

Experimental Validation of Anti-Collision Protocols for RFID Sensor Networks

Laura Arjona, Hugo Landaluze, Asier Perallos
Faculty of Engineering
University of Deusto
Bilbao, Spain
laura.arjona@deusto.es

Joshua R. Smith
Department of Computer Science and Engineering
University of Washington
Seattle, USA
jrs@cs.washington.edu

Abstract—Currently, there is an increasing interest in the use of RFID systems with passive or battery-less tags with sensors incorporated, also known as computational RFID (CRFID) systems. These passive tags use the reader signal to power up their microcontroller and an attached sensor. Following the current standard EPC C1G2, the reader must identify the tag (receive the tag's identification code) prior to receive data from its sensor. In a typical RFID scenario, several sensor tags share the reader interrogation zone, and during their identification process, their responses often collide, increasing their identification time. Therefore, RFID application developers must be mindful of tag anti-collision protocols when dealing with CRFID tags in dense RFID sensor networks. So far, significant effort has been invested in simulation-based analysis of the performance of anti-collision protocols regarding the tags identification time. However, no one has explored the experimental performance of anti-collision protocols in an RFID sensor network using CRFID. This paper: (i) demonstrates that the impact of one tag identification time over the total time required to read one sensor data from that same tag is very significant, and (ii) presents an UHF-SDR RFID system which validates the improvement of FuzzyQ, a fast anti-collision protocol, in relation to the protocol used in the current RFID standard.

Index Terms—RFID, anticollision, DFSA, passive sensors, CRFID, SDR.

I. INTRODUCTION

During the last few years Radio Frequency Identification (RFID) systems have been popularized due to their improved features over other auto identification techniques such as barcodes, or biometrics [1]. Recently, computational RFID (CRFID) systems have gained protagonism thanks to the use of sensors without the need of batteries [2], [3]. CRFID use passive or battery-less tags with sensors incorporated and are powered using the reader RF signal. A wireless sensor network using these passive sensors inherits all the benefits of RFID: passive operation, low cost, a small sticker form factor, and an operation range of over 10 meters from the reader [4].

An RFID sensor network is composed of at least one reader and several CRFID tags that include at least one sensor. The tag collision problem acquires the main focus when interrogating these tags. The communication channel is shared among them and, therefore, their responses need to be arbitrated in order to avoid simultaneous responses that will lead to collisions. This problem is one of the main cause of energy wastage and tag identification time increase. The solution to

the tag collision problem is the use of an anti-collision protocol that minimizes the impact of these collisions. The existing anti-collision protocols in the literature can be classified into two main groups: tree-based and aloha-based protocols [5].

Tree-based protocols [6], in essence, split colliding tags into subsets, and further split the subsets repeatedly up to the successful response of all the tags that are within the interrogation zone.

Aloha-based protocols present four main variants [5]. In Pure Aloha (PA), a tag responds to the reader command randomly after being energized. Slotted Aloha (SA) divides time into slots and schedules tags to respond only at the boundary of the time slots. Frame Slotted Aloha (FSA) and Dynamic Frame Slotted Aloha (DFSA) divide time into frames and frames into slots and mandate each tag to respond in only one slot per frame. The most popular aloha variant is DFSA, which focuses on adjusting the frame size L so that L is adapted to the set of tags to be identified. EPCglobal Class 1 Generation 2 (EPC C1G2), the current standard in RFID systems [7], adopts a DFSA strategy. This standard, also included in ISO 18000-6C, is used in every commercial reader.

A significant amount of literature can be found on simulation-based analysis of the performance of tree-based and aloha-based anti-collision protocols [6], [8]–[11]. However, to the best of our knowledge, no experimental performance of anti-collision protocols in RFID sensor networks can be found. The particularities of an RFID sensor network involves not only the identification of all the sensors in the network, but also the time needed to read all those sensors. The presented work analyzes the relevance of these two processes and the time involved, and presents the experimental performance of an RFID sensor network with CRFID tags using two different aloha-based anti-collision protocols from the literature: Slot Counter [7] and FuzzyQ [10]. The Slot Counter is the reference protocol in RFID technology, because it is used in the standard EPC C1G2. The FuzzyQ protocol is also evaluated because it achieves a more abrupt variation of the frame size than Slot Counter, adapting faster to a dynamic tag population. Thus, FuzzyQ is a suitable candidate for RFID sensor networks, where fast and dynamic tags identification is sought. Thus, this paper contains the following two contributions:

- An analysis of the time needed to identify a tag (receive its Electronic Product Code (*EPC*)) with respect to the time needed to read its sensor.
- A comparative of the performance evaluation of two anti-collision protocols physically implemented with an UHF-SDR RFID experimentation hardware. The evaluation is made in terms of their tag identification rate.

