# Pseudo-random Aloha for inter-frame soft combining in RFID systems

Castiglione, Paolo ; Ricciato, Fabio ; Popovski, Petar

*Published in:*
IEEE 2013 18th International Conference on Digital Signal Processing

*DOI (link to publication from Publisher):*
10.1109/ICDSP.2013.6622746

*Publication date:*
2013

*Document Version*
Early version, also known as pre-print

# Pseudo-Random Aloha for Inter-frame Soft Combining in RFID Systems

Paolo Castiglione*†, Fabio Ricciato‡†, Petar Popovski§
*AKG Acoustics GmbH, Austria
†Telecommunications Research Center Vienna (FTW), Austria
‡Università del Salento, Italy
§Aalborg University, Denmark

***Abstract*—In this work we consider a recently proposed variant of the classical Framed Slotted-ALOHA where slot selection is based on a pseudo-random function of the message to be transmitted and of the frame index. We couple this feature with convolutional encoding, that allows to perform Inter-frame Soft Combining (ISoC) of multiple (re)transmission attempts of the same payload across different frames. The ISoC scheme, proposed here for the first time, requires less memory usage and computational complexity at the receiver digital signal processor compared to existing techniques based on inter-frame signal cancellation (instead of combining). Numerical simulation results show that the ISoC scheme brings a noticeable throughput gain over traditional schemes in a dense RFID scenario with multiple concurrent Tag transmissions.**

## I. Introduction and motivations

The popular RFID standard EPC Global Generation-2 or simply "Gen2" [1] uses an interrogation scheme based on Dynamic Framed Slotted Aloha (DFSA) to deal with collisions among the tags. This choice of the MAC protocol is motivated by the restrictions on the passive tag and a rather simplistic model of the receiver chain at the Reader, based on the assumption that tag collisions[1] are always destructive events causing the loss of all colliding messages. This motivates the adoption in Gen2 of a Collision Avoidance (CA) mechanism to improve the contention efficiency: DFSA is executed by using shorter 16-bit messages, termed `RN16`, and upon receiving the 16-bit `ACK` from the reader, a tag sends its full 128-bit ID. Since at most one tag can be acknowledged in each slot, the `RN16`/`ACK` exchange serves as a channel reservation mechanism, and therefore collisions can occur only between short `RN16`, not between full ID messages.

A number of recent works have challenged the core assumption in the DFSA protocol, namely that the collisions are destructive and thereby useless. It was shown that the so-called Collision Recovery (CR) techniques can be successfully applied to disentangle Tag signals colliding in the same slot (see [2], [3] and references therein). Moreover, a new research direction [4]–[7] in random access protocols has been recently started in which the random access process inherently embraces the collisions and uses Successive Interference Cancellation (SIC) *across different slots* to decode the collided packets at the receiver. As pointed out in [7], with CR methods in place at the receiver *collisions cease to be a problem and become an advantage* (at least to some degree) calling for the elimination of the CA mechanism based on `RN16`/`ACK` exchange. The two recent works [6] and [7] have independently proposed two related methods for canceling the correctly decoded signals from past slots in order to unveil other signals colliding therein. This procedure is *backwards cancellation*, enabled by replacing the *random* slot selection, featured in classical ALOHA, with *deterministic* slot selection based on a pseudo-random function of the message to be transmitted (and of the frame index). In this way, once that a generic tag message has been correctly decoded after $k$ transmission attempts, the receiver can deterministically identify the position of the $k-1$ past transmission slots. This simple but powerful idea has been used in [7] in a method termed Inter-frame SIC (ISIC).

In principle, backward cancellation based on pseudo-randomization can be applied independently from the choice of the modulation and coding. Therefore, ISIC from [7] does not require any change to the standard Gen2 PHY format, i.e., ASK modulation with Miller encoding. Compliance with legacy PHY specification is a clear advantage. However, on the quantitative side, the actual throughput gain achievable by ISIC does depend on the signal format and, as we will see, it would benefit from a more robust encoding. Another aspect to be considered is that the implementation of a full backward cancellation scheme requires the receiver to keep in memory all signal samples received since the beginning of the reading cycle.

In this follow-up work we explore an alternative scheme where slot pseudo-randomization is leveraged to enable *soft combining across different slots* instead of cancellation. Following [7], we present the concept of Inter-frame Soft Combining (ISoC) in the framework of a simple Framed Slotted-ALOHA protocol with fixed frame size. This allows a more direct comparison between ISoC and ISIC on the basis of a common MAC scheme.

