Non-asymptotic Coded Slotted ALOHA

Mohammad Fereydounian
University of Pennsylvania
mferey@seas.upenn.edu

Xingran Chen
University of Pennsylvania
xingranc@seas.upenn.edu

Hamed Hassani
University of Pennsylvania
hassani@seas.upenn.edu

Shirin Saeedi Bidokhti
University of Pennsylvania
saeedi@seas.upenn.edu

***Abstract*—Coding for random access communication is a key challenge in Internet of Things applications. In this paper, the well-known scheme of Coded Slotted Aloha (CSA) is considered and its performance is analyzed in the non-asymptotic regime where the frame length and the number of users are finite. A density evolution framework is provided to describe the dynamics of decoding, and fundamental limits are found on the maximum channel load (i.e., the number of active users per time slot) that allows reliable communication (successful decoding). Finally, scaling laws are established, describing the non-asymptotic relation between the probability of error, the number of users, and the channel load.**

I. Introduction

The technology of Internet of Things (IoT) has brought new challenges in the design of multi-access communication systems. In traditional networks, the number of users is small and it is hence practical to coordinate them for transmission. For example, this can be done with the help of a common clock that can be implemented through a low rate (vanishing by blocklength) communication link. In IoT applications, however, the number of users is very large, orders of magnitude larger than the blocklength. For example, in Low-Power Wide-Area Networks (LP-WANs), the number of users is in the order of tens of millions. Clearly, coordinating all the users is infeasible in such scenarios and hence communication should be assumed uncoordinated. This has motivated the key challenge of coding for massive random access communication, where senders communicate their packets in a bursty manner at random times.

The literature on massive random access communication ranges from traditional (slotted) ALOHA-type protocols [1–3], to recent information theoretic frameworks and code designs in the regime of operation where the total number of users scale linearly with the blocklength [4,5]. Slotted ALOHA [1] is one of the first random-access protocols that deals with collisions through random re-transmission of packets and it is still in use for cellular and satellite communication. In [3,6], the problem of recovering from collisions has been tackled from the perspective of error correcting codes. The idea is to encode redundancy into the transmitted packets via repetition coding, leading to the class of Irregular Repetition Slotted ALOHA (IRSA) schemes [3], or more generally via an error correcting code, leading to the class of Coded Slotted ALOHA (CSA) schemes [6]. The redundancy is then exploited in decoding by successive cancellation and a corresponding iterative message passing algorithm. Exploiting a bipartite graph representation, [6] derives density evolution equations for CSA and analyzes the SIC process in an asymptotic setting where the frame length and the number of users both tend to infinity (their ratio remaining constant).

In this work, we provide a non-asymptotic analysis of CSA. The non-asymptotic regime is arguably the practical regime of interest especially in applications where delay is of importance. Previous works on random access schemes, including coded variants of SA, have mostly relied on simulations to address the non-asymptotic performance of the schemes. Some notable exceptions are [7–10]. The works in [7–9] analyze CSA in the error floor region, and [10] provides a finite-length analysis for IRSA in the waterfall region. To the best of our knowledge, a finite-length analysis of the more general class of CSA protocols in the waterfall region has been missing from the literature. In summary, our contributions are as follows:

- We use connections between CSA and low-density parity-check (LDPC) codes that were established in [6] and build on methods of [11] to analyze LDPC codes in the finite blocklength regime. In a recent work, [10] uses a similar approach and provides the non-asymptotic analysis of the special class of IRSA random access codes. While the existing analytical derivations for LDPC ensembles are directly applicable to the analysis of IRSA, they turn out to be insufficient for CSA. Assuming a regular CSA ensemble in which users have the same rate and blocklength, we generalize techniques of LDPC codes to analyze the performance of regular CSA in the non-asymptotic regime. This generalization includes the derivation of a new density evolution procedure for CSA via differential equations (see Section III) as well as *scaling laws* which describe precisely how the probability of error scales with respect to the number of users and the channel load (see Section IV).
- Channel load G is defined to be the number of active users per time slot. It turns out that CSA ensembles encounter a threshold effect, meaning that there is a maximum achievable channel load G^* such that for $G > G^*$, it is almost surely impossible to decode all messages and for $G < G^*$ it is almost surely possible. We obtain G^* analytically as a root of an algebraic equation (see Section III).
- Finally, we simulate the performance of CSA protocols and validate our analytical results (see Section V).

