Received: 2 August 2019 | Revised: 22 June 2020 | Accepted: 30 June 2020
DOI: 10.1002/sat.1370

RESEARCH ARTICLE

WILEY

Complexity analysis for recent ALOHA random access techniques in satellite communications

Selma Zamoum[1,3] | Jérôme Lacan[2] | Marie-Laure Boucheret[3] | Jean-Baptiste Dupe[4] | Mathieu Gineste[5]

[1]TéSA, Toulouse, France
[2]ISAE-SUPAERO, University of Toulouse, Toulouse, France
[3]INPT/ENSEEIHT, University of Toulouse, Toulouse, France
[4]CNES, Toulouse, France
[5]Thales Alenia Space, Toulouse, France

Correspondence
Selma Zamoum, TéSA, Toulouse, France.
Email: selma.zamoum@tesa.prd.fr

Summary

In this paper, we study the the complexity of packet localization at reception, for recent synchronous Random Access (RA) techniques based on the protocol ALOHA for satellite communications. The promising CRDSA (Contention Resolution Diversity Slotted ALOHA) offers better throughput, in comparison to the traditional slotted ALOHA protocols, thanks to the use of Successive Interference Cancellation (SIC) along with multireplica transmission. MARSALA (Multi-replicA decoding using corRelation baSed locALizAtion) is one of the many variants and enhancement schemes of CRDSA that have been proposed in the literature. It is applied to CRDSA each time a decoding deadlock situation is reached (when no packets can be retrieved by CRDSA). MARSALA first localizes the replicas of collided packets on a chosen reference time slot using correlations. Then it performs coherent signal combination of packet replicas prior to decoding. However, despite the good performance offered by MARSALA, its localization process adds a significant complexity to the receiver in terms of correlation operations. R-SPOTiT (Random Shared POsition Technique for Interfered random Transmissions) mitigates this complexity by introducing a shared information between the receiver and each of the transmitters, about all potential packets' locations on the frame, without any additional signaling overhead. We focus in this paper on the analysis of the total number of correlations which are needed to localize packets' replicas for both MARSALA and R-SPOTiT, with a single or with multiple Gold preambles. This should include preamble detection operations that are performed at CRDSA with a coarse and fine tracking. The results show that the most suitable system to use is the multi-preamble R-SPOTiT with two preambles.

KEYWORDS

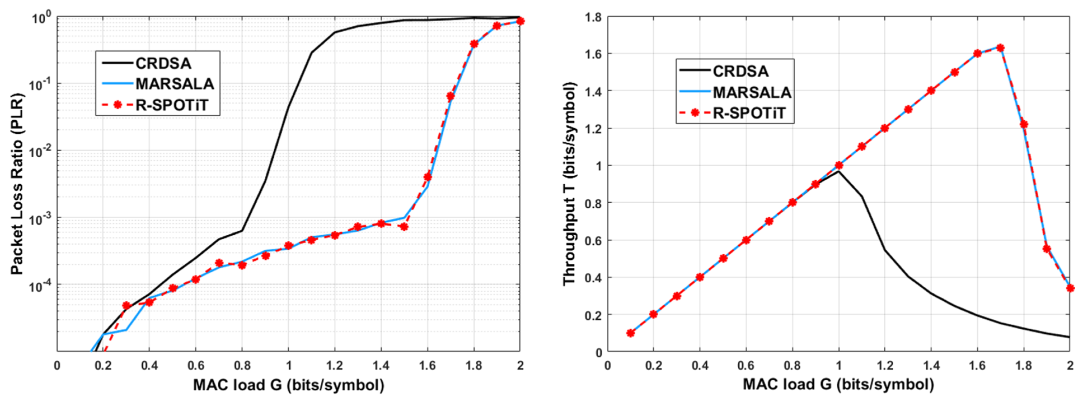
packet localization, Random Access, receiver's complexity, satellite communication, slotted ALOHA protocols

1 | INTRODUCTION

The satellite access channel when there are multiple transmitters has long been a subject of study due to its characteristics such as the propagation delay, multipath fading, and shadowing effect that are significant, compared to terrestrial transmissions. Random Access (RA) over satellite has in recent past become an important subject of study, especially as Demand Assignment happens to be unsuitable for traffic profiles with sporadic and short packets transmissions like HTTP requests or IoT. Thus, multiple solutions based on synchronous and asynchronous[1-3] transmissions with spread spectrum[4-6] or/and with ALOHA protocol[7] have been proposed. In the scope of this paper, we target low rate interactive applications with sporadic transmissions, such as Supervisory Control and Data Acquisition (SCADA) systems, with fixed terminals and geostationary satellites. Also, we have chosen to work in the synchronous environment that has been defined in the DVB-RC2 (Digital Video Broadcasting–Return Channel via Satellite) standard. CRDSA (Contention Resolution Diversity Slotted ALOHA)[8] using multiple replicas per packet at transmission and Successive Interference Cancellation (SIC) at reception has emerged as a leading RA technique in the standard. Different aspects and parameters have been assessed by various ALOHA based methods in order to offer better performance in terms of PLR (Packet Loss Ratio) and Throughput. Among what is to be found in the literature, IRSA (Irregular Repetition Slotted ALOHA),[9] which took part in the DVB-RCS2 standard[10] along with CRDSA, proposed a number of replicas that can vary from one transmitter to another. However, a comparison between IRSA with a maximum number of replicas equal to 8 and CRDSA with 3 replicas in Mengali et al.,[4] using the same system parameters in terms of FEC (Forward Error Correction) code rate, packets' size, modulation scheme, and with equipowered packets showed that CRDSA slightly outperforms IRSA with no advantages from the latter. Moreover, CSA (Coded Slotted ALOHA)[11] and MuSCA (Multi-Slots Coded ALOHA)[12] focused on packet encoding before or after fragmentation with no packet replications. However, the signaling overhead related to the number of fragments for each packet makes CSA and MuSCA difficult to implement.[13] Later, MARSALA (Multi-replicA decoding using corRelation baSed locALizAtion)[14] was used for solving CRDSA's deadlock situations when there are no more decodable packets. First, it will localize replicas of collided packets on a randomly chosen time slot using correlations between the latter and the remaining time slots on the frame. Let us call this first step the global localization. Then, it will proceed to coherent signal combination between replicas of the same packet for a better chance of decoding thanks to a higher SNIR (Signal-to-Noise plus Interference Ratio). This is valid as long as the number of replicas is equal to two. However, if more than two replicas are used, an association step shall follow the signal combination step. It is meant to associate all the replicas that belong to the same packet together, which requires extra correlations between a combined signal (the signal of the time slot of reference with any of the correlation peak signals from the global localization step) and the remaining correlation peak slots from the global localization step. A new signal combination will then be performed between the already combined signal that gathers two replicas and the newly identified slot that contains another replica of the same packet. This process alternation between the replicas' association and the signal combination is repeated until all the replicas of the same packet are associated and combined together.

Despite the good performance MARSALA offers in terms of PLR and Throughput, compared to CRDSA, it induces an additional processing complexity due the localization process using correlations. Furthermore, with two replicas per packet (which results in less complexity than scenarios with a higher number of replicas) MARSALA witnesses an error floor as in CRDSA. This is due to the loop phenomenon that occurs when two or more packets have their replicas transmitted in the exact same positions. The probability of loop occurrence have been derived for CRDSA in Del Rio Herrero and De Gaudenzi[15] Appendix D. It was shown that the PLR floor for CRDSA, which is observable in low loads, is strongly related to the high probability of the loop phenomenon occurrence. It was also concluded that a higher number of replicas or a higher number of slots per frame decreases the loop phenomenon occurrence probability. The same comments and observations can be made regarding the PLR floor of MARSALA. The difference is that the PLR floor in MARSALA is lower and can be observed with in higher loads than in CRDSA (see Figure 1) due to the extra collision resolution that is performed after CRDSA's deadlock. It was also shown in Zidane et al.[16] that the PLR floor is mitigated when three replicas are used instead of two replicas per packets. Nevertheless, for complexity matter (regarding the number of localization correlations, as explained for MARSALA in the paragraph above), we consider a system with a minimum number of replicas and slots.

In order to mitigate the complexity of replicas localization and loop phenomenon, two methods using a Shared POsition Technique for Interfered random Transmission (SPOTiT) have been introduced. Both rely on making the receiver be aware of the potential positions, on the frame, of all packets of users attached to it. The first technique, Random SPOTiT[17] (noted R-SPOTiT in the rest of the paper), allows the receiver and each



(A) Packet Loss Ratio (B) Throughput (bits/symbol)

FIGURE 1 R-SPOTiT, MARSALA, and CRDSA performance, 100 information bits, QPSK modulation, turbo code of rate 1/3, 100 slots per frame, AWGN channel, and E_S/N_0 = 10 dB [Colour figure can be viewed at wileyonlinelibrary.com] of the transmitters, without any additional signaling information, to have a common knowledge of the time slot positions in order to mitigate the replicas localization complexity. Moreover, pseudo-orthogonal preambles have been used to further reduce the localization complexity in terms of number of correlations. The second technique, Smart SPOTiT,[18] constructs an optimal distribution with packets' positions on the frame where no loops can be created,with minimum complexity and with signaling information that is sent only once to the receiver. R-SPOTiT and MARSALA do not introduce modifications to the receiver, unlike Smart SPOTiT that requires position management and signaling. Therefore, we focus in this paper mainly on defining a whole version of R-SPOTiT, its parameters, and to compare its complexity to that of MARSALA. The complexity assessment that was reported in Zamoum et al.[17] concerns the number of data correlations that are necessary to localize replicas of collided packets on a given reference time slot. It was derived per user and without considering preamble detection. In this paper, we derive the localization complexity over the whole frame, in addition to including preamble detection correlations. Besides, the case of a single preamble for Random SPOTiT, which was not considered in Zamoum et al.,[17] will be described. The complexity of Smart SPOTiT is not in the scope of this paper and is a subject of study in future work.

For paper organization, we introduce the system parameters and legacy RA techniques in Section 2. R-SPOTiT is described in Section 3, complexity parameters in a single preamble or multi-preamble environment are described in Section 4, simulation scenarios with their related complexity are presented in Sections 5 and 6, respectively, and finally, simulation results are analyzed in Section 7 before concluding in Section 8.