The basic idea of ISoC is to combine soft samples from different slots where the same tag message was (re)transmitted in order to increase the probability of successful decoding.

[1]We refer hereafter exclusively to "tag collisions" (in uplink), i.e. when two or more tags collide on the same slot at the reader receiver. Reader collisions (in downlink) caused by the contemporary transmission of multiple readers are not relevant to this work.

978-1-4673-5807-1/13/$31.00  

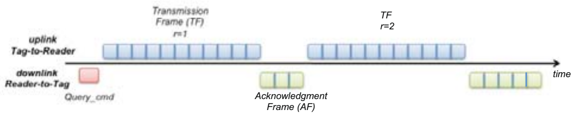


Fig. 1. Reference protocol timeline.

The rationale for that lies in the fact that *often the receiver can recover from a single slot only a fraction of the message information bits, not all: whenever such a fraction is sufficient to identify the position of another slot (past or future) containing the same message, the receiver can attempt to combine the two signals*. In order to increase the ISoC gain, we propose to encode the transmitted bits with a convolutional code. In so doing, we are departing from the PHY specification of Gen2. On the other hand, the implementation of ISoC is considerably less resource demanding than ISIC. First, in terms of memory: for binary ASK modulation ISoC can be implemented with a single soft value of 4 bits per symbol, while ISIC would require the storage of multiple 16-bit complex samples per symbol (the exact number depending on computation vs. memory trade-off of the specific implementation). Second, ISoC does not require the accurate reconstruction of the received signal in all past frames, as needed in ISIC for the purpose of cancellation. In absolute terms, the implementation complexity of ISoC depends on the number of attempted signal combinations $M$, however the numerical simulations presented hereafter show that small values of $M$ are sufficient to obtain a visible gain over legacy techniques.

Finally, it should be remarked that ISIC and ISoC are not mutually exclusive: if pseudo-random slot selection is employed, one can choose to implement ISIC, ISoC or both in the same receiver. In this work, however, we compare ISIC and ISoC separately for the same reference scenario, and leave the integration of these two techniques to future work.

## II. Inter-Frame Soft Combining (ISoC)

### A. Protocol Overview and Reference Scenario

We consider to elementary version of the (static) Framed Slotted-ALOHA protocol considered in [7] and depicted in Fig. 1. The initial `Query Command` (`QC`) sent by the Reader starts a new Reading Cycle (resets all Tag flags, broadcasts initialization parameters, etc.). The `QC` message is followed by a sequence of alternating Transmission Frames (TF) in uplink and Acknowledgment Frames (AF) in downlink. Each TF is divided into $K$ "transmission slots" of fixed duration sufficient to accommodate the 128-bit Tag ID plus the preamble. Note that no CA mechanism is considered, i.e., the `RN16`/`ACK` messages of Gen2 have been eliminated.

At each frame, the generic Tag selects one slot according to the pseudo-random function described later, and transmits its 128-bit ID therein. It then listens to the AF: if acknowledged it will leave the Reading Cycle, otherwise it will retry at the next frame.

In our simulations we retain the Gen2 standard compliant preamble and modulation format, i.e. binary ASK. For the payload we replace the Miller encoding with a convolutional code as explained in Sec. II-C. Note however that synchronization and channel estimation are performed based on the Gen2 standard compliant preamble (Miller-8 preamble). The Tag signal is attenuated by a frequency-flat slow fading channel and corrupted by the additive interference due to collisions from other Tags plus the Gaussian noise.

The following notation is introduced:

- $i \in \{1, 2 \ldots I\}$ is the Tag index.
- $\mathbf{x}_i$ is the 128 bits payload (ID) of tag $i$.
- $r$ is the TF index.
- $K$ is the number of slots in a generic TF. For simplicity, we assume $K = 2^n$.
- $s_{i,r} \in \{1, \ldots, K\}$ is the slot index selected by the $i$th Tag in the $r$th TF.
- $\mathbf{c}_i = [c_{i,1}, \ldots, c_{i,N}]$ is the coded payload $\mathbf{x}_i$.
- $\mathbf{w}_i$ is the vector of soft values corresponding to $\mathbf{c}_i$ computed at the receiver[2].

