

Article

A Dynamic Access Probability Adjustment Strategy for Coded Random Access Schemes

Jingyun Sun ¹, Rongke Liu ^{1,*} and Enrico Paolini ^{2,*}

¹ School of Electronics and Information Engineering, Beihang University, 37 Xueyuan Road, Haidian District, Beijing 100191, China; sunjingyun@buaa.edu.cn

² Department of Electrical, Electronic, and Information Engineering, University of Bologna, via Dell'Università 50, 47522 Cesena (FC), Italy

* Correspondence: rongke_liu@buaa.edu.cn (R.L.); e.paolini@unibo.it (E.P.);
Tel.: +86-10-8233-9475 (R.L.); +39-0547-339137 (E.P.)

Received: 13 August 2019; Accepted: 25 September 2019; Published: 27 September 2019



Abstract: In this paper, a dynamic access probability adjustment strategy for coded random access schemes based on successive interference cancellation (SIC) is proposed. The developed protocol consists of judiciously tuning the access probability, therefore controlling the number of transmitting users, in order to resolve medium access control (MAC) layer congestion states in high load conditions. The protocol is comprised of two steps: Estimation of the number of transmitting users during the current MAC frame and adjustment of the access probability to the subsequent MAC frame, based on the performed estimation. The estimation algorithm exploits a posteriori information, i.e., available information at the end of the SIC process, in particular it relies on both the frame configuration (residual number of collision slots) and the recovered users configuration (vector of recovered users) to effectively reduce mean-square error (MSE). During the access probability adjustment phase, a target load threshold is employed, tailored to the packet loss rate in the finite frame length case. Simulation results revealed that the developed estimator was able to achieve remarkable performance owing to the information gathered from the SIC procedure. It also illustrated how the proposed dynamic access probability strategy can resolve congestion states efficiently.

Keywords: congestion; estimation; irregular repetition slotted ALOHA; medium access control; random access; successive interference cancellation

1. Introduction

In machine-type and Internet-of-Things (IoT) communications, users generate a large amount of bursty traffic to transmit over a shared communication medium. Coordinated multiple access schemes turn to be impractical and generally inefficient in such scenarios. For this reason, random access schemes have attracted a renewed interest, as they provide a practical way for uncoordinated users to contend for channel resource.

Pure ALOHA scheme [1] was proposed in 1968 to share a channel among a number of users sending packets as soon as they have data to transmit. Classical slotted ALOHA [2] is a distributed random access scheme in which time is divided into slots of equal duration with each transmission starting only at the beginning of a time slot. In both variants, an absence of coordination among users may lead to collisions (two or more packets are received in overlapping time windows). All packets involved in a collision are often reported as useless and are retransmitted after a random delay, according to some probability distribution, or (in the framed case) in the next frame. As a result, pure ALOHA and slotted ALOHA suffer from a throughput penalty and an under-utilization of channel resource. The optimal normalized throughput of pure ALOHA is 0.18 and the throughput of slotted ALOHA is increased to 0.37.

The expression of “coded random access” refers to a set of random access schemes that combine the packet repetition of users with successive interference cancellation (SIC) at the receiver. The first coded random access scheme is collision resolution diversity slotted ALOHA (CRDSA) [3], where each user sends two packet replicas in two random slots of the frame, and then SIC is applied to recover the collided packets in an iterative fashion. After CRDSA, CRDSA++ [4] was proposed to further improve throughput by increasing the number of packet replicas. In [5], where irregular repetition slotted ALOHA (IRSA) was proposed, the SIC-based random access process is conveniently described by a bipartite graph, establishing a bridge between the SIC procedure and the iterative erasure decoding of graph-based codes. In IRSA, the packet repetition rate is irregular from user to user and is chosen independently by each active user according to a suitably designed probability distribution. Since then, coded random access emerged as a new paradigm and has been the subject of several investigations over the past few years (e.g., [6–12] and references therein). As a result, the throughput has substantially increased which makes it a practical and efficient solution to support uncoordinated access.

Despite their numerous advantages, coded random access schemes exhibit lower critical points in traffic load. In other words, the throughput of these schemes is maximized for load values less than 1 and, for larger values of the load, it decreases very rapidly. Congestion occurs when the number of active users is greater than the receiver processing capacity. Several control methods for random access schemes have been investigated, which may be classified into two kinds: Dynamic frame length based methods and dynamic access probability based methods. In dynamic framed slotted ALOHA (DFSA) systems, the frame size is adjusted dynamically according to the estimated number of active users in order to maximize the system efficiency [13–17]. In dynamic access probability based schemes, on the other hand, an access controller is required to adjust the users access probability under high traffic loads in order to limit the number of transmitting users [18–21]. However, in [18–20], the estimation process was simply based on the status of frame slots before the application of SIC and in [21], the estimation is assumed to be ideal at the receiver. Furthermore, the proposed random access control mechanisms in [18] are based on random access schemes without SIC at the receiver, which is not applicable for coded random access schemes. In both [19,20], users directly employ the load threshold from [5], which is obtained via asymptotic analysis (frame length and user population size tending to infinity, their ratio remaining constant). When applied to the finite frame length case, asymptotic load thresholds tend to be beyond the actual critical point, which may yield considerable throughput losses.

In this paper, a dynamic access probability based strategy for coded random access schemes is proposed to resolve congestion. The proposed strategy performs two main tasks: Estimation of number of transmitting users in the current frame and the adjustment of access probability in the next frame based on the estimation results. In our previous work [22–24] techniques for a more reliable estimation of the number of transmitting users in coded random access schemes were developed and more specially, the number of transmitting users in the current frame was estimated using a posteriori information gathered throughout the SIC process. A posteriori estimation was considered for CRDSA in [22], for IRSA in [23] and for CRDSA over a packet and slot erasure channel in [24]. Notably, [22–24] were entirely focused on the estimation process, without any attempt to exploit it within a dynamic access probability adjustment protocol. The usage of a target load threshold tailored to the finite frame length case and the introduction of a state judgment to avoid not fully reliable estimation in high traffic load conditions are other original features of this manuscript.

The system model and some preliminary definitions are provided in Section 2. The estimation algorithm for the number of transmitting users in the current frame is addressed in Section 3.1, while the access probability adjustment strategy is proposed in Section 3.2. Numerical results are illustrated in Section 4 and concluding remarks are given in Section 5.

2. Preliminaries

2.1. System Model

We consider a scenario where multiple users contend for access to a single central receiver. The medium access control (MAC) layer is organized into frames and the random access scheme is a slotted one. We denote random variables by capital letters, while their realizations and deterministic quantities are denoted by lower case letters. The frame length is fixed and divided into m time slots with equal duration.

Active users are the ones who have packets to transmit. Congestion occurs when the number of active users is too large in comparison to the available resources (a more precise definition of congestion is given in Section 3.2). We use the subscript (k) to represent the index of the MAC frame. If there is no congestion or the congestion is resolved, the index (k) is re-initialized to (0) in the next frame, otherwise, it keeps counting. As such, a frame index $k \geq 1$ indicates that we are in the k -th frame of the current congestion event.

User population size is n_{pop} . The number of active users is unknown to the receiver and is modeled by a random variable N_a ; the number of active users at the beginning of frame k is $N_a^{(k)}$. No new user activates before the current congestion has been resolved. Denoting by $\Delta^{(k)}$, the number of users that are recovered while processing frame k , for $k \geq 1$ we have:

$$N_a^{(k)} = N_a^{(k-1)} - \Delta^{(k-1)}. \quad (1)$$

Transmitting users are the ones who are allowed to transmit their packets in the frame. Let $T_a^{(k)}$ be the number of transmitting users during frame k . Moreover, denote by $p_{ac}^{(k)}$ the access probability of the active users during frame k . At the beginning of the k -th frame, each active user becomes a transmitting one with probability $p_{ac}^{(k)}$, independently of other active users. Hence, the conditional expected value of $T_a^{(k)}$ is:

$$\mathbb{E}[T_a^{(k)} | n_a^{(k)}] = n_a^{(k)} p_{ac}^{(k)}. \quad (2)$$

Each transmitting user is frame- and slot- synchronous and attempts at most one packet transmission per frame.