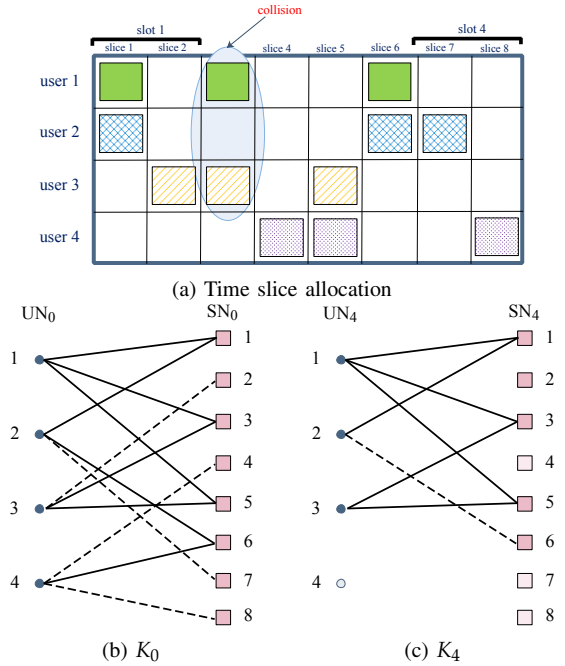
We refer to [12] for a long version of this paper with all the proofs.

II. Preliminaries

Suppose we have a multi-access channel with N_a active users. An active user is one that has a message to send at the current time. In Slotted Aloha (SA) protocols, channel resources (e.g., time) are divided into slots. Let M denote the number of time slots. Channel load G is then defined as

$$G = \frac{N_a}{M}.$$

Each active user chooses a set of time slots to send its message. If two users transmit during the same time slot, a *collision* occurs (see Figure **1**a). In case of collision, the receiver has only access to the summation of all the collided packets. We assume that a packet cannot be recovered in a time slot unless all the other collided packets are previously recovered and hence can be subtracted from the summation. To increase reliability, one can encode a message in one packet and repeat this packet n times. In this way, the information rate is reduced to $R = 1/n$. This scheme is called Repetition Slotted ALOHA (RSA). Replacing repetition coding with a general coding scheme is a fundamental way to increase the rate while ensuring high reliability in communication systems. This approach leads to Coded Slotted Aloha (CSA) introduced in [6].



(a) Time slice allocation

(b) K_0 (c) K_4

Figure 1: A realization of CSA scheme with parameters $N_a = 4, M = 4, n = 3, k = 2, m = kM = 8$. The edges connected to singletons are depicted in dashed lines.

The CSA scheme is described as follows: Each time slot is divided into k smaller time slices. We thus have $m = kM$ time slices in total. Each message is also divided into k smaller packets, and a coding scheme is used to encode these k packets into n coded packets. Each user chooses n distinct time slices out of m time slices uniformly at random to send its n coded packets. The setting is illustrated in Figure **1**a for 4 users with $n = 3, k = 2$, and $m = 8$. We assume that fully recovering any set of k coded packets out of n coded packets is enough to recover the original message. Therefore, the resulting rate is $R = k/n$ (note that the special case $k = 1$ is equivalent to RSA). For simplicity, we consider a regular scenario in which all users utilize the same blocklength n. We refer to such a CSA scheme by CSA(n, k, N_a, M).

For decoding, we use a successive interference cancellation (SIC) procedure: At each step $q \in \{0, 1, 2, \dots\}$, we first find a collision-free time slice containing only one packet and then we fully recover that packet. From this packet, we identify the corresponding user, and we mark this packet as "decoded". For any user, when k out of n packets are marked as decoded, then the user's original message can be fully recovered, and as a result, all of its n packets are known. We thus can subtract those n packets from their corresponding time slices, remove the user, and mark the user as *resolved*. This elimination may result in further collision-free slices with only one packet inside. We then continue decoding by finding the next collision-free slice. The decoding stops when there is no collision-free slice left. At this point, if all users are resolved, decoding is successful, otherwise, we declare a *B-error* (in analogy to block error in decoding LDPC codes). If a B-error occurs, the fraction of unresolved users is statistically known as *packet loss probability* (PLP). In terms of analysis, PLP can be simply computed in terms of the probability of B-error (see [13]).