### B. Slot Selection for ISoC: The Tradeoff

In order to support ISoC, the pseudo-random function $h(\cdot) : (\mathbf{x}_i, r) \mapsto s_{i,r}$ mapping the payload $\mathbf{x}_i$ to the slot position $s_{i,r}$ in the $r$th frame must fulfill the following requirements:

- $s_{i,r}$ must be deducible from the soft coded symbols $\mathbf{w}_i$ computed at the receiver;
- the probability to correctly extract $s_{i,r}$ from the corrupted received signal must be as high as possible;
- the sequence of slot positions must emulate a memoryless random process to keep the probability of repeated collisions low.

To address the above requirements, we make the *election of the random access slot dependent on the data that is sent in that slot*. This approach is related to the notion of *protocol coding* [8] where the actions taken by a communication protocol are used to encode data. In order to motivate such an approach, let us assume that user $i$ sends the same packet in $M$ consecutive frames. The slot position $s_{i,r}$ in each frame is determined by selecting $n = \log_2 K$ bits of $\mathbf{c}_i$ according to a bitmask. Since the same data is sent in all slots, $s_{i,r}$ would be constant for all $r$. If the receiver learns $s_{i,r}$, then the benefit is two-fold: (a) it perfectly recovers $n$ bits of $\mathbf{c}_i$ and (b) it can combine the received soft values of all frames in order to make a reliable decision on the remaining bits of $\mathbf{c}_i$.

The first objection to the described idea is that $s_{i,r}$ cannot be known perfectly. The receiver should compute $s_{i,r}$ by applying the bitmask to the hard-converted bits of the soft values $\mathbf{w}_i$. This implies a certain iterative process, where some of the bits in $\mathbf{c}_i$ from a slot are recovered, based on those a pointer to the slot position is extracted, then these are combined with

[2]The soft values $\mathbf{w}_i$ are here the well-known log-likelihood metrics for coherent binary modulation over the Gaussian channel. This metric is suboptimal, since the Gaussian model for the interference distribution is a coarse approximation, and the results can be further improved by adopting more sophisticated choices.

the values from the new slot, etc. The second objection is that the $n$ data bits used to select the slot can be identical for two users, such that these users will repeatedly collide in all $M$ slots. To alleviate this we can use one or both of the following: (1) the group of $n$ bits used as an input to $h(\cdot)$ is changed from frame to frame in a deterministic way (e.g., by cyclic shift) and (2) only part of the bits used as an input to $h(\cdot)$ are random, not related to the data being sent, as in the usual ALOHA. Regarding (1), the bitmasks selected for successive frames indexes should have minimal intersection, in order to emulate a memoryless process. The problem with (2) is that it introduces uncertainty in the pointer to the slot position.

Hereafter, we devise a practical scheme that allows to control the above trade-off between randomization and predictability of the slot positions via two explicit parameters.

### C. Slot Selection: Randomization and Protocol Coding

Although the payloads can be scrambled and interleaved, this might be a too weak countermeasure to avoid repeated collisions with a deterministic bit mask. We therefore use *controlled randomization* in $h(\cdot)$ in the following way. We use the bitmask to select data bits from $\mathbf{c}_i$. Only $n-q$ of $\mathbf{c}_i$ are selected for the computation of $s_{r,i}$ and they are supplemented with $q$ random bits, which are unrelated to the data and known only at the transmitter. The receiver must then explore $2^q$ possible realizations to establish the slot positions of the signal in another frame. Clearly, one can increase the degree of randomness by increasing $q$ (it becomes a conventional ALOHA for $q = n$) at the expense of larger complexity of the combining process. This trade-off between complexity and degree of randomness must be tuned properly in the system design phase. On the other hand, since only $n-q$ deterministic bits need to be correctly estimated (instead of $n$), the probability that the correct slot positions are detected increases with a beneficial effect for ISoC. In the following, we will denote the random function output as a vector, e.g., $\mathbf{s}_{r,i} = h(\mathbf{c}_i, r)$, that contains the $2^q$ equally probable realizations of $s_{r,i}$.

Next, we propose a coding technique that allows the efficient use of the $n-q$ redundant bits in the data. We adopt as code the mothercode of the widespread family of Rate-Compatible Punctured Convolutional (RCPC) codes [9]. In our solution, the slot position bits are selected via the puncturing table (acting as bitmask) of one daughtercode of the same family. In this way, we ensure that the $n-q$ redundant bits provided by the known current slot position are able to correct the most likely errors (with small Hamming distances) in the region of code where they belong. Slot positions act as incremental redundancy that is efficiently exploited by RCPC codes. In practice, at the decoder the current deterministic slot position bits of $s_{r,i}$ are used as replacement for the corresponding received soft coded symbols of $\mathbf{w}_i$. As these bits are known, the corresponding log-likelihood values exhibit the largest possible magnitude. The performance enhancement attained by means of the insertion of known slot position bits in $\mathbf{w}_i$ is here referred to as *protocol coding gain*.