2 | SYSTEM PARAMETERS AND LEGACY RA TECHNIQUES

In this section, the system parameters will be presented. These parameters that characterize the structure of the signal to be transmitted, the frame, and the channel are the same that are used for simulations presented in Section 7. The general idea of the decoding algorithm is described afterwards along with the legacy RA techniques CRDSA and MARSALA.

2.1 | System parameters' overview

We focus in this paper on the return link of a satellite communication system (example of Figure 1) where users transmit their information over the same frequency within a frame of N_S time slots, on an AWGN channel. An E_S/N_0 equal to 10 dB is considered in the rest of the paper. The satellite can be a bent pipe that relays information to a gateway, which means that the return link includes an uplink from the user terminals to the satellite and a downlink from the satellite to the terrestrial gateway (Figure 2). Otherwise, if demodulation is implemented on board the satellite (regenerative satellite), only an uplink is required for the return link, from the user terminals to the regenerative satellite.

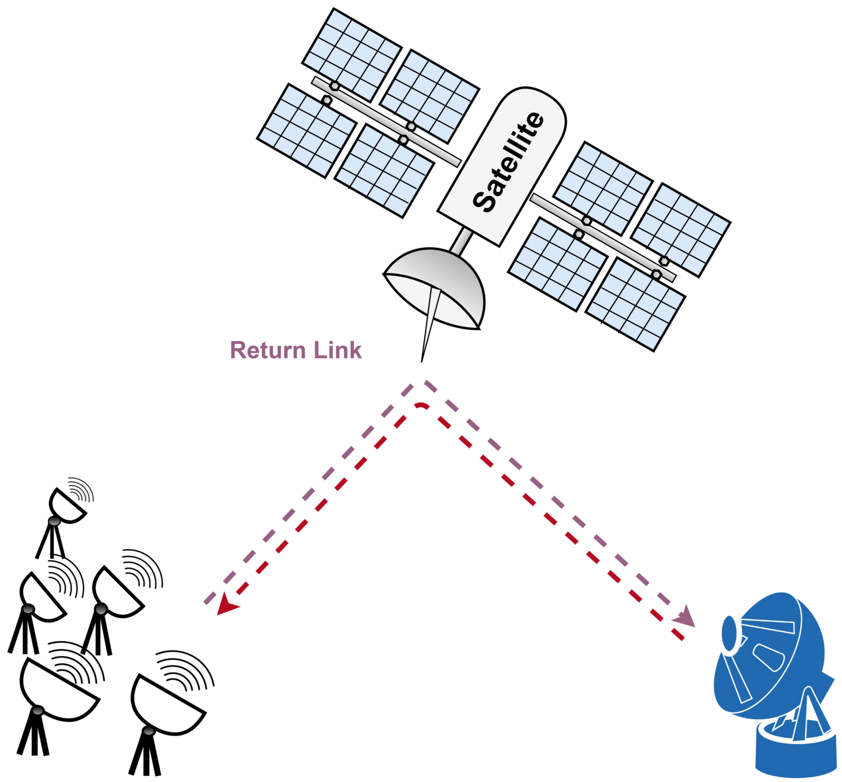


FIGURE 2 Satellite communication system example with a bent pipe satellite [Colour figure can be viewed at wileyonlinelibrary.com]

At each frame, two replicas (N_R = 2) of transmitted packets are sent over different time slots. A packet is composed of a preamble which can be unique or from a selection of N_P Gold pseudo-orthogonal codes,[19] a postamble, randomly distributed pilot fragments for potential synchronization errors, and a payload. The latter is a set of 150 symbols generated after Turbo coding of rate 1/3 and QPSK modulation of a fixed-length binary information of 100 bits. These values have been taken according to the literature.[13,15] In addition, it has been shown in Zidane et al.[20] that 3GPP turbo coding presents better results with MARSALA in terms of throughput in comparison to DVB-RCS2 and CCSDS turbo codes. At reception, multiple iterations are made with CRDSA decoding process that applies SIC. It analyzes the frame, slot by slot, to look for replicas free from collision, decode them, reconstruct the other replica, and then suppress both of them. This process is performed iteratively until no more packets are on the frame or until a deadlock situation is reached. If CRDSA can no more retrieve packets, R-SPOTiT or MARSALA, described next, will take over the decoding process. In other words, we establish a two-step procedure for packets decoding at reception that come from a multi-access transmission channel. The first step being CRDSA until it reaches a deadlock condition, the second one is to unlock the system with one of the methods R-SPOTiT or MARSALA.

2.2 | Legacy RA techniques

In this section, we will describe the CRDSA and MARSALA methods. R-SPOTiT will be introduced in the next section.

2.2.1 | CRDSA

As stated earlier, two or more time slot positions are randomly selected on the frame, by each user to transmit a packet several times. Transmitting multiple replicas of a same packet on the same frame was initially introduced by Diversity Slotted ALOHA (DSA).[21] At reception, a preamble search procedure is first adopted in order to perform the interference cancellation algorithm. If multiple preambles are used, parallel correlations are performed around the search region. The presence of the single or multiple preambles is checked over a region that includes the guard interval around the preamble length at the beginning of each time slot. Initially in CRDSA, multiple preambles were used in order to estimate the channel, independently from the interfering packets.[13] Furthermore, the carrier phase error for a given packet is not the same for all its replicas, as it can vary from one time slot to another. As a consequence, preamble-based phase estimation was necessary. However, it was shown that using a single preamble was sufficient for channel estimation even in high loads, which makes packet detection possible. Small values of frequency, phase, or timing offset appear to sufficiently decorrelate the colliding packets. For packet cancellation, the full packet was reused once detected, in order to enhance the carrier phase estimate. When a preamble is successfully detected, channel parameters are estimated before decoding. If the packet is decoded, SIC operations exploit the signaling information introduced by CRDSA at the payload of each burst to point to the other replicas' locations on the frame. This means that whenever one of the replicas is decoded, information about its burst locations is retrieved, and therefore, all of the replicas can be suppressed from their respective positions. The removal of interference with SIC requires channel estimation and compensation for all replicas on their respective positions to avoid residual errors. Each of the timing offset, frequency shifting, and amplitude can be estimated from the decoded replica. However, carrier phase related to each replica has to be estimated on each position independently from the others. This frame analysis is made in a successive way until no more packets can be retrieved (deadlock situation) or until all packets are decoded.

2.2.2 | MARSALA

It is worth noting that with the proposed MODCOD with low rate FEC, there is a high probability that two colliding packets can be decoded if they are alone on the time slot. However, the deadlock situation (no decodable packets) occurrence of CRDSA increases with the increase of the number of transmitters (high loads). This means that the SNIR is too low to decode, and therefore, the locations of replicas on the frame can no longer be known. We recall that this information is only accessible when the payload is recovered. At this point, MARSALA can intervene to unlock the system and re-trigger CRDSA again.

First, a Reference Time slot (RTS) with collided packets is randomly chosen on the frame; then, correlations are made between the RTS and the remaining time slots on the frame. This step is meant to localize, on the frame, the other replicas of the collided packets on the RTS thanks to the detected correlation peaks. Second, coherent signal combination between localized replicas of the same packet is introduced by MARSALA before decoding. Consequently, a higher power of the packet of interest is observed, which results in a higher Signal-to-Interference Ratio (SIR). This concerns the SIR value of the combined replicas compared to the SIR value of a single replica (which is undecodable using CRDSA alone). Let us take, for instance, a case with no loops, equipowered packets of power P, and consider that a given packet (N_R = 2) is collided with two other packets. In other words, each of the replicas of that packet of interest has two interfering packets. Three interfered packets are thus present on each of the two slots: the replica of the packet of interest plus two interfering packets. This means that the total number of packets that are interfering with the packet of interest, considering both slots, is equal to four. The SIR_r over one of the undecodable replicas of the packet of interest is equal to $SIR_r = P/(P + P) = 1/2$. When CRDSA is accompanied by MARSALA, a combination step between the two slots is performed. In other words, the resulting SIR_u of the combined replicas takes into account all the interfering packets that are collided with both replicas as follows: $SIR_u = 4*P/(P + P + P + P) = 1$. The SIR value of the combined packet during MARSALA is doubled in this case, compared to the SIR value of one of the replicas during CRDSA. If the number of replicas is more than two, an association step is added to the previous localization step in order to associate each N_R localized replicas to a given packet. In order to associate the N_R localized replicas to a given packet, it was first proposed to combine the RTS with the highest correlation peak signal. This corresponds to a second replica of one of the collided packets. Then, extra correlations are performed between the combined signal and the remaining previous peaks' positions that regroup all replicas of collided packets on the RTS. The association step is obviously spared if the number of replicas is equal to two, as any two combinations will associate the two replicas of a given packet. Moreover, a particularly important task for MARSALA to fulfill is the estimation and compensation of the timing offset and phase shift differences between replicas for a maximized coherent combination gain. Proposed solutions and signal processing details are provided in the literature.[16,22] A strategy based on combining the use of the EM (Expectation Maximization) algorithm at the preamble and postamble, along with an initialization of the channel parameters that relies on an auto-correlation operation instead of a random initialization, has shown good results in terms of PER compared to the traditional EM. Moreover, an estimation using pilot symbol assisted modulation has also been proposed for MARSALA in the same research work. When these are combined with a joint estimation and decoding (which should enhance the SIC performance) to estimate the different channel parameters, very low loss of the system performance is observed. These techniques can be applied to R-SPOTiT as it is based on the same combination principle as in MARSALA. Phase noise can also be considered, especially when low cost local oscillators are used. Zidane et al.[23] estimate the impact of phase noise for recent RA techniques like CRDSA and MARSALA and optimize different carrier phase estimation algorithms to cope with this issue. The objective of our contribution is mainly to evaluate the complexity and not to evaluate the performance of RA techniques. Thus, in this paper, we consider perfect channel estimation and compensation.

3 | RANDOM SPOTIT

R-SPOTiT with multiple preambles has been described in previous work.[17] However, a complete definition will be presented here. We will include a single preamble adoption in the presentation. In Section 5, we will consider two scenarios: R-SPOTiT with only one preamble and R-SPOTiT with multiple preambles.

Random SPOTiT reaches the same performance as MARSALA in terms of PLR and throughput (see Figure 2), but with a lower packet localization complexity (the number data correlations meant to localize replicas of collided packets). A single preamble can be used or there can be a set of pseudo-orthogonal codes. In the latter case, the preamble associated with each user will be shared with the receiver as well. The use of multiple preambles in R-SPOTiT is motivated by a potential mitigation of the packet localization complexity with an optimal gain. In other words, having multiple preambles should help reduce the number of data correlations that are meant to localize a packet.