For a refined analysis of the SIC process discussed above, we need a more structured modeling based on a bipartite graph which we call the decoding graph. This is analogous to the so-called Tanner graph used for the analysis of LDPC codes [13]. Define a bipartite graph K with two sets of nodes: A set of N_a user nodes UN and a set of m slice nodes SN. The i^{th} user node represents the i^{th} active user, where $i \in \{1, \dots, N_a\}$ and the j^{th} slice node represents the j^{th} time slice, where $j \in \{1, \dots, m\}$. For each user, there is an edge to the n slice nodes in which its packets are sent. As a result, each user node has degree n. However, the degree of a slice node is potentially between 0 and N_a. We recall that neighbours of a user node are chosen uniformly at random among slice nodes. A slice node with degree 1 is called a *singleton*. Moreover, at each step, the graph resulted from the previous step is called the *residual graph*. We refer to this SIC decoding process as the *peeling* process.

Figure **1**a shows the initial time slice allocation while Figures **1**a, **1**b, and **1**c illustrate the SIC process for a CSA realization with 4 users and parameters $n = 3, k = 2$, and $m = 8$. Figures **1**b and **1**c represent the residual graphs at iterations $q = 0$ and $q = 4$. As it can be seen, the initial graph K_0 includes 4 singletons, namely, slice nodes $2, 4, 7$, and 8. The slice nodes $2, 4, 7$, will be removed in iterations $q = 0, 1, 2$, respectively. In iteration $q = 3$, we first remove the slice node 8. At this point, 2 out 3 slice nodes connecting to user node 4 are decoded. This means user node 4 is resolved and it must be removed from the graph. This makes a new singleton i.e., slice node 6 and the process goes to iteration 4.

A. Notation and Definitions

Throughout this paper, the set of edges of a graph K is denoted by $E(K)$, the number of elements of a set S is denoted by $|S|$, and the expectation of a random variable X is denoted by $\mathbb{E}[X]$. Consider a continuous time t. Suppose step $q \in \{0, 1, 2, \dots\}$ of the peeling process happens at $t = q\Delta t$ where $\Delta t = 1/E$, and $E = nN_a$ is the total number of edges in K. Also, let t_f be stopping time due to failure or success. Let K_t be the residual graph at time t and UN_t and SN_t be the corresponding sets of user nodes and slice nodes of K_t. Note that $K_0 = K$, $UN_0 = UN$, $SN_0 = SN$ and thus $|UN_0| = N_a$ and $|SN_0| = m = kM$. Now, for $i \in \{n-k+1, \dots, n\}$, $j \in \{0, \dots, N_a\}$, and $t \in [0, t_f]$, we define

$$L_i(t) := \mathbb{E}\big[\,|\{e \in E(K_t) : e \text{ is connected to } u \in UN_t \text{ with } \deg(u) = i\}|\,\big],$$

$$l_i(t) := \frac{L_i(t)}{E} = L_i(t)\Delta t, \quad \lambda_i := l_i(t)\Big|_{t=0},$$

$$R_j(t) := \mathbb{E}\big[\,|\{e \in E(K_t) : e \text{ is connected to } s \in SN_t \text{ with } \deg(s) = j\}|\,\big],$$

$$r_j(t) := \frac{R_j(t)}{E} = R_j(t)\Delta t, \quad \rho_j := r_j(t)\Big|_{t=0}.$$

Also define $e(t) := \mathbb{E}\left[|E(K_t)|\right]/E$ which represents the normalized expected number of edges in the residual graph.

III. Asymptotic Analysis

A. Differential Equations for Density Evolution

Consider the CSA(n, k, N_a, M) model defined in Section II. By applying the peeling process to the elements of this model, as N_a increases, the sequence of residual graphs closely follows a "typical path". We now construct a set of coupled differential equations describing this typical behavior. This method is known as the Wormald's method and has been applied to variety of problems [13, 14]. We call the sequence $\{l_i(t), r_j(t)\}_{i,j}$, the degree distribution of residual graph at time t. This degree distribution constitutes a sufficient statistic for tracking the distribution of the residual graph (see [13]).

In order to obtain differential equations describing the dynamics of $l_i(t)$ and $r_j(t)$, our first step is to compute the change in $L_i(t)$ at each step of the peeling process i.e., computing $L_i(t+\Delta t) - L_i(t)$. As defined earlier, $L_i(t)$ denotes the expected number of edges which are connected to degree i user nodes at time t. Consider the peeling process at time t. First, a singleton (a degree 1 slice node) $s \in SN_t$ is chosen arbitrarily. Let e be the connecting edge. Then, s and e are removed. By this removal, a change in $L_i(t)$ happens only when e is connected to a degree i or $i+1$ user node. With probability $l_i(t)/e(t)$, e is connected to a degree i user node u. In this case, by removing e, u is not of degree i anymore and thus $L_i(t)$ will be decreased by i units. Also, with probability $l_{i+1}(t)/e(t)$, e is connected to a degree $i+1$ user node u'. In this case, by removing e, u' becomes a degree i node and thus $L_i(t)$ will be increased by i units. Note that the latter case is valid when $i < n$ since $i+1$ must exist. Thus, based on this argument, we have the following equations for the case $n-k+1 \le i \le n-1$ and the case $i = n$:

$$\begin{cases} L_i(t+\Delta t) - L_i(t) = -i \cdot \dfrac{l_i(t)}{e(t)} + i \cdot \dfrac{l_{i+1}(t)}{e(t)}, \\ L_n(t+\Delta t) - L_n(t) = -n \cdot \dfrac{l_n(t)}{e(t)}. \end{cases} \tag{1}$$

Considering $l_i(t) = L_i(t)\Delta t$ and $N_a \to \infty$, we have $E = nN_a \to \infty$ and $\Delta t = 1/E \to 0$. Therefore, (1) becomes

$$\begin{cases} \dfrac{dl_i(t)}{dt} = i \cdot \dfrac{l_{i+1}(t) - l_i(t)}{e(t)}, \; n-k+1 \le i < n, \\ \dfrac{dl_n(t)}{dt} = -n \cdot \dfrac{l_n(t)}{e(t)}. \end{cases} \tag{2}$$

Let us now explain how to derive similar equations for the variables r_j. Consider the decoding procedure at time t. Define $a(t)$ to be the expected number of the edges that are removed in this time step. Let $s \in SN_t$ be the singleton that is removed and e be the connecting edge. With probability $l_{n-k+1}(t)/e(t)$, e is connected to a degree $n-k+1$ user node u. In this case, after peeling s and e, u becomes of degree $n-k$. This leads to the removal of u which further results in removing $n-k$ other edges (connected to u) and in total $n-k+1$ edges will be removed in this step. Otherwise, if e is not connected to a user node with degree $n-k+1$, only e will be removed and decoding process goes to the next step. Thus, we have

$$a(t) = (n-k+1) \cdot \frac{l_{n-k+1}(t)}{e(t)} + \sum_{i=n-k+2}^{n} 1 \cdot \frac{l_i(t)}{e(t)}. \tag{3}$$

Note that the expected number of deleted edges other than e, is $a(t)-1$. Now, consider one of these deleted edges, namely, e'. Then with probability $r_j(t)/e(t)$, e' is connected to a degree j slice node. In this case, $R_j(t)$ will be decreased by j units. Also, with probability $r_{j+1}(t)/e(t)$, e' is connected to a degree $j+1$ slice node and in this case, R_j will be increased by j units. This is true for any such e' and expected number of them is $a(t)-1$. Here, it is important to note that for j, the degree of a slice node, we have $j \le N_a$. Thus, the above argument is valid when $j < N_a$ since $j+1$ must exist. This argument results in the following equations:

$$\begin{cases} \text{for } \; 2 \le j < N_a: \; R_j(t+\Delta t) - R_j(t) \\ = \left(-j \cdot \dfrac{r_j(t)}{e(t)} + j \cdot \dfrac{r_{j+1}(t)}{e(t)} \right) \cdot (a(t)-1), \\ R_{N_a}(t+\Delta t) - R_{N_a}(t) = -N_a \cdot \dfrac{r_{N_a}(t)}{e(t)} \cdot (a(t)-1). \end{cases} \tag{4}$$

Substituting $r_j(t) = R_j(t)\Delta t$ and considering $N_a \to \infty$ then gives $E = nN_a \to \infty$. As a result, $\Delta t = 1/E \to 0$ and only the first equation in (4) matters. Therefore, (4) becomes

$$\frac{dr_j(t)}{dt} = j \cdot \big(r_{j+1}(t) - r_j(t)\big) \cdot \frac{a(t)-1}{e(t)}, \qquad j \ge 2. \tag{5}$$

In order to solve (2) and (5) analytically, the following change of variable is useful since it eliminates $e(t)$:

$$t \longmapsto x = \exp\left(\int_0^t \frac{d\tau}{e(\tau)} \right) \Rightarrow \frac{dx}{x} = \frac{dt}{e(t)}. \tag{6}$$