In every frame corresponding to $k = 0$, all active users transmit their packets to the receiver, i.e., we have $p_{ac}^{(0)} = 1$ and $t_a^{(0)} = n_a^{(0)}$. The instantaneous channel load over frame k is defined as:

$$G^{(k)} = \frac{t_a^{(k)}}{m} \quad (3)$$

and represents the average number of packet transmissions per slot. The throughput over frame k is defined as:

$$T_h^{(k)} = \frac{t_a^{(k)}}{m} (1 - P_L) \quad (4)$$

representing the average number of successfully recovered packets per slot by the receiver. The quantity P_L in Equation (4) is the packet loss rate over the frame, which is expressed as:

$$P_L = 1 - \frac{\delta^{(k)}}{t_a^{(k)}}. \quad (5)$$

In IRSA, each transmitting user sends L packet replicas to slots picked uniformly at random. The number of replicas, named user degree, is a discrete random variable probability mass function (p.m.f.) $\{\Lambda_l\}$, where $\Lambda_l = P(L = l)$ is the probability that a user generates l packet replicas. Users choose their replica factor (i.e., user degree) L independently of each other, with no coordination, and the values of user degree are according to distribution $\{\Lambda_l\}$. We also represent $\{\Lambda_l\}$ in polynomial form, as $\Lambda(x) = \sum_{l=1}^{d_{\max}} \Lambda_l x^l$, where d_{\max} is the maximum number of packet replicas per user. Both information about the transmitting user index (assuming users are indexed from 1 to n_{pop}) and pointers to the slots where the other replicas have been transmitted are included in the header of each packet replica. CRDSA can be seen as IRSA with $\Lambda(x) = x^{d_{\max}}$.

In this paper, a classical collision channel model is adopted. After packet replica transmissions, each slot takes one of the following three states: Empty slot (no packet replica transmitted in that slot), singleton slot (only one packet replica transmitted in that slot), and collision slot (two or more than two packet replicas transmitted in that slot). The receiver can always correctly classify the state of each slot. Collision slots provide no information to the receiver about the number and content of collided packet replicas directly. However, as soon as the contribution of interference, generated by some transmitting users on the slot, is canceled and only one packet replica is left in it, the slot status is updated to singleton. Similarly, if all of the packet replicas transmitted in the slot are recovered by the receiver, the slot status (singleton slot or collision) is updated to empty. Packet replicas from singleton slots are always correctly received, which means that packet losses may only be generated by unresolved collisions.

After transmissions, the pointers to twin replicas in the header of the packet enable SIC at the receiver. At first, the receiver stores the content of the frame. Then, the receiver performs iteratively the following procedure, consisting of two subsequent steps:

1. Pick out the singleton slots in the frame. For each singleton slot, extract the transmitting user index, the content of the packet replica, and positions of other twin replicas. Identified users in this step become recovered users;
2. For each user recovered at step 1, remove the user's contribution of interference in the slots where the packet replicas have been transmitted. A new singleton slot will appear if, after interference cancellation, they contain only one replica.

The iterative SIC procedure terminates when all slots are empty ones, in which case SIC succeeds, or when no singleton slot can be found but collision slots still exist, in which case it fails. At the end of the SIC procedure, the residual number of empty slots in the frame is denoted by M_e , and the residual number of collided slots per frame by M_c . Obviously, we have $M_e + M_c = m$.

Example 1. With reference to Figure 1, $t_a = 4$ users transmit their packets to a frame with $m = 5$ slots. User u_1 generates three replicas of his packet, and sends them to s_1 , s_3 , and s_4 , respectively. Each of the other users generate two replicas of the corresponding packets and transmit them as illustrated in the figure. At the receiver, slots s_1 and s_4 are singleton slots and the left s_2 , s_3 , and s_5 are collision slots.

Figure 2 provides a graphical interpretation (first proposed in [5]) of the iterative SIC procedure performed on the frame of Figure 1. In the presented graph, "slot nodes" represent slots and "user nodes" represent users. In the first SIC iteration, s_1 and s_4 are singleton slots and the corresponding packet replicas are correctly received, making u_1 a recovered user. The pointer to slot s_3 , where the twin of the replica in s_1 has been transmitted, is extracted (step 1). After the interference from recovered user u_1 in slot s_3 is canceled and only one packet replica is left in s_3 , making s_3 a new singleton slot (step 2). Then a second iteration is triggered. After three SIC iterations, users u_2 and u_3 remain unrecovered, there are no singleton slots in the frame, and SIC terminates with failure.

The feedback frame configuration signal is $\{0, 1, 0, 0, 1\}$ which indicates that s_1 , s_3 and s_4 are empty slots and that s_2 and s_5 are unresolved collision slots. Receiving this feedback signal, u_2 and u_3 become aware that their packets have not been successfully received.

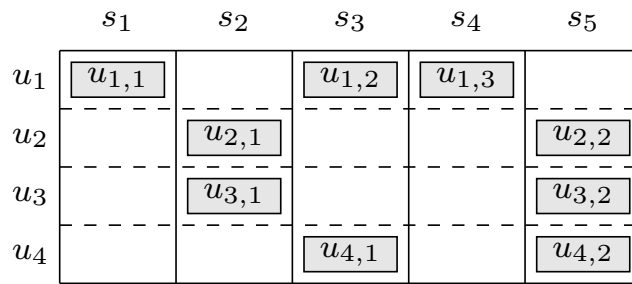


Figure 1. Example of a MAC frame with $t_a = 4$ transmitting users and $m = 5$ slots. User u_1 sends three packet replicas and the other users each send two packet replicas. Slots s_1 and s_4 are singleton slots and the left $s_2, s_3,$ and s_5 are collision slots.

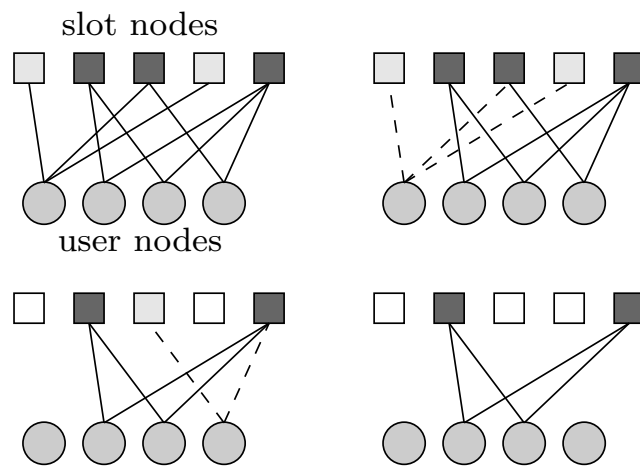


Figure 2. Example of successive interference cancellation (SIC) procedure corresponding to Figure 1. Squares correspond to slots and circles correspond to users.

2.2. Threshold Definition and Notation

Throughout the paper we define a load threshold G° as the maximum load such that the packet loss rate falls below a given target value P_L° . In other words, when the instantaneous load G is below G° , we have $P_L \leq P_L^\circ$, otherwise we have $P_L > P_L^\circ$.

In Table 1, some examples of probability distributions $\Lambda(x)$ are shown with the corresponding target load threshold values. The first two rows in the table represent CRDSA schemes, where each user transmits the same number of replicas. The last two rows represent IRSA schemes, where the number of replicas per user is irregular. The values of G° have been obtained via a Monte Carlo simulation, for MAC frame length $m = 200$ and target packet loss rate $P_L^\circ = 0.01$.