An identical PRNG used at both the transmitter and the receiver (that has identification information as a seed and time slot positions with their associated preamble [if applied] as an output) will allow them to generate the same information. The input seed of the PNRG can be static, using, for instance, only the HID of each user, determined by the receiver thanks to the logon phase; or it can be dynamic if it exploits the frame ID (referring to F_{ID} which is incremental) in addition to the HID. For example, the seed can be computed for a given user u through a simple addition $HID_u + F_{ID}$. In the latter case, the time slot positions of each user varies from one frame to another as the output of the PRNG changes when the seed changes, which avoids a potential continuous loop. In fact, unsolvable loops can be created in some applications where several users generate the same positions, and they transmit successively on the same frames.

On the one hand, using the seed of the PRNG, each terminal will individually select the replicas' positions of its packet and the preamble to use if multiple codes are adopted. On the other hand, the receiver, knowing all users' HIDs and the F_{ID}, will be able to construct an information table that includes all possible positions in a single preamble case thanks to the same seed and PRNG. In a multi-preamble environment, the preamble choice for each user attached to the gateway will be associated to the time slot positions of the packet's replicas on the information table (see Figure 3). It is worth noting that in this slotted environment, the whole system is synchronized, which means that each terminal knows when the next frame should start, thanks to a clock that is regularly transmitted (on the forward link). Hence, all terminals synchronize themselves with the receiver even after a sleep mode or a non-active period.

During the localization process, R-SPOTiT looks for replicas of collided packets on a randomly chosen time slot of reference using data correlation operations. In the case of a single preamble, this is performed over the positions of the second replicas for all potential packets. For example, with 100 time slots per frame and 2000 users with two replicas each, attached to the gateway, there will be an average of $2000*2/100 = 40$ packets per time slot. In this worst case scenario, and with a single preamble, there will be an average of 40 correlations needed using R-SPOTiT to decode a packet, while MARSALA requires $N_S - 1 = 100 - 1 = 99$ correlations. Furthermore, if multiple preambles are used, correlations are made over the second replicas' positions of all potential packets using the same detected preamble on the analyzed time slot of reference.

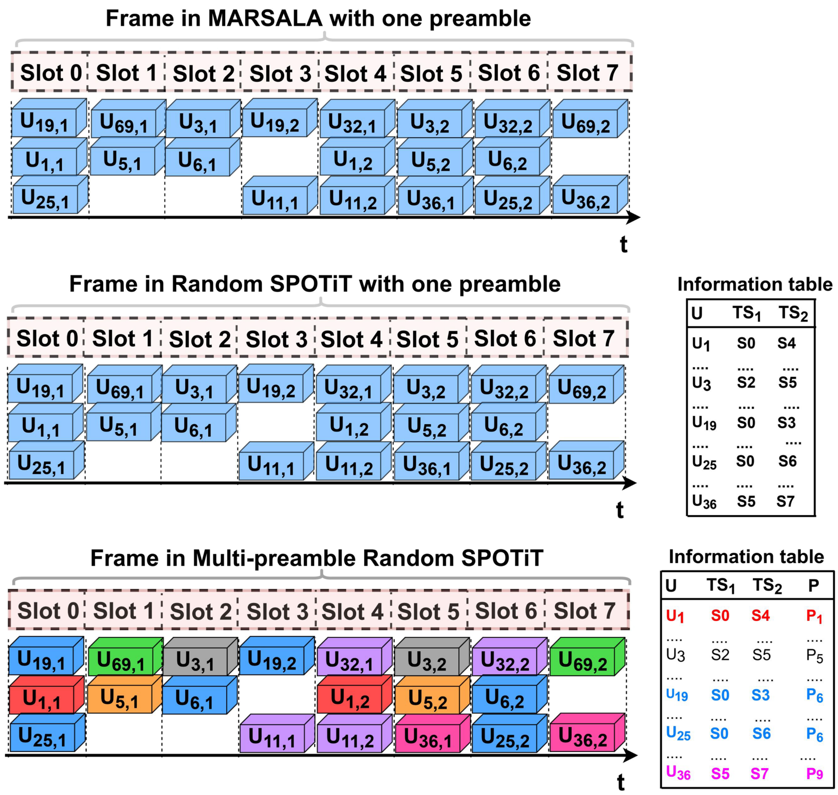


FIGURE 3 Frame structure example at reception. [Colour figure can be viewed at wileyonlinelibrary.com]

Figure 3 displays a frame structure example with the same packet distribution for R-SPOTiT, in a single and multi-preamble case, and for MARSALA. The difference between the two single preamble cases is that R-SPOTiT benefits from supplementary information provided by the information table. The information table as it is illustrated in Figure 3 represents the known and stored information at the receiver, as a result of the PRNG generation using the proper seed to each terminal *U*. Indeed, the receiver is aware of the time slot positions TS_1 and TS_2. For instance, if the randomly selected RTS is the slot 0, R-SPOTiT with a single preamble makes correlations only on slots 3, 4, and 6. These are the locations of the second replicas' positions of users U_1, U_{19}, and U_{25} that are collided on slot 0 (identified in the information table). In the multi-preamble case, each user *U* selects a preamble *P* before transmission. Information about *P* is provided in the information table at reception. This means that, if the blue preamble is detected on slot 0, which is the RTS, only two correlations have to be performed: on slot 3 and on slot 6, or no correlations at all if the red preamble is detected (as it is the unique potential transmitter on slot 0 according the information table). We recall that decoding one packet is sufficient to retrigger CRDSA. At the same time, MARSALA would perform correlations over the $N_S - 1$ time slots: from slot 1 to slot 7.

As long as the number of potential users having the same detected preamble is smaller than $N_S - 1$ (for the worst case of MARSALA with two preambles), Random SPOTiT is less complex. Therefore, there can be a maximum number of active terminals beyond which the complexity between MARSALA and Random SPOTiT remains the same. A solution to minimize the localization correlations complexity in Random SPOTiT is to choose the reference time slot that has the minimum number of potential transmitters over one of the detected preambles. This information can be retrieved from the information table at the reception. The worst case scenario where MARSALA and R-SPOTiT have the same localization complexity can be described as follows: when, with a certain number of active terminals, the minimum of number of potentially collided packets for a given preamble on the reference time slot (optimally chosen) is equal to $N_S - 1$, and all these potential collided packets have their replicas on different time slots. This means that Random SPOTiT should correlate the reference time slot with the $N_S - 1$ different slots. In other words, R-SPOTiT will have exactly the same behavior as MARSALA. The probability of occurrence of such situation (similar complexity between MARSALA and R-SPOTiT) is proportional to the number of preambles and to the number of active users, which in turn depends on the number of logged terminals. This situation might endure for few iterations before R-SPOTiT reaches a smaller complexity than MARSALA due to the interference cancellation. However, the verification of the truthfulness of the latter statement is not in the scope of this paper.

4 | COMPLEXITY PARAMETERS IN A MULTI-PREAMBLE ENVIRONMENT

In our overall localization complexity analysis, we focus on two main parts: firstly, the preamble detection that is necessary to the CRDSA decoding process and, secondly, the replicas localization required in R-SPOTiT or MARSALA before signal combination. We recall that CRDSA is applied first until no more packets can be retrieved, then a complementary treatment of R-SPOTiT or MARSALA is solicited. Once CRDSA is unlocked, it will be triggered again. This process is repeated until no more packets are solved. The whole process ends when one of the three following conditions is fulfilled: (1) when CRDSA alone has decoded all packets on the frame, (2) when CRDSA plus the complementary treatment have decoded all packets, and (3) when they have decoded the maximum number of packets before both of them are blocked due to the high level of collisions at high loads.

4.1 | Preamble detection correlations

Preamble detection is performed only at CRDSA. The packet decoding process at this step is attempted when one or more preambles are correctly detected over the analyzed frame. Consequently, CRDSA is blocked when no more preambles are detected or when packets cannot be retrieved even with detected preambles, due to the high level of collision.

In the preamble detection analysis, we compute the overall number of correlations over a frame until the system reaches a blocking situation. Let us consider Gold code preambles. Parallel correlations are made over the preamble search region to look for correlation peaks. Later, a preamble correlation will be expressed in terms of a data correlation. A preamble detection method is proposed below, which is also valid for the single preamble case.

A transmitted replica r of a user u on a given time slot can incur a phase error $\phi_{u,r} \in [0; 2\pi]$ and can be shifted in time with $\tau_u \in [-2T_S; 2T_S]$ where T_S is the symbol duration. This means that preamble correlation peak search is performed over the duration of four symbols ($4T_S$). Each replica has a Gold code preamble of length 31. It corresponds to the pair of the maximum length pseudo noise sequences (m-sequences) for the shift registers which generate the Gold codes. Thus, we describe the preamble region signal P_T including the guard interval at time instant t as the following two step process:

First, a coarse tracking is made considering a small number of samples per symbol, and then a fine tracking is performed over the strongest symbol with more samples. In other words, the coarse tracking step makes N_C correlations for one preamble detection, and the fine tracking step makes N_F correlations for the same preamble. Hence, a total of $N_{bt} = N_C + N_F$ correlations is considered necessary for every preamble in order to spot the closest location to its position (provided that the package has been transmitted) or the position itself. For example, if the coarse step presents two samples per symbol and the fine step eight samples, N_{bt} will be equal to 16 basic preamble correlations (two correlations over each of the four symbols during the coarse step plus eight correlations over the strongest one during the fine tracking). Figure 4 gives a general overview of how preamble detection is made during CRDSA. The dashed parts show the information that the receiver exploits, such as the information table of R-SPOTiT and the result of packet decoding. An alternative method to the two-step approach would be the use of an interpolation among the correlation points to find the peak. However, this method has not been studied in this paper.