As a result, for the initial point, we have $t = 0 \longmapsto x = 1$. For simplicity, we consider this new variable x as "time". By applying this change of variable to (2) and (5), we have

$$\begin{cases} \dfrac{dl_i(x)}{dx} = i \cdot \dfrac{l_{i+1}(x) - l_i(x)}{x}, \; n-k+1 \le i < n, \\ \dfrac{dl_n(x)}{dx} = -n \cdot \dfrac{l_n(x)}{x}, \end{cases} \tag{7}$$

$$\frac{dr_j(x)}{dx} = -j \cdot \big(r_{j+1}(x) - r_j(x)\big) \cdot \frac{a(x)-1}{x}, \qquad j \ge 2. \tag{8}$$

Note that by using the change of variable $t \mapsto x$ described in (6), as $N_a \to \infty$, we have

$$e(x) = \sum_{i=n-k+1}^{n} l_i(x) = \sum_{j \ge 1} r_j(x). \tag{9}$$

$$\Rightarrow r_1(x) = e(x) - \sum_{j \ge 2} r_j(x). \tag{10}$$

In order to solve these equations, we need to determine their initial conditions. Recall that we defined $\lambda_i = l_i(t=0) = l_i(x=1), \rho_j = r_j(t=0) = r_j(x=1)$, and that λ_i denotes the fraction of edges connected to a degree i user node in the initial graph K_0. As we discussed earlier, initially each user node has degree n which means

$$\lambda_i = \begin{cases} 0, & n-k+1 \le i < n, \\ 1, & i = n. \end{cases} \tag{11}$$

Determining ρ_j requires further computations. We ignore empty time slices (corresponding to $j = 0$) and focus on $j \ge 1$. The following lemma gives ρ_j for $j \ge 1$.

Lemma 1. *Consider CSA(n,k,N_a,M) and suppose $N_a \to \infty$ and $M \to \infty$ with $G = N_a/M$ to be a fixed constant. Also, suppose $R = k/n$. Then for $j \ge 1$, we have*

$$\rho_j = \frac{1}{(j-1)!} \left(\frac{G}{R} \right)^{j-1} \exp\left(-\frac{G}{R} \right). \tag{12}$$

The set of differential equations given by (7), (8), and, (10) together with their initial conditions given by (11) and (12) characterize the evolution of degree distribution of the residual graph. In the rest of this manuscript, we refer to them as *density evolution*.

B. Solving Density Evolution

In this section, we provide solutions for the density evolution discussed in the previous section.

Theorem 1. *The finite recursive sequence of differential equations defined in* (7) *together with initial conditions $l_i(1) = \lambda_i$, where λ_i is given by* (11)*, result in the following solution:*

$$l_i(x) = \sum_{j=n-k+1}^{n} \frac{\alpha_j^{(i)}}{x^j}, \qquad i \in \{n-k+1, \cdots, n\}, \tag{13}$$

where $\left\{\alpha_j^{(i)}\right\}_{i,j}$, $n-k+1 \le i \le j \le n$, *is a finite 2-dimensional recursive sequence of integers which is (uniquely) determined by the following equations:*

$$\begin{cases} \alpha_n^{(n)} = 1 \\ \alpha_i^{(i)} = \sum_{j=i+1}^{n} (-1)^{j-i+1} \binom{j-1}{i-1} \alpha_j^{(j)}, \; n-k+1 \le i < n, \\ \alpha_j^{(i)} = (-1)^{j-i} \binom{j-1}{i-1} \alpha_j^{(j)}, \; i < j \le n. \end{cases} \tag{14}$$

Lemma 2. *Consider* $e(x)$ *as given in* (9)*, then we have*

$$e(x) = \sum_{j=n-k+1}^{n} \frac{\beta_j}{x^j}, \tag{15}$$

where β_j *is a finite sequence of integers defined below.*

$$\beta_j := \alpha_j^{(j)} (-1)^j \sum_{i=n-k+1}^{j} (-1)^i \binom{j-1}{i-1}, \tag{16}$$

and $\alpha_j^{(j)}$ *is given by* (14)*.*

In order to find the solution of the infinite recursive sequence of differential equations described in (8), we use a function $\lambda(x)$. Let $\lambda(x)$ be the function satisfying

$$\frac{\lambda'(x)}{\lambda(x)} = \frac{a(x)-1}{x}, \qquad \lambda(1) = 1, \tag{17}$$

where $a(x)$ is obtained from (3) by applying the change of variable in (6).