Table 1. Load threshold G° for different probability distributions $\Lambda(x)$, for MAC frame length $m = 200$, and packet loss rate target $P_L^\circ = 0.01$.

| Distribution, $\Lambda(x)$ | G° |
|---|-----------|
| $\Lambda_1(x) = x^2$ | 0.35 |
| $\Lambda_2(x) = x^4$ | 0.69 |
| $\Lambda_3(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$ | 0.705 |
| $\Lambda_4(x) = 0.25x^2 + 0.6x^3 + 0.15x^8$ | 0.76 |

Figure 3 shows the packet loss rate P_L versus instantaneous load G for the distributions in Table 1 and frame length $m = 200$. As previously remarked, the SIC process in IRSA can be described by a bipartite graph, where unresolved collisions are associated with graphical structures known, in the low-density parity-check (LDPC) coding jargon, as stopping sets. It is well known that the impact of small stopping sets on the finite-length performance is strictly related to the fraction of degree-2 variable nodes in its bipartite graph and a similar role is played by degree-2 users in IRSA. As observed in the figure, the limitation of degree-2 repetition has a better error floor performance, but a poorer waterfall performance. The detailed packet loss rate performance analysis for IRSA schemes have been addressed in [5].

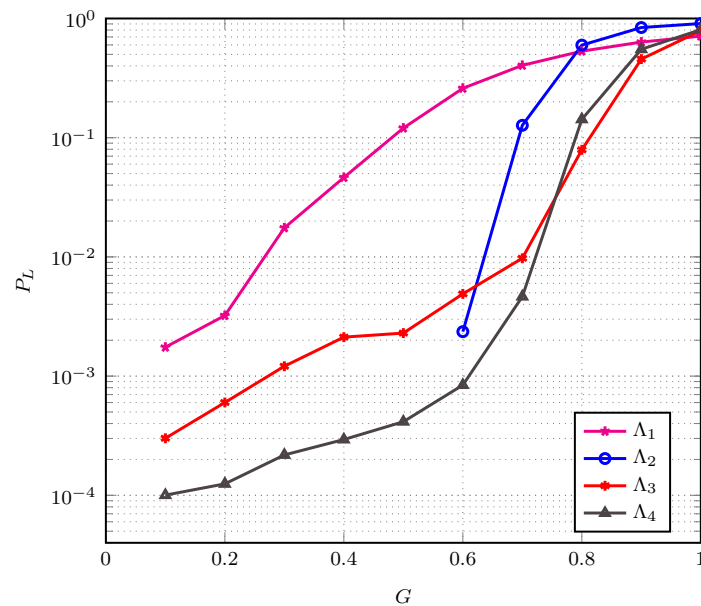


Figure 3. Packet loss rate P_L versus instantaneous load G for frame length $m = 200$ and the distributions in Table 1.

2.3. Combinatorial Parameters

We denote by $|\vec{v}| = \sum_{i=1}^n |v_i|$ the ℓ_1 norm of a real-valued vector $\vec{v} = (v_1, \dots, v_n)$. Moreover, given a second vector $\vec{w} = (w_1, \dots, w_n)$ whose elements are nonnegative integers, we use the compact notation $\vec{v}^{\vec{w}}$ for $v_1^{w_1} \dots v_n^{w_n}$.

Let $\vec{o} = (o_1, \dots, o_{d_{\max}})$ be a vector whose elements are all nonnegative integers. Let $\mathcal{M}(\vec{o}, b)$ be the set of all $|\vec{o}| \times b$ binary matrices \mathbf{M} , with rows and columns indexed from 1 to $|\vec{o}|$ and from 1 to b , respectively, that fulfill the following properties: 1. The matrix \mathbf{M} has the structure:

$$\mathbf{M} = [\mathbf{M}_1^T \ \mathbf{M}_2^T \ \dots \ \mathbf{M}_{d_{\max}}^T]^T$$

where \mathbf{M}_i has dimension $o_i \times b$ and all of its rows have Hamming weight i . 2. Every column of \mathbf{M} has Hamming weight at least 2.

Example 2. Let $\vec{o} = (o_1, o_2, o_3) = (0, 1, 3)$ and $b = 5$. Each matrix in $\mathbf{M} \in \mathcal{M}(\vec{o}, b)$ has dimension 4×5 . Its row indexes should be thought as partitioned into the two subsets $\{1\}$ and $\{2, 3, 4\}$. The row of index 1 has weight 2, and the rows of indexes 2, 3, and 4 have weight 3. Every column of \mathbf{M} has weight of at least 2. An example of matrix $\mathbf{M} \in \mathcal{M}(\vec{o}, b)$ is:

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}.$$

The following lemma provides a formal expression for the cardinality of the set $\mathcal{M}(\vec{o}, b)$.

Lemma 1. For given \vec{o} and b , let $h(\vec{o}, b)$ be the cardinality of the set $\mathcal{M}(\vec{o}, b)$. Moreover, let $\vec{x} = (x_1, x_2, \dots, x_{|\vec{o}|})$ and:

$$\vec{q} = (\underbrace{1, \dots, 1}_{o_1}, \underbrace{2, \dots, 2}_{o_2}, \dots, \underbrace{d_{\max}, \dots, d_{\max}}_{o_{d_{\max}}}). \tag{6}$$

Define the multivariate polynomials $A(\vec{x})$ and $B_{j,l}(\vec{x})$ as:

$$A(\vec{x}) = \prod_{i=1}^{|\vec{o}|} (1 + x_i) - \left(1 + \sum_{i=1}^{|\vec{o}|} x_i \right) \tag{7}$$

and:

$$B_{j,l}(\vec{x}) = \left(\sum_{i=1}^{|\vec{o}|} x_i \right)^l \left(\prod_{i=1}^{|\vec{o}|} (1 + x_i) \right)^j. \tag{8}$$

Then, we have:

$$h(\vec{o}, b) = \text{coeff}((A(\vec{x}))^b, \vec{x}^{\vec{q}}) \tag{9}$$

$$= \sum_{j=0}^b \sum_{l=0}^{b-j} \binom{b}{j} \binom{b-j}{l} (-1)^{b-j} \text{coeff}(B_{j,l}(\vec{x}), \vec{x}^{\vec{q}}) \tag{10}$$

where $\text{coeff}(P(\vec{x}), \vec{x}^{\vec{r}})$ is the coefficient of $\vec{x}^{\vec{r}}$ in the multivariate polynomial $P(\vec{x})$.

Proof. Let $\vec{c}^T = (c_1, \dots, c_{|\vec{o}|})^T$ be the generic column and define a multivariate enumerating function for valid columns (i.e., columns with weight of at least 2):

$$A(\vec{x}) = \sum_{\vec{c}: |\vec{c}| \geq 2} \vec{x}^{\vec{c}}. \tag{11}$$

It is easy to recognize that an equivalent expression for $A(\vec{x})$ is the one shown in Equation (7). This is because $(1 + x_1) \cdots (1 + x_{|\vec{o}|})$ provides the sum of all monomials in the variables $x_1, \dots, x_{|\vec{o}|}$ with a unitary coefficient, to which we subtract all monomials of degrees 0 and 1 as required by the condition of validity.

Considering now b columns and applying properties of generating functions, $\text{coeff}((A(\vec{x}))^b, \vec{x}^{\vec{w}})$ is the number of $|\vec{o}| \times b$ binary matrices such that all matrix columns are valid and such that the weight of row i is w_i . This immediately leads to Equation (9). The equivalent expression of Equation (10) is obtained by simple algebraic manipulation of the multivariate polynomial $(A(\vec{x}))^b$.