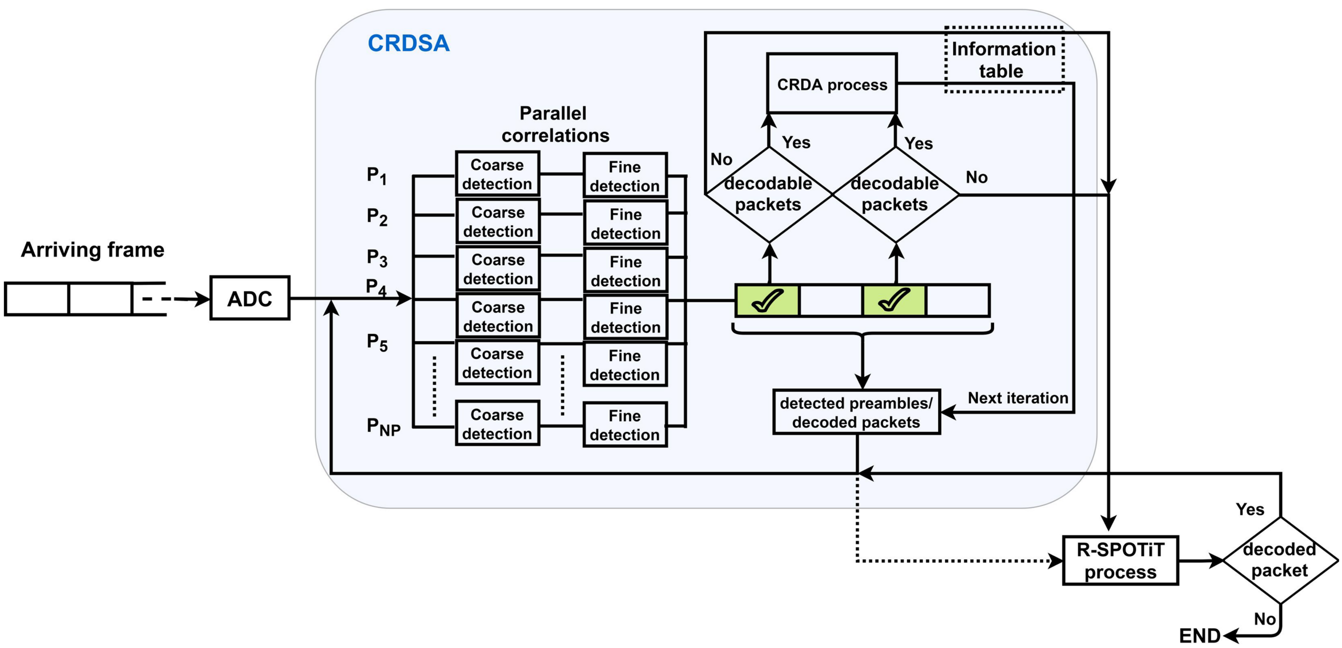


FIGURE 4 General overview of the receiver [Colour figure can be viewed at wileyonlinelibrary.com]

4.2 | Data localization correlations

The data localization correlations are defined according to which complementary treatment is used to unlock CRDSA. As MARSALA does not take into account the preamble results from CRDSA, it will proceed with the same coarse and fine tracking operations (as described in the above paragraph) on the whole packet. This means that N_{bt} data correlations made for each localization operation for each packet. As the same preamble is used for all users, the beginning of a packet is almost impossible to determine in high loads. This is equally the case for R-SPOTiT with only a single preamble. However, R-SPOTiT with multiple preambles, and its information table that considers the preamble detection result from the previous CRDSA iteration, requires only one data correlation for a localization operation. This is due to the fact that the beginning of a packet is already determined thanks to its detected preamble at CRDSA.

Also, the process of randomly choosing a reference time slot in the complementary treatment is performed repetitively until a packet is decoded, which will unlock CRDSA.

When we take a closer look at the two correlations that are considered in our complexity analysis, we can observe that the only difference between them is that the preamble correlation is shorter than the complete data correlation. Indeed, a correlation over a preamble of 31 symbols can be perceived as a fifth correlation over the 150 data symbols.

4.3 | Total number of correlations per frame

To sum up, the total number of correlations C_T per frame includes the preamble detection operations C_P and the data localization operations C_D. C_P is thus the total number of preamble correlations, including the coarse and fine tracking over all CRDSA iterations, both before and after the complementary treatment and until the whole system is blocked. C_D is performed by R-SPOTiT or MARSALA to retrigger CRDSA each time it is blocked until the whole system reaches a deadlock. The total number of correlations over a frame C_T is described below:

$$C_T = \sum_{\delta=1}^{\Delta}\left(\sum_{it=1}^{N_{it}} C_P(\delta, it) + \sum_{\lambda=1}^{\Lambda(\delta)} C_D(\delta, \lambda)\right). \tag{1}$$

This equation expresses the overall frame complexity during the whole CRDSA/R-SPOTiT or CRDSA/MARSALA process. δ is the frame analysis index, and Δ is the maximum value of δ that is reached when the whole system is blocked. Its value can vary from one frame to another. N_{it} is the number of CRDSA iterations. λ is best explained through Λ. $\Lambda(\delta)$ is the total number of reference time slots that can be randomly chosen during a given frame analysis index δ. Thus, λ indexes the choice of a reference time slot, not the reference time slot itself. For example, for $\lambda = 1$, the randomly chosen time slot could be number 20, and for $\lambda = 2$, the time slot could be number 13 and so on. $\Lambda > 1$ is invoked when no packet can be decoded by the complementary treatment for $\lambda = 1$. This means no solution were found for the randomly chosen time slot at $\lambda = 1$.

Let us consider a scenario where all packets are decoded by CRDSA, and no complementary treatment is needed. In this case, the last sum in (1) can be considered 0 ($\Lambda(\delta) = 0$). In this scenario $\Delta = \delta = 1$, and $C_T = \sum_{it=1}^{N_{it}} C_P(\delta, it)$. In a scenario where CRDSA finds itself in a deadlock, a complementary method will attempt to decode one single packet. To do that, $\Lambda(1)$ (δ is still equal to 1) attempts are performed. In MARSALA, $\lambda \in [1; \Lambda(\delta) = N_S]$ where N_s is the number of possible reference time slots (number of slots on the frame). In R-SPOTiT, several operations can be performed on a given reference time slot before passing on to the next ($\lambda = \lambda + 1$). The number of operations depends on the number of preambles in the time slot. When CRDSA is retriggered, the frame is analyzed again and $\delta = 2$. When $\delta = \Delta$ given that $\Delta \geq 1$ (see Table 1 of Section 7), the whole system is solved if all packets are decoded, or blocked if neither CRDSA nor R-SPOTiT or MARSALA can decode new packets. Basically, For each δ, the complexity of the frame analysis consists of preamble detection operations C_P at CRDSA and data localization operations C_D at R-SPOTiT or MARSALA. Also, in the rest of the paper, we consider a basic correlation C_b that corresponds to the data correlation, as a unit for computation of the complexity. Thus, a basic preamble correlation C_{bp} will be expressed in regard to C_b such as $C_{bp} = \frac{C_b}{R}$. R is the ratio between the data and the preamble lengths ($R = 5$ if 150 symbols are considered for the data with a preamble of 31 symbols). A detailed overview of the receiver applying R-SPOTiT with CRDSA is provided in Figure 5. We recall that the dashed parts show the information that the receiver exploits (information table of R-SPOTiT and packet decoding result).

5 | SIMULATION SCENARIOS AND GENERAL ALGORITHMS

In this section will be described each of the scenarios regarding CRDSA complemented by R-SPOTiT or by MARSALA. A single preamble is always considered for MARSALA. The complexity of R-SPOTiT will be assessed for a single preamble case as well as with multiple preambles. The scenarios below describe when the preamble detection correlations and data localization operations take place.

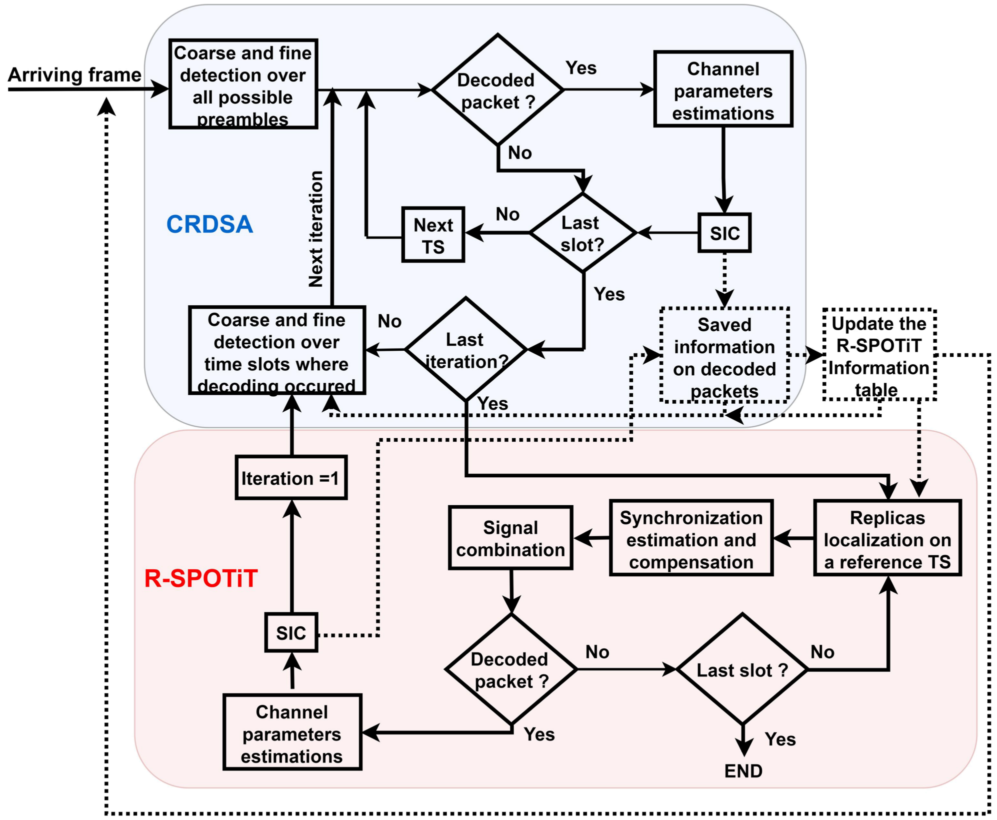


FIGURE 5 Detailed overview of the receiver [Colour figure can be viewed at wileyonlinelibrary.com]

5.1 | MARSALA with one preamble

For the sake of comparison with R-SPOTiT, we consider here a single preamble with the same length as the multiple pseudo-orthogonal preambles in R-SPOTiT. At the first frame analysis index ($\delta = 1$) and first iteration of CRDSA ($it = 1$), a preamble detection operation is performed over each slot of the frame. Thus, the number of preamble correlations is equal to $N_S N_{bt} C_{bp}$. At an nth iteration, preamble detection is only applied to $N_{crdsa}(\delta, n - 1)$ time slots where successful decoding took place at the previous iteration ($n - 1$ with $\delta = 1$). Hence, the number of preamble correlations is equal to $N_{crdsa}(1, n - 1)N_{bt}C_{bp}$.