Lemma 3. *Consider* $\lambda(x)$ *defined in* (17)*, then we have*

$$\lambda(x) = \exp\left((n-k) \int_1^x \frac{\sum_{j=0}^{k-1} \alpha_{n-j}^{(n-k+1)} y^j}{\sum_{j=0}^{k-1} \beta_{n-j} y^{j+1}} dy \right), \tag{18}$$

where $\alpha_j^{(j)}$ *is determined by* (14) *and* β_j *is defined in* (16)*.*

Theorem 2. *The infinite recursive sequence of differential equations defined in* (8) *together with initial conditions* $r_j(1) = \rho_j$ *given in* (12) *result in the following solution:*

$$r_j(x) = \frac{1}{(j-1)!\lambda^j(x)} \left(\frac{G}{R}\right)^{j-1} \exp\left(-\frac{G}{R\lambda(x)}\right), j \ge 2. \tag{19}$$

Moreover, this result, together with (15) *and* (10)*, imply*

$$r_1(x) = \left[\sum_{j=n-k+1}^{n} \frac{\beta_j}{x^j}\right] - \frac{1}{\lambda(x)} \left(1 - \exp\left(-\frac{G}{R\lambda(x)}\right)\right), \tag{20}$$

where β_j *is defined in* (16) *and* $\lambda(x)$ *is given in* (18)*.*

C. Finding The Maximum Achievable Channel Load G^*

We now use the results of the previous section to find an analytic expression for the threshold G^*, i.e. the maximum load for which decoding succeeds with high probability. Suppose that the decoding procedure stops at time x_f. Decoding is successful if and only if the residual graph at time x_f is empty; otherwise, there are some unresolved user nodes left. Note that regardless of success or failure, we have $r_1(x_f) = 0$. To illustrate better, it is helpful to consider the plots given in Figure **2**. When $G < G^*$, the function $r_1(x)$ takes value 0 (i.e. touches the x-axis) only when the decoding graph becomes empty and all the other variables become 0. The function is strictly positive otherwise. It can be easily argued (see the longer version of this paper [12]) that, at the threshold $G = G^*$, the function $r_1(x)$ becomes tangent to the x-axis exactly once at a point x^* (see Figure **2**). In order to find the relation between x^* and G^*, we consider (20). For any x, we define $\tilde{G}(x)$ as the channel load which results in $r_1(x) = 0$. Hence, $\tilde{G}(x)$ is found as

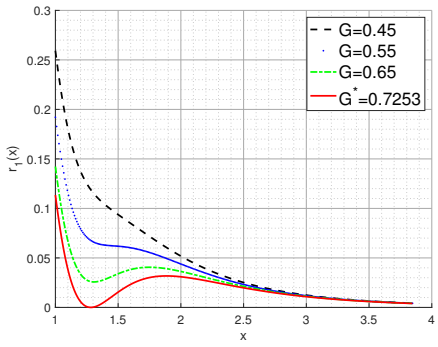


Figure 2: The curves $r_1(x)$, computed from (20) with $n = 6$ and $k = 2$, for different channels loads $G = 0.45, 0.55, 0.65$, as well as the threshold $G^* = 0.7253$. Note that at G^* the curve $r_1(x)$ is tangent to the x-axis at $x^* = 1.2822$. The values of x^* and G^* are computed from Theorem 3.

$$\tilde{G}(x) = -R\lambda(x) \log\left(1 - e(x)\lambda(x)\right), \tag{21}$$

where $e(x)$ is given in (15). We further know that exactly at the threshold $G = G^*$ the function $r_1(x)$ is tangent to the x-axis at $x = x^*$. From this, we can conclude that the derivative of the function $\tilde{G}(x)$ is zero at $x = x^*$ (see [12]). We thus have

$$\left.\frac{d\tilde{G}(x)}{dx}\right|_{x=x^*} = 0, \qquad G^* = \tilde{G}(x^*). \tag{22}$$

Using these relations, an algebraic formulation for G^* is found in the following theorem.

Theorem 3. *Consider* $l_{n-k+1}(x)$*,* $e(x)$*,* $\lambda(x)$*, and* $\tilde{G}(x)$ *which are given by* (13)*,* (15)*,* (18)*, and* (21)*, respectively, and define* $h(x) = e(x)\lambda(x)$*. Then the maximum achievable channel load* G^* *satisfies* $G^* = \tilde{G}(x^*)$*, where* x^* *is the solution of the following algebraic equation:*

$$\log(1-h(x)) = \frac{1-h(x)}{h(x)} \left(1 + \frac{xe'(x)}{(n-k)l_{n-k+1}(x)}\right). \tag{23}$$