In particular, it is obtained by applying Newton's binomial formula twice and by exploiting the identity $\text{coeff}(\sum_i \alpha_i P_i(\vec{x}), \vec{x}^{\vec{w}}) = \sum_i \alpha_i \text{coeff}(P_i(\vec{x}), \vec{x}^{\vec{w}})$. \square

3. Dynamic Access Probability Algorithm

In this section we introduce the proposed multiple access strategy based on a dynamic adjustment of the users access probability. Section 3.1 addresses estimation of the number of transmitting users; Section 3.2 exploits the developed estimator to perform congestion detection and resolution via dynamic access probability adjustment.

3.1. Number of Transmitting Users Estimation

In this subsection, we exploit frame configuration information at the end of SIC to estimate the number $t_a^{(k)}$ of transmitting users in the k -th frame, when an SIC failure occurs. For the sake of notational simplicity, the superscript (k) is temporarily omitted.

The total number of transmitting users is denoted by t_a . We also denote by $t_{a,l}$ the number of such users that employ the replica factor l . Clearly, we have $t_a = \sum_{l=1}^{d_{\max}} t_{a,l}$. The vector $\vec{t}_a = (t_{a,1}, t_{a,2}, \dots, t_{a,d_{\max}})$ is referred to as transmitting users configuration at the beginning of the frame. The number of transmitted users that are recovered at the end of the SIC process is denoted by $\delta \leq t_a$. Out of these δ recovered users, $\delta_l \leq t_{a,l}$ are the ones using replica factor l , so that $\delta = \sum_{l=1}^{d_{\max}} \delta_l$. The vector $\vec{\delta} = (\delta_1, \delta_2, \dots, \delta_{d_{\max}})$ is referred to as the recovered users configuration at the end of SIC.

Hereafter we develop a compact expression for the a posteriori probability distribution of the configuration \vec{t}_a of transmitting users, given the number m_c of residual collision slots and the configuration $\vec{\delta}$ of recovered users observed at the end of SIC. This probability is denoted by $P(\vec{t}_a | m_c, \vec{\delta})$. Note that, as transmitting users pick their slots uniformly at random, it is sufficient to condition to the number of collision slots (and not to their positions in the frame). The corresponding probability distribution of the number t_a of transmitting users is given by:

$$P(t_a | m_c, \vec{\delta}) = \sum_{\vec{t}_a: |\vec{t}_a| = t_a} P(\vec{t}_a | m_c, \vec{\delta}). \quad (12)$$

A maximum a posteriori (MAP) estimator for the number of transmitting users then returns the value:

$$\hat{t}_a = \underset{t_a}{\operatorname{argmax}} P(t_a | m_c, \vec{\delta}). \quad (13)$$

Theorem 1. The a posteriori probability distribution of the configuration \vec{t}_a of the transmitting users fulfills:

$$P(\vec{t}_a | m_c, \vec{\delta}) \propto \binom{\vec{t}_a}{\vec{\delta}} \frac{h(\vec{t}_a - \vec{\delta}, m_c)}{\prod_{l=1}^{d_{\max}} \binom{m_c}{t_{a,l}}} P(\vec{t}_a) \quad (14)$$

where $h(\vec{\delta}, b)$ is given by Lemma 1, $\binom{\vec{t}_a}{\vec{\delta}} = \prod_l \binom{t_{a,l}}{\delta_l}$, and $P(\vec{t}_a)$ is the a priori probability that the transmitting users configuration equals \vec{t}_a .

Proof. From Bayes' rule we have:

$$\begin{aligned} P(\vec{t}_a | m_c, \vec{\delta}) &= \frac{P(m_c, \vec{\delta} | \vec{t}_a) P(\vec{t}_a)}{P(m_c, \vec{\delta})} \\ &\propto P(m_c, \vec{\delta} | \vec{t}_a) P(\vec{t}_a). \end{aligned} \quad (15)$$

Let $T(\vec{t}_a, m_c, \vec{\delta})$ be the number of ways in which $|\vec{t}_a|$ transmitting users with configuration \vec{t}_a can transmit their packet replicas in the frame so that, at the end of SIC, there are m_c unresolved collision slots and a recovered users configuration $\vec{\delta}$. Moreover, let $T(\vec{t}_a)$ be the number of ways in which $|\vec{t}_a|$ transmitting users with configuration \vec{t}_a can place their packet replicas in the frame. The conditional probability $P(m_c, \vec{\delta} | \vec{t}_a)$ can be expressed as:

$$P(m_c, \vec{\delta} | \vec{t}_a) = \frac{T(\vec{t}_a, m_c, \vec{\delta})}{T(\vec{t}_a)}. \quad (16)$$

The quantity $T(\vec{t}_a)$ is readily shown to be given by:

$$T(\vec{t}_a) = \prod_{l=1}^{d_{\max}} \binom{m}{l}^{t_{a,l}}. \quad (17)$$

To develop an expression for $T(\vec{t}_a, m_c, \vec{\delta})$, we proceed as follows. At the end of SIC, $|\vec{t}_a - \vec{\delta}|$ transmitting users with configuration $\vec{t}_a - \vec{\delta}$ remain unrecovered. The number of ways in which these users transmit their packet replicas to m_c slots, forming m_c collisions (at least two replicas per slot) is $h(\vec{t}_a - \vec{\delta}, m_c)$. If we let $g(\vec{\delta}, m_c)$ be the number of ways in which $|\vec{\delta}|$ transmitting users with configuration $\vec{\delta}$ can place their packet replicas in a frame with $m - m_c$ free slots and m_c unresolvable collision slots, so that SIC can recover all of them, we can write (no formal expression for $g(\vec{\delta}, m_c)$ is provided because this parameter, not depending on \vec{t}_a does not play any role in the estimation process of Equation(13)):

$$T(\vec{t}_a, m_c, \vec{\delta}) = \binom{\vec{t}_a}{\vec{\delta}} \binom{m}{m_c} h(\vec{t}_a - \vec{\delta}, m_c) g(\vec{\delta}, m_c). \quad (18)$$

Incorporating Equation (17) and Equation (18) into Equation (16) and then Equation (16) into Equation (15), and omitting all terms not depending on \vec{t}_a , we obtain Equation (14). \square

Although Equations (13) and (14) define an exact MAP estimator, computing $h(\vec{t}_a - \vec{\delta}, m_c)$ turns out to be a complex task, becoming already intractable for frame sizes in the order of a few tens. For this reason we employ an approximated MAP estimator. In the approximation, all packet replicas, even from the same user, are regarded as distinguishable packets. Equivalently, each user chooses l slots with replacement. In this approximate setting, we have

$$P(\vec{t}_a | m_c, \vec{\delta}) \propto \binom{\vec{t}_a}{\vec{\delta}} \frac{h((\sum_{l=1}^{d_{\max}} (t_l - \delta_l)l), m_c)}{m^{\sum_{l=1}^{d_{\max}} t_l l}} P(\vec{t}_a) \quad (19)$$

where $(\sum_{l=1}^{d_{\max}} (t_l - \delta_l)l)$ represents a vector with only one element, corresponding to $\vec{\delta} = (o_1)$ in $h(\vec{\delta}, b)$. The value of $h((\sum_{l=1}^{d_{\max}} (t_l - \delta_l)l), m_c)$ is the number of ways in which $\sum_{l=1}^{d_{\max}} (t_l - \delta_l)l$ packet replicas are sent to m_c slots, such that each slot receives not less than 2 packet replicas [25]; $m^{\sum_{l=1}^{d_{\max}} t_l l}$ is the total number of ways in which $\sum_{l=1}^{d_{\max}} t_l l$ packet replicas can be accommodated into the m slots.

As an estimator performance measure we consider the MSE, defined as:

$$M_c = \mathbb{E}[\epsilon^2] \quad (20)$$

where $\epsilon = \hat{t}_a^{(k)} - t_a^{(k)}$ is the estimation error.