### D. Protocol Coding Gain vs Randomization

A higher correlation between the subsets of $\mathbf{c}_i$ selected to represent the slot positions in neighboring frames leads to higher protection of these slot positions (see discussion below on a practical way to achieve it), but that comes at the cost of increased probability of repeated collisions, i.e. the same set of Tags colliding again in future slots. Converesely, lower correlation on the subsets of $\mathbf{c}_i$ reduces the probability of repeated collisions but increases the probability of errors in establishing the slot positions.

In order to balance the probability of correct detection of slot positions with their degree of randomness across the frame indexes, we propose to select $s_{r,i}$ as follows: the bits of the position $s_{r,i}$ are the first $n-q$ punctured bits obtained by applying the puncturing table of the selected daughtercode to the cyclically shifted mothercode bits $[c_{\tau r,i} \dots c_{N,i}, c_{1,i} \dots c_{\tau r-1,i}]$, where $\tau \geq 1$ is the shift per frame[3]. The remaining $q$ bits are chosen randomly. As illustrated by the example in Fig. 2, by selecting a small shift $\tau$, the slot position bits of neighboring frames belong to overlapping regions of the code, i.e., they likely belong (completely or in part) to the same error event patterns of the convolutional code. Once the known slot position bits are inserted into the received soft coded bits $\mathbf{w}_i$, the Viterbi decoder might correct these error events (see [9] and references therein), hence, the reliability of the slot bits of the neighbouring frames can improve. In the particular example depicted in Fig. 2, we observe that the insertion of $n-q=5$ known bits allows to correct the first error event and to recover the slot bits associated to the next frame when $\tau = 1$. On the other hand, the larger shift $\tau = 20$ places the slot bits associated to the neighbouring frame in a different region of the code, such that they remains corrupted.

In summary, we have introduced two parameters, namely $q$ and $\tau$, that act as tuning knobs to trade-off randomization vs. identificability of the slot positions along the two dimensions of frame index and payload. The parameter $q$ controls the trade-off between randomization of the slot positions and search complexity. The parameter $\tau$ governs the trade-off between randomization of slot position across different frames and the ability to exploit the protocol coding gain to improve the detection of the slot positions over neighbouring frames.

### E. Proposed ISoC algorithm

The function $h(\cdot)$ defined above is known at the receiver. If the detection of payload $\mathbf{x}_i$ in current frame $r = \bar{r}$ and slot $s_{\bar{r},i} = \bar{s}$ fails, the receiver tries to identify the slot positions where the same payload has been transmitted in other frames, and therefore combines the soft values of the signals therein. In general, the more signals are combined, the larger the soft combining gain can get. However, as there is no guarantee in general that the slot positions are correctly identified, one needs to develop an ISoC algorithm that is able to balance the soft combining gain with the risk of combining signals that

[3]We assume here that the codeword length is sufficiently larger than the typical Reading Cycle duration such that $N \gg \tau \cdot r$.