When the frame analysis index is higher than 1 ($\delta > 1$), it means that MARSALA has been applied. At the first iteration of CRDSA ($C_P(\delta > 1, 1)$), preamble detection is performed over the time slots where successful decoding of MARSALA happened at the previous frame analysis index. As only one packet decoding is required for MARSALA to trigger CRDSA again, only the positions of the N_R replicas of the decoded packet are considered. At an nth iteration, the preamble detection still takes into account CRDSA decoding of the previous iteration.

During MARSALA process, data localization operations are accomplished between a randomly chosen reference position and the rest of the time slots on the frame. If a packet is decoded, CRDSA is retriggered again; otherwise, another reference time slot is chosen, and same operations are performed one more time. This process is repeated $\Lambda(\delta)$ times until a packet is decoded; thus, the number of basic data correlations is equal to $\Lambda(\delta)(N_S - 1)N_{bt}C_b$. The maximum value of $\Lambda(\delta)$ can be reached without being able to decode any packet; in this case, the whole system is blocked.

5.2 | Random SPOTiT with one preamble: scenario 1

In this scheme, a single preamble is used for R-SPOTiT in the same way as in MARSALA. This means that similarly to MARSALA, with $\delta = 1$ and $it = 1$, there are $N_S N_{bt} C_{bp}$ correlations for preamble detection. Furthermore, at an nth iteration, preamble detection is only applied to $N_{crdsa}(1, n - 1)$ time slots where successful decoding took place at the previous iteration; hence, the number of preamble correlations is equal to $N_{crdsa}(1, n - 1)N_{bt}C_{bp}$.

After CRDSA is blocked, R-SPOTiT intervenes ($\delta > 1$). The preamble detection of the first iteration at any index δ is performed over the time slots where successful decoding of one packet by R-SPOTiT happened (at $\delta - 1$). At an nth iteration, the positions of the packets decoded by CRDSA at the previous iteration are analyzed for preamble detection.

During R-SPOTiT processing, data localization operations are held between a reference time slot and the second replicas' positions of all potentially collided packets on the RTS. The RTS is derived from the R-SPOTiT information table that is updated after each decoding. This means that when replicas of a packet are decoded on their positions, they will be suppressed from the potential packets in collision on their respective time slots. Similarly to MARSALA, if a packet is decoded, CRDSA will be launched again; if not, another reference time slot is chosen ($\Lambda(\delta) > 1$), and same operations are performed.

5.3 | Random SPOTiT with multiple preambles: scenario 2

One of the characteristics of R-SPOTiT is the use of pseudo-orthogonal preambles to reduce data correlations compared to MARSALA. Nevertheless, if MARSALA is operational with one preamble, the CRDSA part of the algorithm will be more complex with R-SPOTiT than with MARSALA. This is due to the parallel preamble detection operations. It then becomes important to assess the overall complexity of R-SPOTiT including preamble detection along with data localization and compare it with a single preamble case of MARSALA.

In this scenario, $N_P \in [2; N_{PT}]$ pseudo-orthogonal Gold codes are considered, and N_{PT} is the total number of pseudo-orthogonal Gold preambles with a given length. At the frame analysis index $\delta = 1$ and first iteration of CRDSA $it = 1$, $N_{pp}(s)$ parallel preamble detection operations are performed on each time slot s, where $N_{pp}(s) \in [1; N_P]$. The value of $N_{pp}(s)$ is derived from the R-SPOTiT information table; it corresponds to the number of potential preambles that could be transmitted on a given slot s among the N_P possible preambles. Therefore, the number of preamble correlations is equal to $\sum_{s=1}^{N_S} N_{pp}(s) N_{bt} C_{bp}$. After each decoding, the information table of R-SPOTiT is updated. Thus, at an nth iteration, the preamble detection process will take into consideration the previous decoding result ($N_{crdsa}(\delta, n-1)$, as noted before) along with the updated information table $\left(\sum_{s=i_1}^{i_{N_{crdsa}}(\delta,n-1)} N_{pp}^{u}(s) N_{bt} C_{bp}\right)$. Here, $N_{pp}^{u}(s)$ is simply the updated $N_{pp}(s)$ after decoding occurred on slot s ($N_{pp}^{u}(s) < N_{pp}(s)$). $i_{N_{crdsa}}(\delta, n-1)$ is the time slot index of the last replica suppressed by CRDSA at the previous iteration. When R-SPOTiT has been applied, with a frame analysis index exceeding one ($\delta > 1$), the preamble detection process at the first iteration of CRDSA should consider the time slot positions of one packet (N_R) where decoding happened with R-SPOTiT $\left(\sum_{s=i_1}^{i_{N_R}(\delta-1)} N_{pp}^{u}(s) N_{bt} C_{bp}\right)$. $i_{N_R}(\delta-1)$ is the time slot index of the last replica belonging to the packet decoded by R-SPOTiT, which unlocked CRDSA. At the next iterations of CRDSA (any n), the previous ($n-1$) one is always taken into account along with the updated information table.

Considering only the packet localization complexity, three cases can be described during the R-SPOTiT process. First, if both preambles of a given packet are correctly detected at the last CRDSA iteration, the packet will be considered localized, and thus, no data localization operations are needed. Second, when one of the two preambles is detected on a reference time slot, data localization operations are performed over the second replicas' positions of potentially collided packets having the same detected preamble. Considering that the replicas of a given user are synchronized at the frame level, a localization correlation is made only once due to the fact that we know the packet time shift after its preamble is detected.

Third, when there are no more detected preambles, the single preamble case can be copied. Indeed, preambles can be ignored; thus, the information table will be exploited only regarding the packets' positions. A localization correlation in this case is performed N_{bt} times as no information about the packet's time shift is available. This way, R-SPOTiT with any number of preambles will have exactly the same performance as MARSALA.

6 | SCENARIO-BASED COMPLEXITY CALCULATION

What has been expressed in (1) is the overall frame complexity during the whole CRDSA/R-SPOTiT or CRDSA/MARSALA processes. This notation is adopted in this section to differentiate the two main terms of correlations, that is, preamble correlations at CRDSA and data correlations during the complementary treatment. Each term will be detailed here according to the scenarios explained in Section 5. Two main cases are constructed depending on whether $\delta = 1$ or when $\delta > 1$. Moreover, each case will have two subcases for CRDSA: when $it = 1$ and when $it > 1$ for the preamble detection.

On the one hand, in CRDSA/MARSALA, C_P varies from one iteration n to another, with respect to δ and according to $N_{scen}(\delta,n)$, which is the total number of operations performed. $N_{scen}(\delta,n)$ is equal to N_S, $N_{crdsa}(\delta, n-1)$, or N_R, which are, respectively, the number of slots, the number of slots where packets have been decoded at the previous iteration with CRDSA, and the number of slots where MARSALA decoded a packet at the previous frame analysis index.

On the other hand, in CRDSA/R-SPOTiT, C_P also varies, with respect to $N_{scen}(\delta,n)$. $N_{scen}(\delta,n)$ is equal in this case to N_S, $N_{crdsa}(\delta, n-1)$, N_R or to $\sum_s N_{pp}^{u}(s)$. The first of the two latter terms concerns R-SPOTiT with one preamble and the last concerns R-SPOTiT with N_P preambles. N_R stands here for the time slots where packets have been decoded by R-SPOTiT with one preamble at the previous frame analysis index; $\sum_s N_{pp}^{u}(s)$ is the number of possible preambles, considering the updated information table, on the time slots where packets have been decoded at the previous iteration of CRDSA or at the last frame analysis index. We recall that when a packet is decoded at a previous iteration or a past frame analysis index, it will be suppressed from the potential packet candidates in the information table. The latter is also being exploited at each analyzed time slot to avoid having parallel preamble detection operations over the whole number of used preambles N_P. In (2), (3), (4), and (5), C_P is conventionally expressed, in terms of basic data correlations C_b, according to each case δ and *it* described at the previous section and for each of the scenarios: MARSALA, R-SPOTiT with one preamble (R-SPOTiT(1)), and R-SPOTiT with N_P preambles (R-SPOTiT(N_P)).

- C_P when $\delta = 1$ and $it = 1$

$$C_P(1,1) = N_{scen}(1,1) N_{bt} C_{bp} = \begin{cases} N_S N_{bt} \dfrac{C_b}{R} & \text{if MARSALA} \\ N_S N_{bt} \dfrac{C_b}{R} & \text{if R-SPOTiT}(1) \\ \sum_{s=1}^{N_S} N_{pp}(s) N_{bt} \dfrac{C_b}{R} & \text{if R-SPOTiT}(N_P) \end{cases} \tag{2}$$

- C_P when $\delta = 1$ and $it = n$

$$C_P(1,n) = N_{scen}(1,n) N_{bt} C_{bp} = \begin{cases} N_{crdsa}(1,n-1) N_{bt} \dfrac{C_b}{R} & \text{if MARSALA} \\ N_{crdsa}(1,n-1) N_{bt} \dfrac{C_b}{R} & \text{if R-SPOTiT}(1) \\ \sum_{s=i_1}^{i_{N_{crdsa}}(1,n-1)} N^u_{pp}(s) N_{bt} \dfrac{C_b}{R} & \text{if R-SPOTiT}(N_P) \end{cases} \tag{3}$$

- C_P when $\delta > 1$ and $it = 1$

$$C_P(\delta,1) = N_{scen}(\delta-1,1) N_{bt} C_{bp} = \begin{cases} N_R N_{bt} \dfrac{C_b}{R} & \text{if MARSALA} \\ N_R N_{bt} \dfrac{C_b}{R} & \text{if R-SPOTiT}(1) \\ \sum_{s=i_1}^{i_{N_R}(\delta-1)} N^u_{pp}(s) N_{bt} \dfrac{C_b}{R} & \text{if R-SPOTiT}(N_P) \end{cases} \tag{4}$$

