$$\binom{q^2}{0}+\binom{q^2}{1}+\ldots+\binom{q^2}{q^2}=2^{q^2}. \quad (54)$$

Hence, we can subtract the total power (ranging from 0 to n_1, call it m) from the chosen terms in the numerator from n_1, then multiply by $\binom{-q^2}{n_1-m}$.

3) If it is already difficult to find the number of compatible $\boldsymbol{\theta}_1$, then evaluating (45) will be more difficult. However, is there any trick that we can use to find the exact value of $\overline{S}^{n_2}_{L,r_2}$? For example, do some of the terms sum to a nice number? or can we modify our code design parameters to make the sum nice? (which might be unlikely.) Finally
4) Plots of the weights for $\boldsymbol{\theta}_1$
5) Size of the compatible $\boldsymbol{\theta}_1$ can be considered a high-dimensional cube cut by a (diagonal) hyper-plane. Since all the $0\le a_i\le b_i, i\in\mathcal{Q}_2$ (or $\mathcal{Q}_k$) are parallel constraints. The number of vertices lie on the intersection is the size of the compatible set.

Follow attempt #1, we obtain

$$\frac{\log\alpha_2}{n_2} = \arg\max_{\boldsymbol{\theta}_2\in\mathcal{Q}_2}\frac{\overline{S}^{n_2}_{L,r_2}(\boldsymbol{\theta}_2)}{(M_2^2-1)B(n_2,n_2\boldsymbol{\theta}_2)q^{-2n_2}}, \quad (55)$$

where $M_2=q^{n_2R_2}$ and R_2 is defined in (26). $\overline{S}^{n_2}_{L,r_2}(\boldsymbol{\theta}_2)$ can be calculated from (45)

$$\overline{S}^{n_2}_{L,r_2}(\boldsymbol{\theta}_2) = \overline{S}^{n_2}_{L,r_{\Delta 2}}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_{\Delta 2})\cdot\left(\sum_{\boldsymbol{\theta}_1\in\mathcal{C}(\boldsymbol{\theta}_1(\boldsymbol{\theta}_2,n_1))}\overline{S}^{n_1}_{L,r_{\Delta 1}}(\boldsymbol{\theta}_1)\cdot\frac{\binom{n_1}{n_1\boldsymbol{\theta}_1}\binom{n_\Delta}{n_{\Delta 2}\boldsymbol{\theta}}}{\binom{n_2}{n_2\boldsymbol{\theta}_2}}\right. \quad (56)$$

C. Dispersion-style Analysis

Use the generalized RCU bound

V. Conclusion

The conclusion goes here.

Appendix A
Appendix I

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

Appendix C
Previous (may be incorrect) work

The expressions of the two generating functions are as follows:

For $A_2(\boldsymbol{x}_1,\boldsymbol{x}_2)$, if we want to use the same approach from $A_1(\boldsymbol{x}_1)$, then we need to separate a total of 2^{ρ_2} different cases, as each socket in a group-2 check node might be connected to group-1 variable node (corresponds to $x_{2,1}$) or group-2 variable node (correspond to $x_{2,2}$). This will make calculation extremely difficult after we take $A_2(\boldsymbol{x}_1,\boldsymbol{x}_2)$ to the power of $r_{\Delta 2}$.

To solve this problem, we take a detour and calculate $A_2(\boldsymbol{x}_2)$ and calculate the coefficient $\lfloor (A_2(\boldsymbol{x}_2))^{r_{\Delta 2}} \rfloor_{n_2\lambda_2\boldsymbol{\theta}_2}$, and then we claim that

$$\lfloor A_2(\boldsymbol{x}_{2,1}, \boldsymbol{x}_{2,2})^{r_{\Delta 2}} \rfloor_{n_1\lambda_2\boldsymbol{\theta}_1, n_{\Delta 2}\lambda_2\boldsymbol{\theta}_{\Delta 2}} = \frac{\lfloor (A_2(\boldsymbol{x}_2))^{r_{\Delta 2}} \rfloor_{n_2\lambda_2\boldsymbol{\theta}_2}}{\binom{n_2}{n_1}}, \tag{57}$$

due to two reasons: 1. there are $\binom{n_2}{n_1}$ different ways to decompose a type $\boldsymbol{\theta}_2$ of length n_2 into a pair of sub-types $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_{\Delta 2})$ of lengths $(n_1, n_{\Delta 2})$; 2. The number of codematrices for each pair is identical.

REFERENCES

[1] Y. Liu and M. Effros, "Finite-blocklength and error-exponent analyses for ldpc codes in point-to-point and multiple access communication," in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, 2020, pp. 361–366.

[2] L. G. Roberts, "Aloha packet system with and without slots and capture," *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, 1975.

[3] B. Hajek and T. van Loon, "Decentralized dynamic control of a multiaccess broadcast channel," *IEEE Trans. Aut. Control*, vol. 27, no. 3, pp. 559–569, 1982.

[4] J. Tsitsiklis, "Analysis of a multiaccess control scheme," *IEEE Trans. Aut. Control*, vol. 32, no. 11, pp. 1017–1020, 1987.

[5] H. Wu, C. Zhu, R. J. La, X. Liu, and Y. Zhang, "Fasa: Accelerated s-aloha using access history for event-driven m2m communications," *IEEE/ACM Trans. Net.*, vol. 21, no. 6, pp. 1904–1917, 2013.

[6] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, "Grant-free non-orthogonal multiple access for iot: A survey," *IEEE Comm. Surveys & Tutorials*, vol. 22, no. 3, pp. 1805–1838, 2020.

[7] M. V. Burnashev, "Data transmission over a discrete channel with feedback. random transmission time," *Problems of Information Transmission*, vol. 12, no. 4, pp. 10–30, 1976.

[8] A. Tchamkerten and E. Telatar, "A feedback strategy for binary symmetric channels," in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, 2002, pp. 362–.

[9] S. C. Draper, B. J. Frey, and F. R. Kschischang, "Efficient variable length channel coding for unknown dmcs," in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, 2004, pp. 379–379.

[10] Y. Polyanskiy, H. V. Poor, and S. Verdu, "Feedback in the non-asymptotic regime," *IEEE Trans. Inf. Theory*, vol. 57, no. 8, pp. 4903–4925, 2011.

[11] M. Effros, V. Kostina, and R. C. Yavas, "Random access channel coding in the finite blocklength regime," in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, 2018, pp. 1261–1265.

[12] R. C. Yavas, V. Kostina, and M. Effros, "Gaussian multiple and random access in the finite blocklength regime," in *Proc. IEEE Int. Symp. on Information Theory (ISIT)*, 2020, pp. 3013–3018.

[13] S. Chen, M. Effros, and V. Kostina, "Lossless source coding in the point-to-point, multiple access, and random access scenarios," *IEEE Trans. Inf. Theory*, vol. 66, no. 11, pp. 6688–6722, 2020.

[14] R. Gallager, "Low-density parity-check codes," *IEEE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[15] C. Measson, A. Montanari, and R. Urbanke, "Maxwell's construction: the hidden bridge between maximum-likelihood and iterative decoding," *IEEE Trans. Inf. Theory*, pp. 225–, June 2004.

[16] A. Bennatan and D. Burshtein, "On the application of LDPC codes to arbitrary discrete-memoryless channels," *IEEE Trans. Inf. Theory*, vol. 50, no. 3, pp. 417–438, March 2004.

[17] ——, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 549–583, Feb 2006.