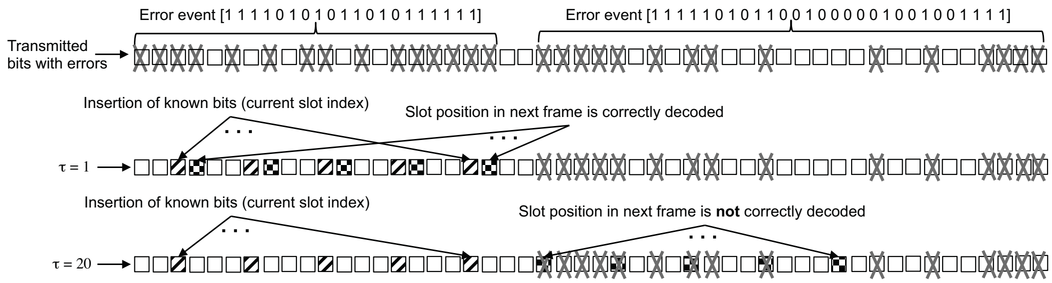


Fig. 2. Segment of bits transmitted by Tag $i$ in slot $s_{r,i}$ of the frame $r$ at the input of the receiver Viterbi decoder (Top). The bits are encoded by the rate $1/4$ RCPC mothercode with octal generators $(23, 35, 27, 33)$. The slot bits are placed according to the periodic bitmask $[0001]$, that is the puncturing table associated to the rate $1/3$ doughter code. Two consecutive error events at free distance $= 15$ occur, i.e., $[11110101011010111111]$ and $[1111101011001000001001001111]$. The insertion of the known current slot position $s_{r,i}$ bits (marked by stripes) allows the correction of the first error event. If shift $\tau = 1$ is selected, the slot position bits associated to the next frame (marked by squares) are correctly decoded, i.e., the received signal correctly points to the next frame slot position $s_{r+1,i}$ (Center). Instead, for the larger shift $\tau = 20$ the decoding of the slot position $s_{r+1,i}$ fails (Bottom).

do not contain the same codeword. This risk increases with the number $m$ of combined signals, which is here referred to as *soft combining degree*.

In what follows, we describe the main features of the proposed ISoC algorithm, which is rigorously detailed in Algorithm 1. The ISoC algorithm is in force upon failure of the decoding at frame $r = \bar{r}$ and slot $\bar{s}$. The receiver evaluates signal combinations with increasing soft combining degree (but limited to a determined value, i.e., $m \leq M$) until decoding is successful. Firstly, the slot positions $\boldsymbol{\sigma} = h(\mathbf{w}_i, \rho)$ of the previous frames $r = \rho$ are estimated by applying the bitmask given in Sec. II-D to the hard-converted bits of $\mathbf{w}_i$. The algorithm will combine each signal $\mathbf{v}^k$ associated to the slot positions $\sigma_k \in \boldsymbol{\sigma}$ with the signal $\mathbf{w}_i$ in slot $\bar{s}$ *only* if $\bar{s} \in h(\mathbf{v}^k, \bar{r})$. This is a crucial step in order both to reduce the risk of combining signals with different payloads and to decrease the algorithm complexity. In other words, we impose that the signals $\mathbf{v}^k$ and $\mathbf{w}_i$ mutually *point* to each other position. If this is true, the signal $\mathbf{v}^k$ is combined with $\mathbf{w}_i$ using a form of soft combining[4]. We specifically adopt *maximum ratio combining*, and denote this operation by $\boldsymbol{\xi} = \mathrm{ISoC}(\mathbf{w}_i, \mathbf{v}^k)$, where $\boldsymbol{\xi}$ indicates the combined soft values signal. The combined signal is then fed into the soft input decoder. If decoding fails, the combined signal $\boldsymbol{\xi}$ is stored in the set $\mathcal{W}^{m=2}$, together with the information regarding the combined signal position (stored in sets $\mathcal{R}^{m=2}$ and $\mathcal{S}^{m=2}$). This stored data will be used in the next iteration, in which the soft combining degree $m$ is increased by 1. Similarly as before, in each of the successive iterations the $l^{\text{th}}$ signal $\mathbf{w}^{m-1,l}$ contained in set $\mathcal{W}^{m-1}$ is combined with each of the pointed signals $\mathbf{v}^k$, *only* if $\mathbf{v}^k$ reciprocally points to the positions of the signals that were combined in $\mathbf{w}^{m-1,l}$ over the previous iterations. The obtained signal $\boldsymbol{\xi} = \mathrm{ISoC}(\mathbf{w}^{m-1,l}, \mathbf{v}^k)$ exhibits a soft combining degree $m$. Again, if decoding fails, signal $\boldsymbol{\xi}$ is stored for being used in the next iterations. The algorithm stops as soon as the decoding of $\boldsymbol{\xi}$ is successful. Note that the number of iterations where $m$ increases needs to be limited to a finite value $M \leq \bar{r}$, in order to bound the combinatorial complexity such that the algorithm is forced to stop after a controlled amount of time. Nevertheless, a larger $M$ can lead to better decoding performance.

We additionally remark that, in a system that employs SIC for collision recovery, the ISoC algorithm can aid the detection of the residual signal that was not successfully decoded during one SIC iteration. If correct detection occurs, other SIC iterations over the same slot $\bar{s}$ become possible (provided that there is some residual signal after cancellation).