3.2. Access Probability Adjustment Strategy

We say that we have a congestion on frame k whenever

$$n_a^{(k)} > G^\circ m, \quad (21)$$

where we recall that $n_a^{(k)}$ is the number users that are active on frame k . Our purpose is to exploit the developed estimator to detect congestion states and dynamically adjust the users access probability to improve overall efficiency. Congestion states are resolved by tuning the access probability to control the number of transmitting users in the next frame.

The proposed scheme is based on the definition of three possible states for a frame, namely:

- **Not fully reliable estimate.** In high load conditions, SIC typically stops prematurely with a relatively small number of recovered users. We say that the estimate $\hat{t}_a^{(k)}$ is not fully reliable when the number of users recovered by processing the frame is smaller than the number of users that could not be recovered:

$$\delta^{(k)} < t_a^{(k)} - \delta^{(k)} \quad (22)$$

or, equivalently,

$$\delta^{(k)} < t_a^{(k)} / 2. \quad (23)$$

- **Congestion with reliable estimate.** The number of active users is above threshold $G^\circ m$, but the number of users recovered by processing the frame is not less than the number of users that could not be recovered:

$$n_a^{(k)} > G^\circ m \quad \text{and} \quad \delta^{(k)} \geq t_a^{(k)} - \delta^{(k)}. \quad (24)$$

- **No congestion.** The number of active users is below threshold $G^\circ m$:

$$n_a^{(k)} \leq G^\circ m. \quad (25)$$

In the first case, a large number of transmitting users is unrecovered, and the packet loss rate is larger than 0.5. As illustrated in the numerical results section, the estimation MSE increases with the number of transmitting users and the estimate is therefore regarded as not suitable to design the access probability p_{ac} in the subsequent frame. In contrast, in the last two cases the access probability in the next frame is calculated directly by employing the estimate of the number of transmitting users.

In the generic frame k , after all transmitting users have performed the transmission of their packet replicas, the receiver performs the SIC procedure. At the end of SIC, the receiver executes the procedure described in Algorithm 1. This procedure is executed regardless of the SIC termination status (success or failure). An explanation of Algorithm 1 is provided in the following.

Algorithm 1: Receiver procedure

```

1 if  $m_c^{(k)} > 0$  &&  $\frac{P(t_a^{(k)} > 2\delta^{(k)} | m_c^{(k)}, \bar{\delta}^{(k)})}{P(t_a^{(k)} \leq 2\delta^{(k)} | m_c^{(k)}, \bar{\delta}^{(k)})} > 1$  then
2    $p_{ac}^{(k+1)} \leftarrow \frac{G^\circ m}{n_{\text{pop}} - \sum_{i=0}^k \delta^{(i)}}$ ;
3    $k \leftarrow k + 1$ ;
4 else
5   if  $m_c^{(k)} == 0$  then
6      $\hat{t}_a^{(k)} = \delta^{(k)}$ ;
7   else
8     calculate  $\hat{t}_a^{(k)}$  according to Equation (19)
9   end
10   $\hat{n}_a^{(k)} = \hat{t}_a^{(k)} / p_{ac}^{(k)}$ ;
11  if  $\hat{n}_a^{(k)} > G^\circ m$  then
12     $p_{ac}^{(k+1)} \leftarrow \frac{G^\circ m}{\hat{n}_a^{(k)} - \delta^{(k)}}$ ;
13     $k \leftarrow k + 1$ ;
14  else
15     $p_{ac}^{(1)} \leftarrow 1$ ;
16     $k \leftarrow 0$ ;
17  end
18 end
19 broadcast  $k$ ,  $p_{ac}^{(k)}$ , and  $\mathcal{C}^{(k)}$  to the users;

```

The first step (line 1) consists of detecting whether Equation (23) is fulfilled or not. When SIC succeeds ($m_c^{(k)} = 0$), the estimation is perfect. The algorithm jumps to line 6 and simply sets $\hat{t}_a^{(k)} = \delta^{(k)}$. In case of an SIC failure ($m_c^{(k)} > 0$), the algorithm applies a two-hypotheses MAP detector, whose development is presented in Appendix A, to decide whether Equation (23) holds (in which case estimation is considered unreliable) or not. Concretely, if

$$\frac{P(t_a^{(k)} > 2\delta^{(k)} | m_c^{(k)}, \bar{\delta}^{(k)})}{P(t_a^{(k)} \leq 2\delta^{(k)} | m_c^{(k)}, \bar{\delta}^{(k)})} > 1, \quad (26)$$

then Equation (23) is assumed to hold and the estimation Equation (19) is regarded as not reliable enough. Otherwise, the estimate $\hat{t}_a^{(k)}$ is employed to design the access probability in the next frame.

When Equation (26) is satisfied, a ‘not fully reliable estimate’ state is detected and the number of transmitting users is detected to be large enough to create a congestion but the relatively large estimation MSE prevents from relying on $\hat{t}_a^{(k)}$ to reliably adjust the access probability in the next frame. At the end of frame k a number $\sum_{i=0}^k \delta^{(i)}$ of active users have been recovered since the beginning of congestion. Therefore, at the beginning of the subsequent frame, the number of unresolved active users fulfills $n_a^{(k+1)} \leq n_{\text{pop}} - \sum_{i=0}^k \delta^{(i)}$. To make the expected number of transmitting users in the subsequent frame below the target number $G^\circ m$, we set the access probability according to (line 2)

$$p_{ac}^{(k+1)} = \frac{G^\circ m}{n_{\text{pop}} - \sum_{i=0}^k \delta^{(i)}}. \quad (27)$$

This way, the conditional expected number of transmitting users in the next frame is

$$\begin{aligned}\mathbb{E}[T_a^{(k+1)} | n_a^{(k+1)}] &= n_a^{(k+1)} p_{ac}^{(k+1)} \\ &= \frac{n_a^{(0)} - \sum_{i=0}^k \delta^{(i)}}{n_{\text{pop}} - \sum_{i=0}^k \delta^{(i)}} G^\circ m,\end{aligned}\quad (28)$$

where $n_a^{(0)} - \sum_{i=0}^k \delta^{(i)}$ represents the actual number of unrecovered active users at the beginning of frame $k + 1$.

When the estimation result is detected to be reliable, an acceptable estimation MSE is assumed by the receiver, which exploits $\hat{t}_a^{(k)}$ (equal to $\delta^{(k)}$ in case of a SIC success or by Equation (19) in case of a failure) to obtain an estimate of the number of active users on frame k . Specifically, the receiver performs (line 10):

$$\hat{n}_a^{(k)} = \hat{t}_a^{(k)} / p_{ac}^{(k)}.\quad (29)$$

The estimate $\hat{n}_a^{(k)}$ is compared with the threshold $G^\circ m$ (line 11). If $\hat{n}_a^{(k)} > G^\circ m$ the receiver declares a congestion with a reliable estimate state. The system is suffering from congestion, but most of (or all of) transmitting users have been recovered by SIC. The relatively low estimation MSE allows confidently using $\hat{n}_a^{(k)}$ to set the access probability for the next frame. If the access probability is kept unchanged in the subsequent frames, the number of transmitting users will deviate progressively from the target $G^\circ m$, leading to a low throughput. To make efficient use of channel resources, we increase the access probability in such a way as to maintain the number of transmitting users close to the target $G^\circ m$ in the next frame. From Equation (1), the estimated number of unrecovered active users at the beginning of the frame $k + 1$ is $\hat{n}_a^{(k+1)} = \hat{n}_a^{(k)} - \delta^{(k)}$. The target conditional expected number of transmitting users in frame $k + 1$ is

$$\mathbb{E}[T_a^{(k+1)} | n_a^{(k+1)}] = G^\circ m.\quad (30)$$

Thus, the access probability over frame $k + 1$ is set to (line 12):

$$p_{ac}^{(k+1)} = \frac{G^\circ m}{\hat{n}_a^{(k)} - \delta^{(k)}}.\quad (31)$$

If $\hat{n}_a^{(k)} < G^\circ m$, a no congestion state is detected. The frame index k is re-initialized to 0 and the users access probability is set to be 1 (lines 15 and 16).