IV. Non-asymptotic Analysis

Let $P_B(N_a, G)$ denote the probability that a B-error (defined in Section II) occurs in the CSA(n, k, N_a, M) scheme under the discussion with N_a active users and $M = N_a/G$ time slots. $P_B(N_a, G)$ encounters a threshold effect in terms of G. In other words, there is a threshold G^* such that if $G > G^*$, then $P_B(N_a, G) \to 1$ and if $G < G^*$, $P_B(N_a, G) \to 0$. In the non-asymptotic case, however, this transition occurs smoothly in a region close to G^* which is called the *waterfall* region. The behavior of P_B in the waterfall region turns out to be governed by the so-called *scaling law*. [11] analyzed the scaling law for LDPC ensembles. In analogy to that analysis, we will have the following formula for our setting:

$$P_B(N_a, G) = Q\left(\frac{\sqrt{N_a}}{\alpha}\left(G^* - \beta N_a^{-2/3} - G\right)\right). \tag{24}$$

where $Q(.)$ is the tail probability of the standard normal distribution and the scaling parameters α and β will be obtained based on the asymptotic analysis.

Our approach for derivation of α and β will be generally based on [11]. Let $d = k+1$ and define the $(d+1)$-dimensional vector $z = (z_0, z_1, \cdots, z_d) = (r_1, r_2, l_{n-k+1}, \cdots, l_n)$. We abuse the notation slightly and use r_j to refer to two different notions: the number of edges connected to degree j slice nodes at time x and the number of degree j slice nodes at time x. We distinguish the two by referring to them as the corresponding edge-based and node-based quantities of r_j, respectively. We exploit similar definitions for $l_i(x)$. Now define $\delta^{(z_i z_j)}(x)$ to be the normalized covariance between the corresponding node-based quantities of z_i and z_j at time x. The analysis of the finite dimensional Markov process over z then leads to the following set of $\binom{d+1}{2} + d + 1$ coupled differential equations for $\delta^{(z_i z_j)}$, where $i, j \in \{0, \cdots, d\}, i \leq j$:

$$\frac{d\delta^{(z_i z_j)}(x)}{dx} = \frac{e(x)}{x}\left[\frac{\hat{f}^{(z_i z_j)}(x)}{n} + \sum_{k=0}^{d} \delta^{(z_i z_k)}(x)\frac{\partial \hat{f}^{(z_j)}(x)}{\partial z_k} + \frac{\partial \hat{f}^{(z_i)}(x)}{\partial z_k}\delta^{(z_k z_j)}(x)\right]. \quad (25)$$

Here, $\hat{f}^{(z_i)}(x)$ represents the expected change of the corresponding edge-based quantity of z_i and $\hat{f}^{(z_i z_j)}(x)$ represents the covariance between the corresponding edge-based quantities of z_i and z_j at time x. $\hat{f}^{(z_i)}$ and $\hat{f}^{(z_i z_j)}$ are called local drifts and local covariances and we refer to the differential system (25) by *covariance evolution*. In order to solve the covariance evolution, local drifts, local covariances and the initial conditions of the functions $\delta^{z_i z_j}(x)$ at $x = 1$ need to be computed for the formulation of CSA. For the derivation of these quantities, we refer to [12, Theorem 4, Lemma 6].

$$\alpha = -\sqrt{\frac{\delta^{(r_1 r_1)}(x)}{n}}\left(\frac{\partial r_1(x;G)}{\partial G}\right)^{-1}\Bigg|_{x=x^*, G=G^*},$$
$$\beta = -\left(\frac{\hat{f}^{(r_1 r_1)}(x)}{n}\right)^{2/3}\left[\sum_{k=1}^{d}\frac{\partial \hat{f}^{(r_1)}(x)}{\partial z_k}\hat{f}^{(z_k)}(x)\right]^{-1/3} \times \left(\frac{\partial r_1(x;G)}{\partial G}\right)^{-1}\Bigg|_{x=x^*, G=G^*}. \quad (26)$$

V. Simulation Results

In this section, we will compare our predictions of maximum achievable channel load G^* and probability of B-error P_B with the outcomes of simulations.

Table I, shows computation results of G^* from Theorem 3 versus values of G^* which are obtained from simulating the corresponding CSA scheme under various settings of n and k. These simulations are obtained with $N_a = 20000$ and by averaging over 2000 trials.