The performance of the approach described in this section is assessed in numerical simulations in Sec. IV. We have adopted the rate-1/4 RCPC mothercode with octal generators $(23, 35, 27, 33)$ and the puncturing table of the respective rate-1/3 daughtercode [9].

## III. Numerical Results

We consider a reference scenario where a large number of Tags, in the order of hundreds or even thousands, must be read by a single Reader in the shortest possible time. The primary goal is therefore to minimize the total Reading Cycle time, or equivalently maximize the reading throughput. This scenario is representative of practical applications where *(i)* the size of tagged items is small relative to the Reader range (note that Tag signals can be correctly received at several tens of meters, see e.g. [10]) and *(ii)* the tagged items and the Reader are in relative motion, thus limiting the average coverage time. For example, think to a mobile Reader inventorying a large warehouse [5], or a moving cart packed with tagged items moving through a RFID Reading gate.

[4]We note for instance that the two signals could be combined by simply substituting the punctured soft values bits of one signal with those of the other according to a puncturing table, without performing any addition operation.

[5]See e.g. "How Can I Read 1000 Tagged Apparel Items Within a Small Area?" at http://www.rfidjournal.com/expert/entry/8891.

**Algorithm 1** ISoC algorithm at current frame $r = \bar{r}$.

**Notations:**

- $m$ is the number of combined signals, i.e., the *soft-combining degree*;
- the $l^{\text{th}}$ vector $\mathbf{w}^{m,l} \in \mathcal{W}^m$ contains the soft values of the combination of $m$ signals;
- the $l^{\text{th}}$ vector $\mathbf{r}^{m,l} \in \mathcal{R}^m$ contains the $m$ frame indexes associated to the signals combined in $\mathbf{w}^{m,l}$;
- the $l^{\text{th}}$ vector $\mathbf{s}^{m,l} \in \mathcal{S}^m$ contains the $m$ slot indexes associated to the signals combined in $\mathbf{w}^{m,l}$, i.e., $s_j^{m,l}$ and $r_j^{m,l}$ define the slot and frame position of the $j^{\text{th}}$ signal combined in $\mathbf{w}^{m,l}$;
- vector $\boldsymbol{\sigma} = h(\mathbf{w}^{m-1,l}, \rho)$ contains the $2^q$ slot indexes $\sigma_k \in \boldsymbol{\sigma}$ of frame $\rho$ pointed by $\mathbf{w}^{m-1,l}$;
- $\mathbf{v}^k$ is the soft values signal in slot $\sigma_k$ and frame $\rho$;
- $\boldsymbol{\xi} = \mathrm{SoCo}(\mathbf{w}^{m-1,l}, \mathbf{v}^k)$ is the soft combination of $\mathbf{w}^{m-1,l}$ and $\mathbf{v}^k$;
- correct/incorrect decoding is denoted by $\mathrm{success} = 1/0$ (in case of SIC, "decoding" refers to the last decoding iteration over the residual signal);
- the operation of stacking element $a$ in vector $\mathbf{a}$ is denoted by $[\mathbf{a}, a]$;