As a last step (line 19), the receiver broadcasts to the users the index of the next frame (index of the current frame increased by 1 if a congestion is detected and 0 otherwise), the access probability to be employed by active users in the next frame, and the list of indexes of collision slots at the end of SIC in the current frame. Upon receiving feedback from the receiver, users behave as follows:

- If $k > 0$ (congestion), in the next frame each backlogged user attempts access to the frame with probability equal to the new access probability. Each non-backlogged user is prevented from transmitting new packets;
- If $k = 0$ (no congestion), users that are in a backlog state retransmit their packet. Users that are not backlogged take their normal access activity.

Users are updated by the receiver about congestion or no congestion simply through the index k . Moreover, each of them knows whether or not it is in a backlog state simply by looking at the list of collision slot indexes $\mathcal{C}^{(k)}$. Note that, if $k = 0$ (no congestion) is broadcasted by the receiver, this does not necessarily mean that SIC has succeeded as there may be a small number of users unrecovered

even though the system is not suffering from congestion. In this case, we simply let backlogged users retransmit packet replicas with probability 1 in the subsequent frame, together with possible fresh replicas from newly activated users.

4. Numerical Results

This section is organized into two subsections. In Section 4.1 we show results on the estimation of the number of transmitting users, while in Section 4.2 we address the performance achieved by the proposed scheme.

4.1. Estimation of Transmitting Users

In this section, we present Monte Carlo simulation results using the approximated estimator discussed at the end of Section 3.1. Let the frame length be $m = 200$ and the user population size be $n_{\text{pop}} = 400$. Users are assumed to activate independently of each other at the beginning of every new frame. In each run, after users transmissions, SIC is applied at the receiver and then the developed approximated estimator is applied.

Figure 4 shows the average throughput and throughput standard deviation versus the instantaneous load G for IRSA with $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$ [5]. The maximum average throughput is achieved at a value of G that is approximately equal to 0.8. However, the realizations of the per-frame throughput fluctuate around its statistical mean, the throughput standard deviation representing a reliable measure of the bobbing range (i.e., dispersion). A large standard deviation makes the average throughput a not fully meaningful parameter since, due to the per-frame throughput fluctuations, we have a higher probability that the system falls into a not fully reliable estimate state. In this respect, the peak average throughput is not necessarily a good working point, as the statistical mean alone is not able to capture the probability of falling into such an “outage” state.

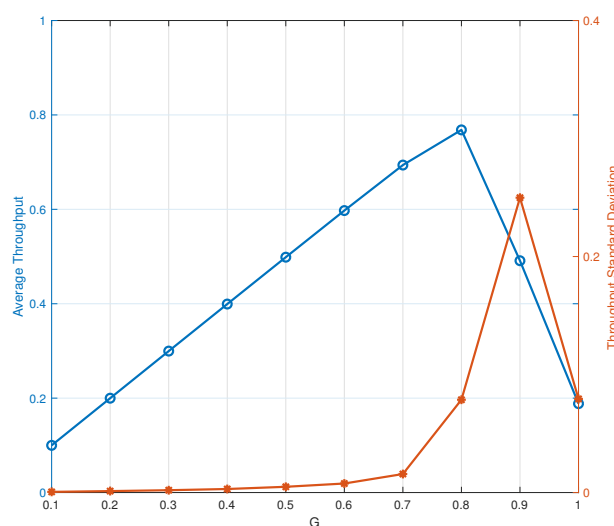


Figure 4. Average throughput and throughput standard deviation versus the instantaneous load G for IRSA with $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$.

Figure 5 shows the estimation performance after SIC iterations, letting the SIC-based receiver run until no active user is recovered. As a comparison, we also consider the estimation performance using the frame configuration before SIC iterations, which is reviewed in Appendix B. In the figure, the solid line is relevant to the proposed estimation making use of the frame configuration and recovered users configuration after SIC iterations. Moreover, a dashed line corresponds to the estimation based on the initially received frame, before SIC is applied. As observed in the figure, the proposed estimation

algorithm is able to reduce the MSE effectively over the whole range of G values. It is also worth noting that the performance of the proposed estimator relies on the SIC performance. In low load conditions, the SIC procedure stops with a large number of users recovered, so in this region the proposed estimation algorithm is more effective. In contrast, in high load conditions SIC almost always stops prematurely, recovering a small number of users, leading the proposed estimation algorithm to be less effectively.

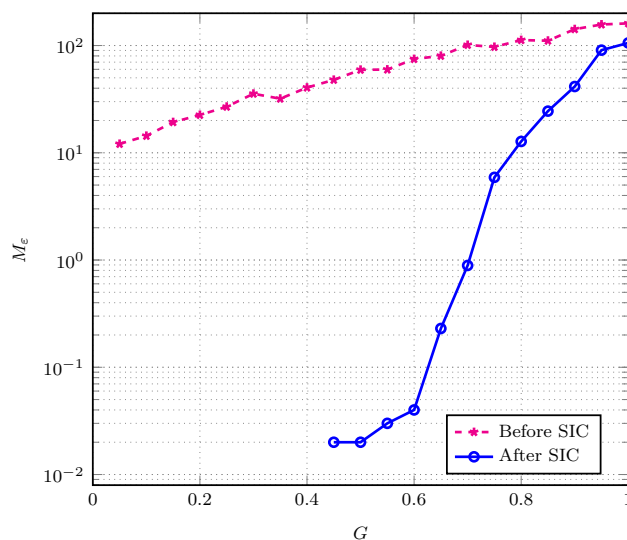


Figure 5. Mean squared error (MSE) versus G for IRSA with $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$.

4.2. Dynamic Access Probability Simulation Results

In this subsection, we present the simulation results for dynamic access probability based coded random access schemes using the mentioned estimation methods. The frame length is $m = 200$ and the user population size is $n_{\text{pop}} = 2000$. Moreover, the considered IRSA distribution is $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$. Each non-backlogged user activates, independently of the other users, with probability $\pi = 0.8$ at the beginning of every frame with $k = 0$. At the first frame, there are no backlogged users. The target traffic threshold G° is set to 0.65, 0.705, 0.80, and 0.938 respectively, of which $G^\circ = 0.708$ is associated with $P_L^\circ = 0.01$ and $G^\circ = 0.938$ is the asymptotic threshold of the considered IRSA distribution [5]. The initial access probability is $p_{ac}^{(0)} = 1$. We analyzed the system performance, during congestion resolution periods, through numerical simulations. Every simulation consisted of a sufficiently large number of runs and, in each run, the simulation was stopped when the congestion was resolved.

As a benchmark, consider transmission without any dynamic access probability adjustment process. The expected number of active users (transmitting users) in the initial frame is 1600. The average repetition rate is 3.6, corresponding to an expected number of 6480 packet replicas transmitted over the 200 slots. At the receiver, we have a vanishing probability to find singleton slots capable of triggering the SIC process. Without dynamic access probability adjustment, the packet loss rate becomes very close to 1 and the throughput very close to 0, meaning that almost no users are recovered in the subsequent frames, making system congestion unresolvable.

Figures 6–8 show that the proposed access probability algorithm works well to resolve congestion. The users access probability is adjusted dynamically to track the number of active users. At frame 1, the access probability is decreased quickly to avoid working in the high load region. In this way, the estimator can provide a reliable estimate at the end of the frame and the receiver is able to perform an accurate access probability design for the users in the next frame. Then the access probability is adjusted dynamically to make the number of transmitting users around the target $G^\circ m$. It is increased

slowly as some users are recovered by the receiver in each transmission. Each curve is plotted up to the maximum value of k for which congestions remain unresolved, which is different for the different choices of the target load threshold.