- C_P when $\delta > 1$ and $it = n$

$$C_P(\delta,n) = N_{scen}(\delta,n) N_{bt} C_{bp} = \begin{cases} N_{crdsa}(\delta,n-1) N_{bt} \dfrac{C_b}{R} & \text{if MARSALA} \\ N_{crdsa}(\delta,n-1) N_{bt} \dfrac{C_b}{R} & \text{if R-SPOTiT}(1) \\ \sum_{s=i_1}^{i_{N_{crdsa}}(\delta,n-1)} N^u_{pp}(s) N_{bt} \dfrac{C_b}{R} & \text{if R-SPOTiT}(N_P) \end{cases} \tag{5}$$

The number of data correlations C_D is expressed in (6):

- C_D for any value of δ and λ

$$C_D(k,\lambda) = \begin{cases} (N_S - 1) N_{bt} C_b & \text{if MARSALA} \\ N^u_{po}(k,\lambda) N_{bt} C_b & \text{if R-SPOTiT}(1) \\ \left(N^u_{pm}(\delta,\lambda) C_b\right)^{(1-\rho)} \left(N^u_{po}(k,\lambda) N_{bt} C_b\right)^{\rho} & \text{if R-SPOTiT}(N_P) \end{cases} \tag{6}$$

At each frame index δ, if necessary, C_D is performed $\Lambda(\delta)$ times until a packet is successfully decoded. The value of C_D is fixed for MARSALA. However, in R-SPOTiT, C_D can be decreased in the same way as for C_P using the information table. In case of a single preamble, $N^u_{po}(\delta,\lambda)$ represents the number of potential packets in collision on the chosen reference time slot with index λ after the last update of the information table for a given frame analysis index δ. When multiple preambles are used in SPOTiT, C_D is characterized by $N^u_{pm}(\delta,\lambda)$ that is the number of potentially collided packets using the same detected preamble over the reference time slot, derived from the updated information table. This means that at least one preamble is detected on the analyzed slot; therefore, we set the boolean variable $\rho = 0$ in (6). When in contrary, there are no detected preambles at any reference time slot ($\rho = 1$), $N^u_{po}(\delta,\lambda)$ is equal to (similarly to the single preamble case) the overall number of potentially collided packets on the chosen reference time slot regardless of their preambles, taking into consideration the updated information table.

The formulas derived in this section rely on random variables which are analytically difficult to estimate such as C_P and C_D. As such, in the next section, we use simulations, in order to put numerical values on these random variables and hence compare the different methods: MARSALA and R-SPOTiT with a single preamble or with multiple preambles.

7 | SIMULATION RESULTS

The simulations conducted in this work apply to the scenarios described above. For each scenario, the overall frame complexity including preamble detection and packet localization is assessed with respect to the number of users that is converted to a channel load in bits/symbol. In scenario 2, the length N_{PT} of the preambles is 31, which are generated using the preferred polynomial pair $\{x^5+x^2+1,\ x^5+x^4+x^3+x^2+1\}$, and N_P varies in {2, 3, 5, 15, 31}. The average values that are derived by simulation in this section allow us to have a computational estimation of the random variables C_P (the total number of preamble correlations on a frame) and C_D (the total number of data correlations on a frame) that constitute the total number of correlations C_T on a frame.

We observe on Figure 6 that in the case of a single preamble, MARSALA's complexity exceeds R-SPOTiT, starting approximately from a channel load of 1.1 bits/symbol. The difference between the two increases with the number of transmitters. MARSALA's complexity is on average four times higher than R-SPOTiT's. At low loads, the complexity is negligible because no complementary treatment is necessary. In this case, no heavy data localization operations are performed, only the single preamble detection operation. However, when multiple preambles are used in R-SPOTiT, low loads experience more correlations than in a single preamble case whereas high loads exhibit up to a certain point (depending on the number of preambles) more complexity than the single preamble case. Then it becomes lower. With 31 and 15 preambles, the number of basic correlations is higher than MARSALA until 1.2 bits/symbol and 1.3 bits/symbol, respectively, then the number of basic correlations evolves gradually, but in a less significant way compared to MARSALA, until 1.7 bits/symbol when the throughput collapses. At this point, the whole system ends earlier when no more packets can be retrieved. Therefore, the number of preamble detection operations decreases. Each of the complexity curves of R-SPOTiT with 15 and 31 preambles crosses the single preamble case at 1.6 bits/symbol and 1.8 bits/symbol, respectively, to become less complex. However, this region is around the throughput collapse point (1.7 bits/symbol).

When N_P is lower, 2, 3, and 5, the complexity at low loads is smaller and closer to the single preamble case than to R-SPOTiT with 31 and 15 preambles. They cross MARSALA's curve at around 1 bit/symbol and evolve differently, in a less significant way. R-SPOTiT with two preambles presents the smallest complexity. These multiple preamble R-SPOTiT curves also cross that of the single preamble at around 1.2 bits/symbol and present a smaller number of basic correlations compared to the single preamble case and for any number of preambles.

To understand better the contribution of C_P and C_D, in each case, to the overall localization complexity at different channel loads, separate metrics are presented. Figures 7 and 8 show, one at a time, C_P and C_D for MARSALA and R-SPOTiT with a single preamble. The number of correlations in the preamble detection is negligible next to the number of correlations for the data localization. Therefore, the overall complexity curve of MARSALA follows the shape of C_D curve. Same comment can be dedicated to R-SPOTiT with one preamble, except that it exhibits considerably lower C_D than MARSALA. However, in R-SPOTiT with 31 and 15 preambles (Figures 9 and 10), C_P is high enough to make the overall complexity curve follow its shape. When the number of preambles is equal to 5, to 3 or to 2 (Figures 9 and 10), the overall complexity shape converges towards C_D because C_P is low.

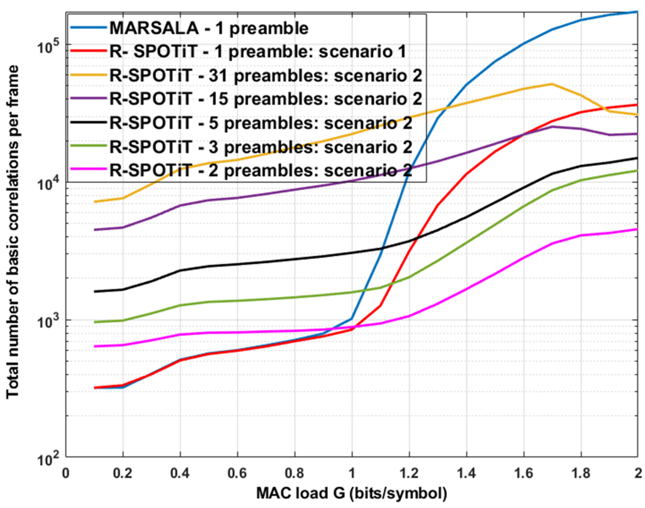


FIGURE 6 Overall frame complexity, with N_{bt} = 16 and R = 5 in CRDSA/MARSALA and CRDSA/R-SPOTiT environments [Colour figure can be viewed at wileyonlinelibrary.com]

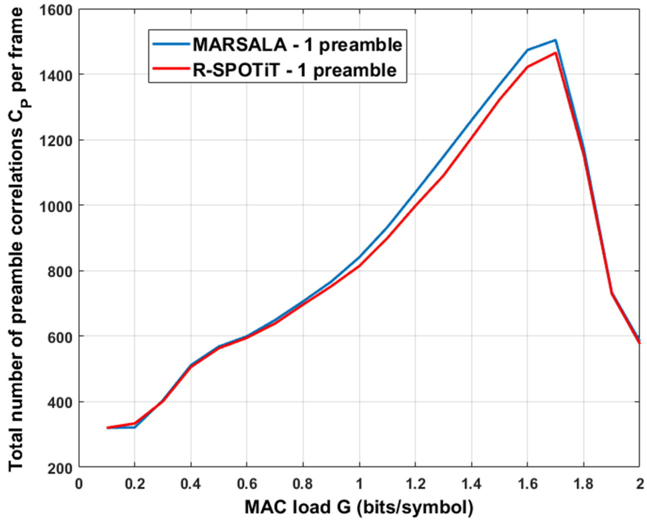


FIGURE 7 Preamble localization complexity in case of MARSALA and single preamble R-SPOTiT [Colour figure can be viewed at wileyonlinelibrary.com]

Consequently, according to the average operational load of a given application, one can select the most adequate complementary treatment with a suitable number of preambles, the smallest in this case, that would take over the decoding process after a deadlock. R-SPOTiT with two preambles is, as stated earlier, the best candidate.

Indeed, the Gold code preamble detection appears to be better when the number of preambles is small. We have computed the total number of times Ψ that R-SPOTiT is used per frame, and how it is distributed according to three cases. Case *A* is characterized with α that is the number of times that a preamble of a given packet is detected on the reference time slot and that its second replica's position exhibits a correlation peak for the same preamble. Case *B* is characterized by β that is the number of times that only one of the two preambles is detected. Finally, case *C* defines γ that is the number of times that no preambles are detected on the whole frame. We recall that during case *A*, the packet is considered localized, and thus, no data correlations are performed. However, during case *B* when $\rho = 0$ (when at least one preamble is detected on an analyzed slot), correlations are made over all second replicas' positions of potentially collided packets on the reference time slot that use the same detected preamble. Thus, the total number of times Ψ that R-SPOTiT is used per frame is defined as follows: $\Psi = \alpha + \beta + \gamma$.