**Start of the algorithm at the first slot $\bar{s}$ that contains signal:**
Initialization:
Detection with/without SIC at slot $\bar{s}$ (if there is signal);
Initialize: vector $\mathbf{w}^{1,1} \equiv \mathbf{w}_i$ contains the soft coded values of the captured signal (or of the residual signal in last iteration of SIC);
Initialize: $\mathcal{W}^m \equiv \varnothing$, $\mathcal{R}^m \equiv \varnothing$, and $\mathcal{S}^m \equiv \varnothing$, for $2 \leq m \leq M$;
Initialize: $\mathbf{r}^{m=1,l=1} \equiv \bar{r}$ and $\mathbf{s}^{m=1,l=1} \equiv \bar{s}$;
Initialize: $m = 2$
**while** $m \leq M$ && $\mathrm{success} = 0$ (increase $m$) **do**
  Initialize: signal index $l = 1$;
  **while** $l \leq |\mathcal{W}^{m-1}|$ && $\mathrm{success} = 0$ (go through signals in $\mathcal{W}^{m-1}$) **do**
    Initialize: frame index $\rho = \bar{r} - 1$;
    **while** $\rho \geq 1$ && $\mathrm{success} = 0$ (go through past frames) **do**
      **if** $\rho \notin \mathbf{r}^{m-1,l}$ (skip signals of same frame) **then**
        Find slots pointed by $\mathbf{w}^{m-1,l}$: $\boldsymbol{\sigma} = h(\mathbf{w}^{m-1,l}, \rho)$;
        Initialize: pointed vector index $k = 1$;
        **while** $k \leq 2^q$ && $\mathrm{success} = 0$ (go through all $\mathbf{v}^k$) **do**
          **if** $\forall j : s_j^{m-1,l} \in h(\mathbf{v}^k, r_j^{m-1,l})$ (signal $\mathbf{v}^k$ points to all components of the combined signal $\mathbf{w}^{m-1,l}$) **then**
            Soft-combining: $\boldsymbol{\xi} = \mathrm{SoCo}(\mathbf{w}^{m-1,l}, \mathbf{v}^k)$;
            Decoding signal $\boldsymbol{\xi}$;
            **if** $\mathrm{success} = 0$ **then**
              **if** $m < M$ (there is a future iteration) **then**
                Update: $\mathcal{W}^m \leftarrow \{\mathcal{W}^m, \boldsymbol{\xi}\}$;
                Update: $\mathcal{R}^m \leftarrow \{\mathcal{R}^m, [\mathbf{r}^{m-1,l}, \rho]\}$;
                Update: $\mathcal{S}^m \leftarrow \{\mathcal{S}^m, [\mathbf{s}^{m-1,l}, \sigma_k]\}$;
              **end if**
            **else if** SIC is in place **then**
              Cancellation + SIC detection;
              **if** $\mathrm{success} = 0$ (after last SIC iteration) **then**
                Save soft values in $\mathbf{w}^{1,1}$ and restart;
              **end if**
            **end if**
          **end if**
          Update pointed vector index: $k \leftarrow k + 1$
        **end while**
      **end if**
      Update frame index: $\rho \leftarrow \rho - 1$
    **end while**
    Update signal index: $l \leftarrow l + 1$
  **end while**
  Update soft-combining degree: $m \leftarrow m + 1$
**end while**
Go to next slot $\bar{s}$ that contains signal and restart;

We run a MATLAB© simulation to compare quantitatively the performance of five PHY receiver structures in combination with fixed-size Frame Slotted-ALOHA:

- I. **Capture**: the traditional receiver without any collision recovery technique, where the strongest signal can be decoded only due to "radio capture" [11].
- II. **SIC**: simple intra-slot SIC.
- III. **Capture + ISoC**: ISoC implemented up to soft combining degree $M$.
- IV. **SIC + ISoC**: ISoC implemented in combination with intra-slot SIC.
- V. **Inter-frame SIC**: the exhaustive ISIC scheme presented in [7].

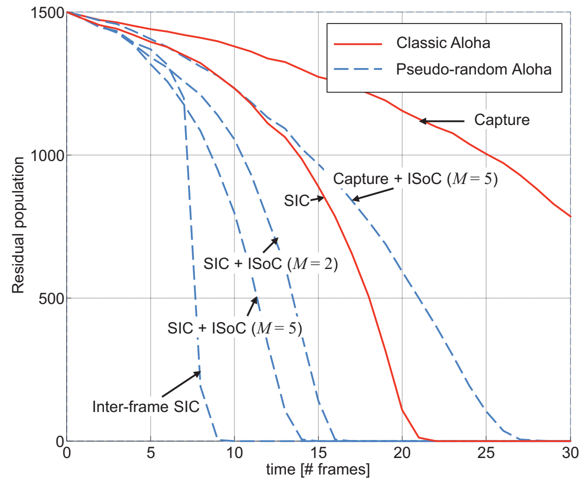


Fig. 3. Reading Cycle of $I = 1500$ tags with slot selection according to the bitmask scheme described in Sec. II-D with parameters $\tau = 1$ and $q = 2$. Channel (block fading) is Rician with K-factor equal to 3dB and SNR equal to 20dB for all tags. Frame duration is equal to $K = 128$ slots.

In Fig. 3 we show the residual population vs time (TF index) for a sample Reading Cycle realization with $I = 1500$ Tags for all receiver structures. The Tags transmit asynchronously, with a relative timing offset up to 2 symbol periods. In our simulator, channel estimation and timing recovery are performed via correlation to the known Gen2 preamble, hence synchronization and channel estimation errors are taken into account. Other sources of non-ideality like e.g. phase noise and clock jitter are instead neglected.