In Figure **3**, we compare our non-asymptotic prediction from (24) to simulation results in the waterfall region. All plots share the same setting where $n = 5$ and $k = 3$. The solid curves correspond to the computation of (24) using the scaling parameters $\alpha = 0.42362$ and $\beta = 0.8629$ obtained from (26). The dashed curves correspond to the probability of error obtained from simulating over 2×10^5 randomly chosen elements from the CSA ensemble and averaging the results. The curves represent the plots for $N_a = 1000, 2000, 4000, 8000, 16000, 32000$ with the leftmost curve corresponding to $N_a = 1000$ and the rightmost curve corresponding to $N_a = 32000$. Moreover, the vertical dashed line accord with $G^* = 0.5840$ which is obtained from (23). It can be seen that (24) accurately predicts the actual probability or B-error with a very high precision.

TABLE I: Computed G^* versus simulated G^* under $N_a = 20000$ by averaging over 2000 trials.

Parameters	Simulated G^*	Computed G^* from Theorem 3
$n = 5, k = 2$	0.737	0.7388
$n = 6, k = 3$	0.669	0.6699
$n = 8, k = 5$	0.545	0.5458
$n = 12, k = 10$	0.266	0.2664
$n = 25, k = 4$	0.459	0.4595

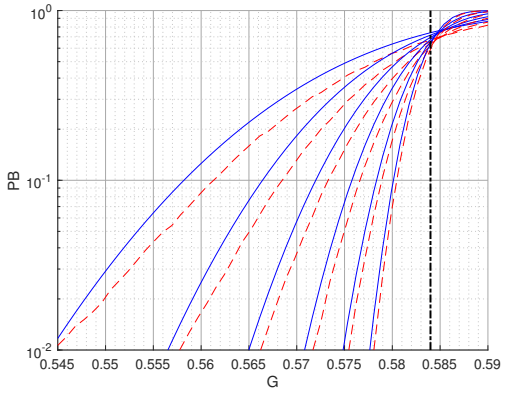


Figure 3: The probability of B-error P_B in terms of G for $N_a = 1000, 2000, 4000, 8000, 16000, 32000$. All curves are obtained using the parameters $n = 5$, $k = 3$. The solid and dashed lines correspond to results of computations and simulations respectively. The vertical dashed line shows $G^* = 0.5840$. The scaling parameters are $\alpha = 0.42362$ and $\beta = 0.8629$.

References

[1] N. Abramson, "The ALOHA system: another alternative for computer communications," in *1970 Fall Joint Computer Conf.*

[2] E. Casini, R. D. Gaudenzi, and O. D. R. Herrero, "Contention resolution diversity slotted ALOHA (CRDSA): An enhanced random access scheme for satellite access packet networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1408–1419, April 2007.

[3] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Transactions on Communications*, vol. 59, no. 2, pp. 477–487, Feb 2011.

[4] X. Chen, T. Chen, and D. Guo, "Capacity of gaussian many-access channels," *IEEE Transactions on Information Theory*, vol. 63, no. 6, 2017.

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

[6] E. Paolini, G. Liva, and M. Chiani, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *IEEE Transactions on Information Theory*, vol. 61, pp. 6815–6832, 2015.

[7] A. Vem, K. R. Narayanan, J. Cheng, and J. Chamberland, "A user-independent serial interference cancellation based coding scheme for the unsourced random access Gaussian channel," in *ITW 2017*, 2017.

[8] M. Ivanov, F. Brännström, A. G. i Amat, and P. Popovski, "Broadcast coded slotted ALOHA: A finite frame length analysis," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 651–662, Feb 2017.

[9] E. Sandgren, A. G. i Amat, and F. Brännström, "On frame asynchronous coded slotted ALOHA: Asymptotic, finite length, and delay analysis," *IEEE Transactions on Communications*, vol. 65, no. 2, pp. 691–704, Feb 2017.

[10] A. Graell i Amat and G. Liva, "Finite length analysis of irregular repetition slotted ALOHA in the waterfall region," *IEEE Communications Letters*, vol. 22, pp. 886–889, 05 2018.

[11] A. Amraoui, A. Montanari, T. Richardson, and R. Urbanke, "Finite-length scaling for iteratively decoded LDPC ensembles," *IEEE Transactions on Information Theory*, vol. 55, no. 2, pp. 473–498, Feb. 2009.

[12] M. Fereydounian, X. Chen, H. Hassani, and S. Saeedi Bidokhti, "Non-asymptotic coded slotted ALOHA," Jan 2018, https://www.seas.upenn.edu/ mferey/ms.pdf.

[13] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge University Press, 2008.

[14] N. C. Wormald, "Differential equations for random processes and random graphs," *The annals of applied probability*, pp. 1217–1235, 1995.