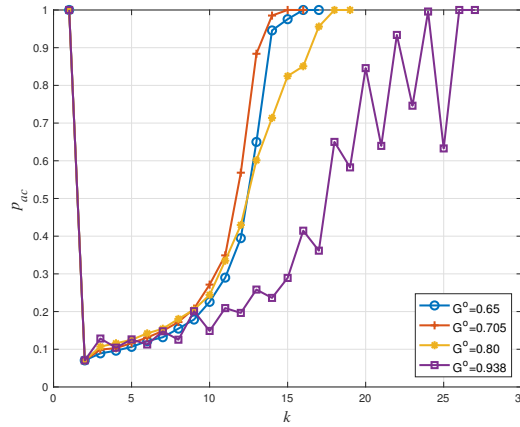


Figure 6. Access probability p_{ac} versus frame index k for IRSA with $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$.

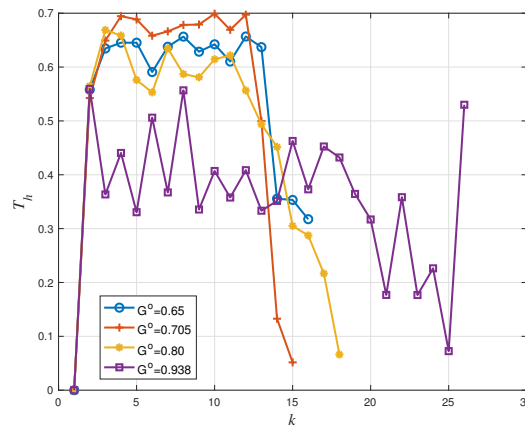


Figure 7. Throughput performance T_h versus frame index k for IRSA with $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$.

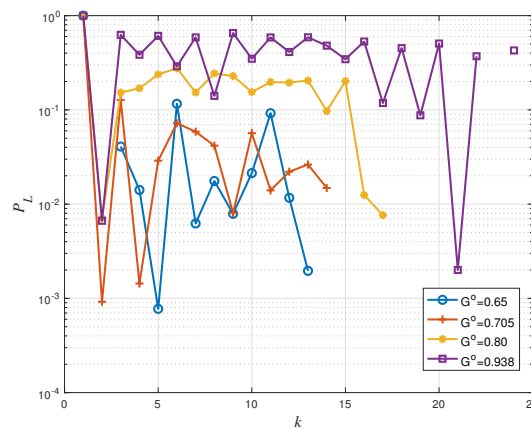


Figure 8. Packet loss rate P_L versus frame index k for IRSA with $\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$.

Back to Figure 3, we have seen that the IRSA scheme tends to show a packet loss rate floor at low offered traffic regimes, the floor appearing around $P_L = 10^{-2}$ (corresponding to $G = 0.705$) for

$\Lambda(x) = 0.5x^2 + 0.28x^3 + 0.22x^8$. For larger values of G (corresponding to the waterfall packet loss rate region), the packet loss rate increases rapidly: A $P_L \simeq 0.08$ is achieved at $G = 0.80$ and a $P_L \simeq 0.5$ is achieved at $G = 0.938$. Consequently, in Figure 8, systems with target $G^\circ = 0.65$ and $G^\circ = 0.705$ have a similar packet loss rate performance, and they perform better than those with target $G^\circ = 0.80$ and $G^\circ = 0.938$. Furthermore, due to estimation errors and to fluctuations of the actual number of transmitting users, we observe a minor packet loss rate deviation between Figure 8 and Figure 3. For example, in Figure 8, the packet loss rate with target $G^\circ = 0.80$ is around 0.2, while in Figure 3, the packet loss rate at $G^\circ = 0.80$ is approximately equal to 0.008.

As a final remark, recall that the throughput is defined as $G(1 - P_L)$. The influence of P_L at $G \leq 0.705$ is small, so that the per-frame throughput $T_h^{(k)}$ is approximately equal to the instantaneous load G . That is why in Figure 7, the throughput performance with target $G^\circ = 0.705$ is better than that with target $G^\circ = 0.65$. However, for the cases $G^\circ = 0.8$ and $G^\circ = 0.938$, the influence of P_L can not be ignored any more. The system performance is worse even though the load target G° is higher, since the packet loss rate is now considerably higher.

5. Conclusion

In this paper, we proposed a technique to estimate the number of transmitting users in each frame of an IRSA-based coded random access system. The estimated number of transmitting users in the current frame was exploited to adjust the users access probability in the next frame. Frame configuration information as well as recovered users configuration information at the end of the SIC procedure were employed to make the estimation more accurate. Numerical results revealed how the derived dynamic access probability strategy could resolve congestion efficiently, with a stable throughput and a target packet loss rate performance for a proper choice of the parameter G° . Interesting directions of investigation include the exact efficient evaluation of the $h(\vec{\delta}, b)$ function (addressed in Lemma 1), to make the optimum estimator applicable to large communication networks. Adjusting the frame length dynamically in situations of slowly varying traffic load over a large scale is another direction of investigation that, to the best of the authors' knowledge, has not been so far addressed in the coded random access context.

Author Contributions: All the authors contributed extensively to the work presented in this paper and to writing the paper itself. J.S. and E.P. conceived the idea and developed the proposed approaches. E.P. gave advice on the research and helped in editing the paper. E.P. and R.L. improved the quality of the manuscript and completed revision.

Funding: This research was funded by the China Scholarship Council (grant No. 201706020024).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Justification and Implementation of Equation (26)

This appendix justifies and addresses the implementation of the two-hypotheses MAP detection rule Equation (26). Let the two hypotheses be H_0 and H_1 . Moreover, let r represent the observation. The optimum detection rule consists of making the decision \hat{H}_0 when $P(H_0|r) > P(H_1|r)$ and of making the decision \hat{H}_1 otherwise.

In our case, H_0 corresponds to a 'not fully reliable estimate' state (satisfied Equation (23)); H_1 corresponds to a 'reliable estimate' state (Equation (23) not holding). The observation is $(m_c^{(k)}, \vec{\delta}^{(k)})$. Hence, we have:

$$\frac{P(t_a^{(k)} > 2\delta^{(k)} | m_c^{(k)}, \vec{\delta}^{(k)})}{P(t_a^{(k)} \leq 2\delta^{(k)} | m_c^{(k)}, \vec{\delta}^{(k)})} \underset{\hat{H}_1}{\overset{\hat{H}_0}{\gtrless}} 1, \quad (\text{A1})$$

where $P(t_a^{(k)} > 2\delta^{(k)} | m_c^{(k)}, \vec{\delta}^{(k)}) = \sum_{t_a^{(k)}: t_a^{(k)} > 2\delta^{(k)}} P(t_a^{(k)} | m_c^{(k)}, \vec{\delta}^{(k)})$ and $P(t_a^{(k)} \leq 2\delta^{(k)} | m_c^{(k)}, \vec{\delta}^{(k)}) = \sum_{t_a^{(k)}: t_a^{(k)} \leq 2\delta^{(k)}} P(t_a^{(k)} | m_c^{(k)}, \vec{\delta}^{(k)})$. The probability $P(t_a^{(k)} | m_c^{(k)}, \vec{\delta}^{(k)})$ can be expressed as:

$$P(t_a^{(k)} | m_c^{(k)}, \vec{\delta}^{(k)}) = \sum_{\vec{t}_a: |\vec{t}_a^{(k)}| = t_a^{(k)}} P(\vec{t}_a^{(k)} | m_c, \vec{\delta}^{(k)}), \quad (\text{A2})$$

where $P(\vec{t}_a^{(k)} | m_c, \vec{\delta}^{(k)})$ comes from the estimator Equation (14).