Notably Inter-frame SIC outperforms all considered techniques. We remark that the relative gain of ISIC observed in this scenario, where we have adopted a rate-1/4 RCPC encoding, is considerably higher than what was found in an earlier work [7] with Gen2 standard compliant Miller-8 encoding (rate-1/16). This clearly indicates that ISIC is sensitive to the encoding format and benefits from more powerful encoding.

From Fig. 3 it can be seen that ISoC has a dramatic gain over simple capture, and the combination SIC + ISoC outperforms the simple SIC. In other words, ISoC can add a visible gain already for small values of $M$ when combined with existing collision recovery algorithms. As expected, the ISoC performance improves when the soft combining degree limit $M$ is increased.

## IV. Conclusions and Outlook

In this work we have presented the idea of Inter-frame Soft Combining (ISoC), a novel approach that builds upon the more general frameworks of pseudo-random ALOHA [6], [7] and protocol coding [8]. The numerical simulation results presented in this work clearly indicate that ISoC has the potential of delivering throughput gains at a relatively modest cost in terms of implementation complexity. However, further investigations are needed to assess the impact of the various non-idealities in the signal transmission and reception chain onto the actual performance, a task that is better achieved by testbed measurements rather than simulations. Therefore, in the progress of the work we are planning to prototype both ISoC and ISIC in Gnu Radio in order to measure their actual performance in real-world operating conditions, and at the same time precisely quantify the memory/computation demands with reference to a common practical implementation.

The relatively low complexity of our ISoC algorithm, compared to more resource-demanding Inter-frame SIC [7] and frameless/rateless approaches [6], makes it appealing to practical DSP implementations. Our technique is able to take advantage of the (so far almost unexplored) interactions between protocol coding — and specifically pseudo-random ALOHA — and well-established base-band digital signal processing such as combining and cancellation in a novel form. In this regard, we believe that the ISoC and ISIC algorithms belong to a much wider family of techniques whose applications are not limited to RFID systems. An interesting evolution of this work will be to consider the combination of ISoC and ISIC with adaptive strategies, e.g., for frame-size and encoding parameters.

## ACKNOWLEDGMENTS

This work was supported by the project NFN SISE (S10607) funded by the Austrian Science Fund (FWF) and by the FTW strategic projects N-0 and I-0 funded in the framework of the COMET program. The work of Petar Popovski was supported by the Danish Council for Independent Research within the Sapere Aude Research Leader program, Grant No. 11-105159.

## REFERENCES

[1] EPCglobal, "EPC Radio-Frequency Identify Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz," 2010.

[2] C. Angerer, R. Langwieser, and M. Rupp, "RFID reader receivers for physical layer collision recovery," *IEEE Trans. on Commun.*, vol. 58, December 2010.

[3] K. Fyhn, R. Jacobsen, P. Popovski, A. Scaglione, and T. Larsen, "Multi-packet reception of passive UHF RFID tags: A communication theoretic approach," *IEEE Trans. on Signal Processing*, vol. 59, September 2011.

[4] E. Cassini, R. D. Gaudenzi, and O. del Rio Herrero, "Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Scheme for Satellite Access Packet Networks," *IEEE Trans. Wireless Commun.*, vol. 6, pp. 1408–1419, Apr. 2007.

[5] G. Liva, "Graph-Based Analysis and Optimization of Contention Resolution Diversity Slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, pp. 477–487, Feb. 2011.

[6] C. Stefanovic, P. Popovski, and D. Vukobratovic, "Frameless ALOHA protocol for wireless networks," *IEEE Communication Letters*, vol. 16, December 2012.

[7] F. Ricciato and P. Castiglione, "Pseudo-random Aloha for enhanced collision-recovery in RFID," *IEEE Communication Letters*, vol. 17, March 2013.

[8] P. Popovski, Z. Utkovski, and K. F. Trillingsgaard, "Protocol coding through reordering of user resources, Part I: Capacity Results," *submitted*, 2012. http://arxiv.org/abs/1202.0307.

[9] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 38, pp. 389–400, Nov. 1988.

[10] D. DeDonno, F. Ricciato, and L. Tarricone, "Listening to tags: Uplink RFID measurements with an open-source software-defined radio tool," *IEEE Trans. on Instrumentation and Measurement*, vol. 62, Jan. 2013.

[11] M. Buettner and D. Wetherall, "A software radio-based UHF RFID reader for PHY/MAC experimentation," in *Proc. of IEEE RFID 2011*, April 2011.