Furthermore, during case *C* when $\rho = 1$ (when there are no detected preambles in any RTS), correlations are made over all second replicas' positions of all potentially collided packets on the reference time slot regardless of their preambles. Figure 11 shows the mean usage percentage of R-SPOTiT per frame during each of the cases *A*, *B*, and *C*. We can deduce that the probability of having both preambles detected on their respective time slots is higher when the number of preambles is low (see the 2-preamble curve). The probability that none of the preambles are detected is hence the lowest when the number of preambles is the smallest. We recall that in this case ($\rho = 1$), a localization correlation is applied N_{bt} times because the beginning of a packet is unknown. For this reason, the total number of data correlations C_D per frame is higher when the number of preambles is high. This is mainly due to the pseudo-orthogonality of preambles. Since preambles are not perfectly orthogonal, a linear

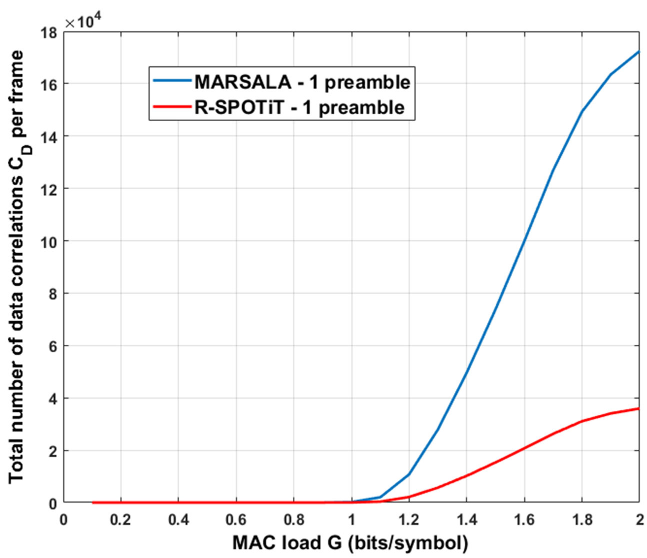


FIGURE 8 Data detection complexity in case of MARSALA and single preamble R-SPOTiT [Colour figure can be viewed at wileyonlinelibrary.com]

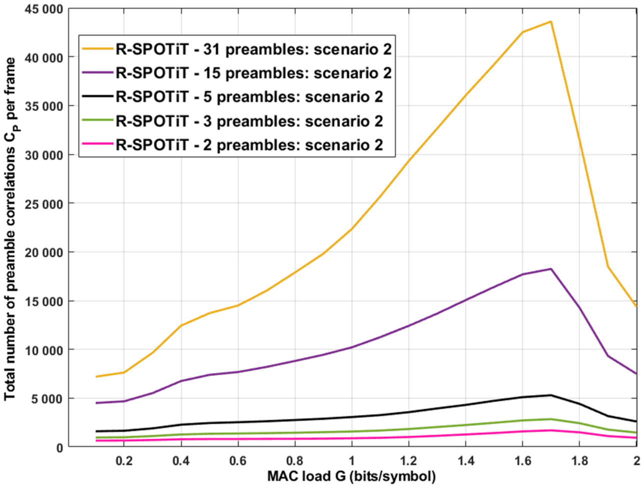


FIGURE 9 Preamble localization complexity in of case multi-preamble R-SPOTiT [Colour figure can be viewed at wileyonlinelibrary.com]

combination of some of them can lead to indistinguishable correlation peaks. Future work will investigate the use of ZCZ (zero correlation zone) sequences[24] instead of Gold sequences. These sequences are promising candidates as temporal uncertainty is low and a very small number of sequences is required.

Furthermore, the complexity introduced in low loads is not very significant in multi-preamble R-SPOTiT compared to the complexity of MARSALA in high loads. Therefore, it is preferable to use multi-preamble R-SPOTiT. Figure 12 sums up the total number of basic correlations, with the various methods and different numbers of preambles for an average load of 1.6 bits/symbol. We can see that the less complex system to use, in this case of pseudo-orthogonal Gold codes, is the multi-preamble R-SPOTiT with 2 preambles.

Moreover, in order to have an idea on how many times the complementary treatment is solicited by CRDSA before the whole system finds itself in a deadlock, we derived experimentally, average values of Δ, which is the maximum frame analysis index δ, in terms of the channel load. These values of Δ define, as well, the average number of packets decoded by R-SPOTiT or MARSALA at each load as only one packet is decoded at each intervention, thus at each δ. Table 1 exhibits the different values of Δ with respect to the channel load and the corresponding number of transmitters N_U. In addition, each value of N_U is associated to the average number of decoded packets by CRDSA when it is used alone, noted D_{crdsa}, and by the complementary treatment D_{CT}, each derived from its corresponding PLR. The complementary treatment here refers to R-SPOTiT or MARSALA that have approximately the same PLR performance. The decoding gain resulting from the complementary treatment compared to CRDSA (Δ) is also derived in percentage (D_{gain}).

Indeed, we can observe that the number of times (Δ) the complementary treatment intervenes to solve CRDSA's deadlock becomes significant in high channel loads. $\Delta = 1$ means that, on average, R-SPOTiT or MARSALA had not taken part of the decoding process, or that their contribution was minimal. Then, until 1.7 bits/symbol, Δ values increase up to 56, then they decrease gradually to reach 14 at a load of 2 bits/symbol along with the throughput collapse. Also, it is noticeable that above the PLR target set to 10^{-3}, the number of decoded packets when CRDSA is used alone (D_{crdsa}) decreases gradually with the increase of the number of transmitters, compared to the total number of users. The same comment can be made regarding D_{CT}, except that with the latter, the decrease of the number of decoded packets compared to the ones transmitted

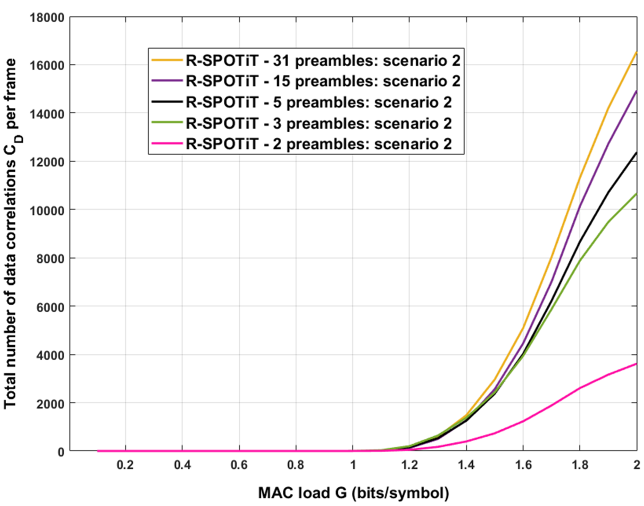


FIGURE 10 Data detection complexity in case of multi-preamble R-SPOTiT [Colour figure can be viewed at wileyonlinelibrary.com]

FIGURE 11 R-SPOTiT usage distribution, with N_{bt} = 16 and R = 5 per frame in percentage [Colour figure can be viewed at wileyonlinelibrary.com]

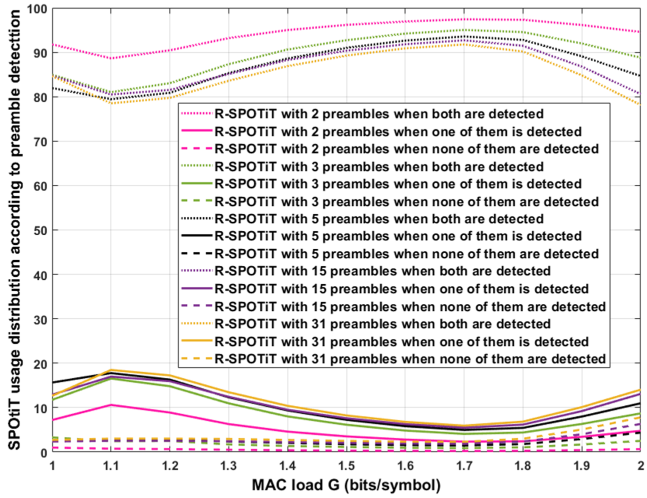
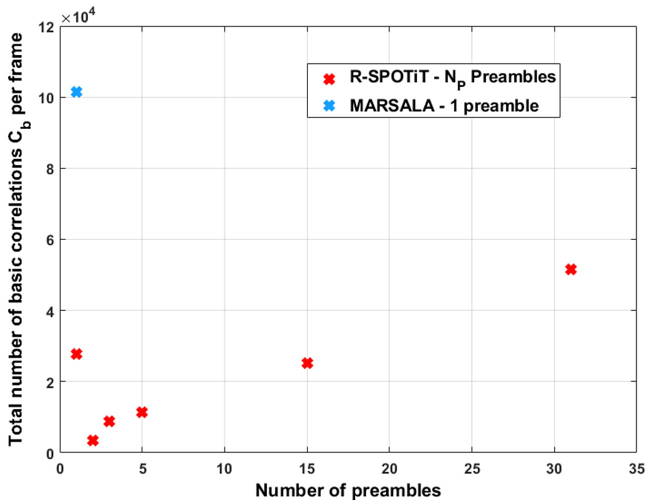


TABLE 1 Average simulation values of the maximum frame analysis index

G (b/s)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
N_U	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255	270	285	300
D_{crdsa}	15	30	45	60	75	90	105	120	134	143	118	77	57	44	30	31	26	17	24	14
D_{CT}	15	30	45	60	75	90	105	120	135	150	165	180	195	210	224	239	239	173	81	50
Δ	1	1	1	1	1	1	1	1	1	1	2	6	13	23	35	47	56	43	23	14
D_{gain}	-	-	-	-	-	-	-	-	0.7%	4.7%	28.5%	61%	70.8%	79%	86%	86.6%	83.5%	57.7%	20%	12%

FIGURE 12 Overall frame complexity, with N_{bt} = 16 and R = 5 in CRDSA/MARSALA and CRDSA/R-SPOTiT with different numbers of preambles, for a channel load of 1.6 bits/symbol [Colour figure can be viewed at wileyonlinelibrary.com]



starts at a channel load of 1.7 bits/symbol, unlike CRDSA that experiences it at a load of 0.9 bits/symbol. Nevertheless, having the complementary treatment that could be R-SPOTiT or MARSALA triggered each time CRDSA incurs a deadlock offers a decoding gain that can reach 86.6% with 47 interventions at a channel load of 1.6 bits/symbol.

8 | CONCLUSION AND FUTURE WORK

The main issue addressed in this paper is the packet localization complexity, over a whole frame, induced by the detection correlations in CRDSA/MARSALA and CRDSA/R-SPOTiT environments. This includes the impact of preamble detection performed during CRDSA in addition to the data localization operations of R-SPOTiT and MARSALA. For an effective detection and less complexity, we introduced a coarse and a fine tracking with different numbers of samples per symbol. In a single preamble scenario, We showed that CRDSA/MARSALA and CRDSA/R-SPOTiT are equivalent in low channel load environments because only preamble detection is performed. This means that only CRDSA is needed for decoding. However in high loads, MARSALA's complexity considerably surpasses R-SPOTiT with a single preamble and is on average four times higher. This means that, compared to MARSALA, R-SPOTiT with one preamble is preferable to use. Nevertheless, when the number of preambles in R-SPOTiT is higher, the system is more complex in low channel loads because of preamble correlations. In high loads, opposite phenomenon is observed. The multi-preamble R-SPOTiT becomes less complex than the single preamble case. This is due to the simplified data localization operations of R-SPOTiT that exploits the pseudo-orthogonal preambles. Also, Gold code preambles turned out to be more effective regarding their detection when their number is small. As such, R-SPOTiT with two preambles presents the smallest complexity in this case. Furthermore, the difference of complexity in low loads between the single preamble R-SPOTiT and the multi-preamble case is less significant than the difference between them in high loads. This becomes even more evident when the number of preambles is small. To summarize, the single preamble case of R-SPOTiT is preferable in low loads, whereas the multipreamble scenario is more suitable in high channel loads. In the latter case, the least complex system is with two preambles. Also, the total number of basic correlations per frame of the single preamble case is smaller than the one of the multi-preamble case. However, the difference between them is less considerable in high loads where the single preamble case complexity surpasses the one of the multi-preamble case. Therefore, we believe that using R-SPOTiT with two preambles is less costly to the receiver.