The rest of the paper is organized as follows. Section II describes the background for the presented analysis and experimentation. Section III analyses the relation between the time required to identify one tag and the time to read its sensor. Section IV presents the implementation of Slot Counter and FuzzyQ protocols. The performance evaluation of the implementation of both protocols is given in Section V. And finally Section VI concludes this paper.

II. BACKGROUND

First, some definitions are provided to properly set the background of this paper and to better understand its main contributions. After this, the two anti-collision protocols implemented in this work are presented: Slot Counter and FuzzyQ.

- *Sensor tag*: it refers to an RFID tag with an integrated sensor, such as an accelerometer.
- *Tag identification*: it refers to the whole needed process to read the tag identifier or *EPC*.
- *Slot*: period of time that separates the tags' responses. Conventionally, three types of slots are considered, in terms of the tags' responses to the reader's commands: single (only one tag replies), collision (more than one tag replies in the same slot), and idle (none of the tags reply).
- *Frame*: sequence of L slots, where L represents the frame size. An identification process is composed of a set of frames and tags can respond in only one slot per frame.
- *Inventory round*: the period of time that begins when the reader transmits the initial command, and ends when the reader interrupts the identification process and the tags loose their state. This work assumes that an inventory round ends when all the tags in the reader interrogation zone have been identified.

A. Slot Counter from EPC C1G2

EPC C1G2 specifies a DFSA protocol to arbitrate collisions: the Slot Counter protocol. The reader starts the identification process by transmitting a Query (Q_c) command. After the first tags' response, the reader alternates between QueryAdjust (QA) and QueryRep (QR) commands. QA starts a new frame with the updated size L and implies that tags must randomly select a slot in the frame (the initial value of their internal slot counter SC), while QR tells the tags to decrement SC . Thus, when $SC=0$, the tag transmits a 16-bit random number $RN16$.

The frame size is set as $L=2^Q$, and its value is dynamically updated with the parameters Q and C , where $Q=\text{round}(Q_{fp})$, and Q_{fp} is updated according to the time slot occupancy with the value of C :

- Single slot: Q_{fp} remains unchanged. In this case, the reader replies to the tag with an acknowledge (ACK) command followed by the same $RN16$ received by the tag; and the tag transmits its *EPC*.
- Collision slot: $Q_{fp} = Q_{fp} + C$.
- Idle slot: $Q_{fp} = Q_{fp} - C$.

The standard does not specify the selection of C . It only recommends using high C values if the previous Q value was low and vice versa, in the range of $[0.1, 0.5]$.

B. FuzzyQ protocol

The protocol FuzzyQ was introduced in [10] for fast RFID tag identification. This protocol follows a similar procedure to the standard EPC C1G2 but integrates a fuzzy logic controller to update L considering the current Q and the collision or idle rate of the current frame. As a result, FuzzyQ significantly decreases the tag identification time by efficiently updating L in a dynamic and adaptive way.

A fuzzy rule based system (FRBS) is defined to model the current Q and the idle or collision rate as fuzzy sets to adaptively calculate L . At slot $p=L/9$ of every frame, L is updated with the FRBS. The value of p is analyzed and set in [10]. Then, a new frame is started if L is modified. Otherwise, the reader continues the identification process in the following slot. Fig. 1 shows a flowchart of FuzzyQ protocol. It achieves an average improvement of 7% in the tag identification time with respect to Slot Counter protocol [10].

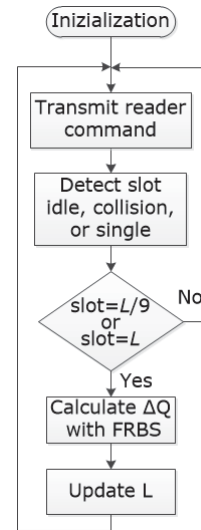


Fig. 1. Flowchart of FuzzyQ protocol

III. TAGS IDENTIFICATION IN RFID SENSOR NETWORKS

Following EPC C1G2 standard, the reader must identify a tag (receive the tag's *EPC*) prior to read data from its sensor. The time associated to these two processes is shown in Fig. 2. This section studies the effect of the time needed to identify all the tags inside the reader interrogation area (identification time) over the time needed to read all the tags' sensor (sensing time), in the single and multiple-tag environment.