Appendix B. Estimation Using Collision Slots before SIC

The user's repetition rate is $\Lambda(x)$. Define $\Lambda'(1)$ as the average user repetition rate given by $\Lambda'(1) = \sum_l l \Lambda_l$. It is easy to verify that the probability that a generic user sends a packet replica within a given slot is $\Lambda'(1)/m$. As the users send packet replicas randomly, the slot degree distribution is binomially distributed. The probability that a slot has l collided users is given by:

$$\Psi_l = \binom{t_a}{l} \left(\frac{\Lambda'(1)}{m} \right)^l \left(1 - \frac{\Lambda'(1)}{m} \right)^{t_a - l}. \quad (\text{A3})$$

Before SIC iterations, the probability p_e that a given slot is empty, the probability p_s that a given slot is singleton and the probability p_c that a given slot is a collision one can be expressed respectively as:

$$p_e = \left(1 - \frac{\Lambda'(1)}{m} \right)^{t_a}, \quad (\text{A4})$$

$$p_s = t_a \frac{\Lambda'(1)}{m} \left(1 - \frac{\Lambda'(1)}{m} \right)^{t_a - 1} \quad (\text{A5})$$

and:

$$p_c = 1 - p_e - p_s. \quad (\text{A6})$$

An estimation for T_a using frame configuration before SIC performs:

$$\hat{t}_a = \operatorname{argmax}_{t_a} P(t_a | w_c), \quad (\text{A7})$$

where w_c is the number of collision slots before SIC iterations. Following Bayes' rule, $P(t_a | w_c)$ may be developed as:

$$\begin{aligned} P(t_a | w_c) &\propto P(w_c | t_a) P(t_a) \\ &\propto p_c^{w_c} (1 - p_c)^{m - w_c}. \end{aligned} \quad (\text{A8})$$

References

1. Abramson, N. The ALOHA system: Another alternative for computer communications. In Proceedings of the Fall Joint Computer Conference, New York, NY, USA, 17–19 November 1970; pp. 281–285.
2. Roberts, L.G. ALOHA packet systems with and without slots and capture. *ACM SIGCOM Comput. Commun. Rev.* **1975**, *5*, 28–42. [\[CrossRef\]](#)
3. Casini, E.; De Gaudenzi, R.; Herrero, O.D.R. Contention resolution diversity slotted ALOHA (CRDSA): An enhanced random access scheme for satellite access packet networks. *IEEE Trans. Wirel. Commun.* **2007**, *6*, 1408–1419. [\[CrossRef\]](#)

4. De Gaudenzi, R.; Herrero, O.D.R. Advances in random access protocols for satellite networks. In Proceedings of the 2009 International Workshop on Satellite and Space Communications, Tuscany, Italy, 9–11 September 2009; pp. 331–336.
5. Liva, G. Graph-based analysis and optimization of contention resolution diversity slotted ALOHA. *IEEE Trans. Commun.* **2011**, *59*, 477–487. [[CrossRef](#)]
6. Amat, A.G.; Liva, G. Finite Length Analysis of Irregular Repetition Slotted ALOHA in the Waterfall Region. *IEEE Commun. Lett.* **2018**, *22*, 886–889. [[CrossRef](#)]
7. De Gaudenzi, R.; Herrero, O.D.R.; Gallinaro, G.; Cioni, S.; Arapoglou, P.D. Random access schemes for satellite networks, from VSAT to M2M: A survey. *Int. J. Satellite Commun. Netw.* **2018**, *36*, 66–107. [[CrossRef](#)]
8. Alvi, S.; Durrani, S.; Zhou, X. Enhancing CRDSA with Transmit Power Diversity for Machine-Type Communication. *IEEE Trans. On Vehic. Tech.* **2018**, *67*, 7790–7794. [[CrossRef](#)]
9. Paolini, E.; Liva, G.; Chiani, M. Coded slotted ALOHA: A graph-based method for uncoordinated multiple access. *IEEE Trans. Inf. Theory* **2015**, *61*, 6815–6832. [[CrossRef](#)]
10. Paolini, E.; Stefanović, Č.; Liva, G.; Popovski, P. Coded random access: Applying codes on graphs to design random access protocols. *IEEE Commun. Mag.* **2015**, *53*, 144–150. [[CrossRef](#)]
11. Taghavi, A.; Vem, A.; Chamberland, J.-F.; Narayanan, K.R. On the design of universal schemes for massive uncoordinated multiple access. In Proceedings of the 2016 IEEE International Symposium on Information Theory (ISIT), Barcelona, Spain, 10–15 July 2016; pp. 345–349.
12. Sandgren, E.; Amat, A.G.; Brännström, F. On frame asynchronous coded slotted ALOHA: Asymptotic, finite length, and delay analysis. *IEEE Trans. Commun.* **2017**, *65*, 691–704. [[CrossRef](#)]
13. Schoute, F. Dynamic frame length ALOHA. *IEEE Trans. Commun.* **1983**, *31*, 565–568. [[CrossRef](#)]
14. Chen, W.-T. An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA. *IEEE Trans. Autom. Sci. Eng.* **2009**, *6*, 9–15. [[CrossRef](#)]
15. Eom, J.-B.; Lee, T.-J. Accurate tag estimation for dynamic framed-slotted ALOHA in RFID systems. *IEEE Commun. Lett.* **2010**, *14*, 60–62. [[CrossRef](#)]
16. Wu, H.; Zeng, Y. Bayesian tag estimate and optimal frame length for anti-collision ALOHA RFID system. *IEEE Trans. Autom. Sci. Eng.* **2010**, *7*, 963–969. [[CrossRef](#)]
17. Zanella, A. Estimating collision set size in framed slotted ALOHA wireless networks and RFID systems. *IEEE Commun. Lett.* **2012**, *16*, 300–303. [[CrossRef](#)]
18. Rivero Angeles, M.E.; Lara Rodriguez, D.; Cruz-Perez, F.A. Random-access control mechanisms using adaptive traffic load in ALOHA and CSMA strategies for EDGE. *IEEE Trans. Vehic. Tech.* **2005**, *54*, 1160–1186. [[CrossRef](#)]
19. Lee, M.W.; Lee, J.K.; Lim, J.S. R-CRDSA: Reservation-Contention Resolution Diversity Slotted ALOHA for Satellite Networks. *IEEE Commun. Lett.* **2012**, *16*, 1576–1579. [[CrossRef](#)]
20. Noh, H.J.; Lee, J.K.; Lim, J.S. Performance evaluation of access control for CRDSA and R-CRDSA under high traffic load. In Proceedings of the 2013 IEEE Military Communications Conference, San Diego, CA, USA, 18–20 November 2013; pp. 1365–1370.
21. Sun, J.; Liu, R. Irregular Repetition Slotted ALOHA with Priority (P-IRSA). In Proceedings of the 2016 IEEE Vehicular Technology Conference (VTC Spring), Nanjing, China, 15–18 May 2016; pp. 1–5.
22. Sun, J.; Liu, R.; Paolini, E. Detecting the Number of Active Users in Coded Random Access Systems. In Proceedings of the 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, Italy, 9–12 September 2018; pp. 1–7.
23. Sun, J.; Liu, R.; Paolini, E. Detecting the Number of Active Users in IRSA Access Protocols. In Proceedings of the 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, Italy, 9–12 September 2018; pp. 1972–1976.
24. Sun, J.; Liu, R.; Paolini, E. Unrecovered Users Distribution in Coded Random Access Systems with Erasures. In Proceedings of the 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–6.
25. Schmetterer, L. An introduction to combinatorial analysis by J. Riordan. *Phys. Today* **1959**, *12*, 158.