In this paper, we have evaluated and compared the complexity of two Random Access methods, R-SPOTiT and MARSALA, that have the same system performance in terms of PLR and throughput. The results are quite encouraging; however, some areas are still open to investigation and improvements in future work. Firstly, we would like to assess the impact of real channel conditions, phase noise, and meteorological events on the overall system performance. Secondly, we would like to investigate the use a more robust type of preambles, such as ZCZ (zero correlation zone) sequences, in order to cope with timing errors. Thirdly, we would like to address the hardware implementation complexity resulting from the use of multiple preambles. This will enable us to find the optimal trade-off between the complexity related to the burst demodulator and the overall localization complexity of the R-SPOTiT algorithm, knowing that the optimal number of preambles is two.

Smart SPOTiT, based on a power of two number of slots, that offers better PLR than R-SPOTiT and MARSALA techniques will be assessed as well in future work. An optimal distribution with any number of slots and preambles should then be derived.

ORCID

Selma Zamoum https://orcid.org/0000-0003-4070-8807

REFERENCES

1. Kissling C. Performance enhancements for asynchronous random access protocols over satellite. *IEEE Int Conf Commun (ICC)*. 2011.
2. Clazzer F, Kissling C. Enhanced contention resolution ALOHA-ECRA. In: SCC 2013; 9th International ITG Conference on Systems, Communication and Coding; 2013.
3. De Gaudenzi R, Del Ro Herrero O, Acar G, Garrido Barrabs E. Asynchronous contention resolution diversity ALOHA: Making CRDSA truly asynchronous. *IEEE Trans Wirel Commun*. 2014;13(11):6193-6206.
4. Mengali A, De Gaudenzi R, Arapoglou P. Enhancing the physical layer of contention resolution diversity slotted ALOHA. *IEEE Trans Commun*. 2017;65(10):4295-4308.
5. Abramson N. Multiple access in wireless digital networks. *Proc IEEE*. 1994;82(9):1360-1370.
6. Del Rio Herrero O, De Gaudenzi R. High efficiency satellite multiple access scheme for machine-to-machine communications. *IEEE Trans Aerospace Elect Syst*. 2012;48(4):2961-2989.
7. Abramson N. The ALOHA system—another alternative for computer communications. *Fall Joint Comput Conf*. 1970;37:281-285.
8. Casini E, De Gaudenzi R, Del Rio Herrero O. Contention resolution diversity slotted ALOHA (CRDSA): an enhanced random access scheme for satellite access packet networks. *IEEE Transactions on Wireless Communications*. 2007;6(4):1408-1419.
9. Liva G. Graph-based analysis and optimization of contention resolution diversity slotted ALOHA. *IEEE Trans Commun*. 2011;59(2):477-487.
10. ETSIEN 301 542-2 v1.1.1. Digital Video Broadcasting (DVB); Second Generation DVB Interactive Satellite System (DVB-RCS2); Part 2: Lower Layers for Satellite Standard; 2012.
11. Paolini E, Liva G, Chiani M. High throughput random access via codes on graphs: coded slotted ALOHA. *IEEE Int Conf Commun (ICC)*. 2011.
12. Bui H, Lacan J, Boucheret M-L. An enhanced multiple random access scheme for satellite communications. *Wirel Telecommun Symposium 2012*. 2012.
13. De Gaudenzi Riccardo, Del Rio Herrero Oscar, Gallinaro Gennaro, Cioni Stefano, Arapoglou P-D. Random access schemes for satellite networks: from VSAT to M2M—a survey. *Int J Satellite Commun Netw*. 2016;36.
14. Bui H, Zidane K, Lacan J, Boucheret M-L. A multi-replica decoding technique for contention resolution diversity slotted ALOHA. In: 2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall); 2015.
15. Del Rio Herrero O, De Gaudenzi R. Generalized analytical framework for the performance assessment of slotted random access protocols. *IEEE Trans Wirel Commun*. 2014;13(2):809-821.
16. Zidane K, Lacan J, Gineste M, Bes C, Deramecourt A, Dervin M. Estimation of timing offsets and phase shifts between packet replicas in MARSALA random access,; 2016.
17. Zamoum S, Lacan J, Boucheret M-L, Dupe J-B, Gineste M. Shared position technique for interfered random transmissions in satellite communications,; 2018.
18. Zamoum S, Lacan J, Boucheret M-L, Gineste M, Dupe J-B. Deterministic distribution of replicas positions for multiuser random transmissions in SatComs. *IEEE Global Commun Conf (GLOBECOM)*. 2018.
19. Gold R. Optimal binary sequences for spread spectrum multiplexing (Corresp.) *IEEE Trans Infor Theory*. 1967;13(4):619-621.

20. Zidane K, Lacan J, Gineste M, Bes C, Bui C. Enhancement of MARSALA random access with coding schemes, power distributions and maximum ratio combining,; 2016:1-8.
21. Choudhury G, Rappaport S. Diversity ALOHA—a random access scheme for satellite communications. *IEEE Trans Commun*. 1983;31(3):450-457.
22. Zidane K, Lacan J, Boucheret M-L, et al. Effect of residual channel estimation errors in random access methods for satellite communications. In: 2015 IEEE 81st Vehicular Technology Conference (VTC Spring); 2015.
23. Zidane Karine, De Gaudenzi Riccardo, Alagha Nader, Cioni S. Phase noise impact on the performance of contention resolution slotted random access schemes. *Int J Satellite Commun Netw*; n/a(n/a). https://doi.org/10.1002/sat.1325
24. Fan PZ, Suehiro N, Kuroyanagi N, Deng XM. Class of binary sequences with zero correlation zone. *Electronics Letters*. 1999;35(10):777-779.

AUTHOR BIOGRAPHIES

Selma Zamoum received a PhD in satellite communications from the university of Toulouse (Toulouse INP: Institut National Polytechnique) in 2019. Her PhD research at TéSA laboratory was about synchronous and asynchronous random access techniques for satellite communications. The research project was cofunded by the Centre National d'Etudes Spatiales (CNES) and Thales Alenia Space. In January 2020, she joined the Space Situational Awareness (SSA) company SpaceAble. Her main research interests concern satellite multiple access techniques, and securing space rendezvous and formation flying for On-Orbit servicing.

Jérôme Lacan received his ME and PhD degrees in Computer Science from University Paul Sabatier, Toulouse, France, in1994 and 1997. In 1996 to 2000, he was assistant professor in University of Franche—Comté, France in Computer Science. In 2001, he joined the department “Mathematics, Computer Science, Automatics” of the “Institut Supérieur de l'Aéronautique et de l'espace” (ISAE) as assistant professor (2001-2008) then professor (since 2008). He leads currently the research group RESCOM (communication networks) of the department DISC (Applied Math, Computer Science, Embedded Systems, Networks, System Engineering, Data Science). He is also a member of the laboratory TéSA. He has cosupervised 20 PhD students. He is the author of eight international patents and has published around 100 papers in international journals and conferences. He was also involved in the standardization of various coding schemes in IETF (3RFC), DVB, and 3GPP. He is the cofounder of the startup SPEERYT. His main research interests are coding theory and network protocols applied in the areas of satellite communications and networks. He is currently interested in constructions and applications of erasure codes, cross-layer coding, network coding, and blockchains.

Marie-Laure Boucheret received her degree in Electrical Engineering from ENST Bretagne, Brest, France in June 1985. In June 1997, she received the PhD degree in Communications from TELECOM Paris Tech and the ‘Habilitation à diriger les recherches’ in June 1999 from INPT University of Toulouse. From 1985 to 1986, she had been a research engineer at the French Philips Research Laboratory (LEP). From 1986 to 1991, she had been an engineer at Thales Alenia Space, first as a project Engineer (TELECOM II program) then as a study engineer at the transmission laboratory. From 1991 to 2005, she was an Associate Professor then a Professor at TELECOM Paris Tech. Since March 2005, Marie-Laure Boucheret is a Professor at the National Polytechnic Institute of Toulouse (ENSEEIHT - University of Toulouse). She is also with the Signal and Communication group of the IRIT Laboratory. Her fields of interest are signal processing for communication and satellite communications.

Jean-Baptiste Dupé graduated from INP-ENSEEIHT, Toulouse, France in 2012. He received the Ph.D in satellite telecommunications from the University of Toulouse in 2015.Since then, he has worked as a research engineer at Centre National d'Etudes Spatiales (CNES), focusing on MAC layer issues, resource allocation, scheduling and random access protocols.

Mathieu Gineste received a PhD in Computer Science in 2006, from Paris VI University (Pierre et Marie Curie). He did his PhD research jointly with the computer science department at ISAE and Thales Alenia Space. He worked 2 years as a researcher at LAAS-CNRS laboratory (French Scientific Research Centre) and is currently working in Telecom R&D department of Thales Alenia Space. He has been involved in many European and national R&D projects (EC, ESA, and CNES) dealing with satellite communication enhancements and end-to-end QoS in heterogeneous networks and was TAS technical responsible for the design of a new satellite communication standard for safety aeronautical services (IRIS/ Antares). He worked on the design of the Space Gate ground product and on system and access architecture for the Iridium next program. He is currently responsible of the end-to-end network architecture for TELESAT LEO constellation and contribute to the

system resource management for the same program. His research interests concern resource management, AI, access layer, M2M/IoT communications, satellite communication, end-to-end QoS management, transport layer, and cross-layer design.

How to cite this article: Zamoum S, Lacan J, Boucheret M-L, Dupe J-B, Gineste M. Complexity analysis for recent ALOHA random access techniques in satellite communications. *Int J Satell Commun Network*. 2020;1–18. https://doi.org/10.1002/sat.1370