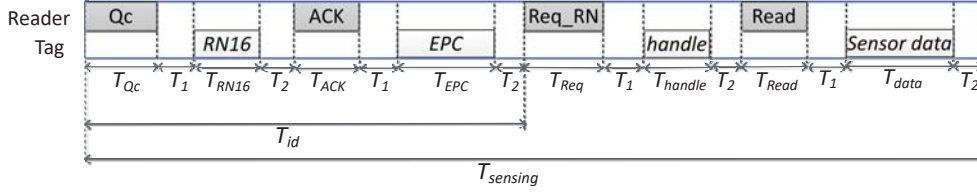


Fig. 2. Definition of tag sensor identification time and sensing time

First, a scenario with a single reader and a single tag inside the reader's interrogation zone is assumed. According to Fig. 2, the time required to identify one tag T_{id} is calculated as:

$$T_{id} = T_{Qc} + T_1 + T_{RN16} + T_2 + T_{ACK} + T_1 + T_{EPC} + T_2, \quad (1)$$

and the time required to read one sensor data from a tag $T_{sensing}$ is calculated as

$$T_{sensing} = T_{id} + T_{Req} + T_1 + T_{handle} + T_2 + T_{Read} + T_1 + T_{data} + T_2, \quad (2)$$

The parameters T_{Qc} , T_{ACK} , T_{Req} , and T_{Read} refer to duration of the reader's transmitted commands Qc, ACK, Req_RN, or Read, respectively, specified by the EPC C1G2 standard [7]. The parameters T_{RN16} , T_{EPC} , T_{handle} , and T_{data} refer to the duration of the tags' responses RN16, EPC, handle, and sensor data, respectively. Finally, T_1 refers to the time interval between a reader command and a tag response, while T_2 refers to the time interval between a tag transmission and a reader command. Table I summarizes the calculation of the reader and tags messages duration.

The duration of the reader's commands is calculated as the Reader-to-Tag synchronization time $T_{FSyncRT}$ or $T_{PreambRT}$, also defined in [7], plus the length in bits of each parameter divided by the reader data rate DR_r , calculated as $DR_r = 1/((T_{data0} + T_{data1})/2)$, where $T_{data0} = T_{ari}$, and $T_{data1} = 1.5 \cdot T_{ari}$. T_{ari} represents the reference time interval for a data-0 transmission. Thus, $T_{Qc} = T_{FSyncRT} + 22/DR_r$, $T_{ACK} = T_{FSyncRT} + 24/DR_r$, $T_{Req} = T_{PreambRT} + 40/DR_r$, and $T_{Read} = T_{PreambRT} + 58/DR_r$.

The duration of the tags' responses is calculated as the length in bits of each message (which includes the 6 bits of the Tag-to-Reader Preamble) divided by the tag data rate DR_t , calculated as $DR_t = BLF/M$, where BLF refers to the Backscatter-Link-Frequency, and M refers the number of subcarrier cycles per symbol. This work assumes that tags encode the backscattered data as FMO baseband ($M=1$). Thus, $DR_t = BLF$, $T_{RN16} = 23/BLF$, $T_{EPC} = 135/BLF$, and $T_{handle} = 38/BLF$. To calculate T_{data} , a WISP 5.1 tag [2] with an incorporated accelerometer sensor is considered. For this tag, a sensor read corresponds to 32 bits, and $T_{data} = 71/BLF$.

From Fig. 2, it is clear that the time required to identify one tag plays a significant role in the total time required to read one sensor data. To evaluate the effect of T_{id} over $T_{sensing}$, it is defined the factor F as the fraction of a sensor

TABLE I
EPC C1G2 TIMING PARAMETERS TO READ ACCELEROMETER SENSOR DATA FROM A WISP 5.1

Parameter	Description	Calculation
T_{data0}	Reader data-0 duration	T_{ari}
T_{data1}	Reader data-1 duration	$1.5 \cdot T_{ari}$
$T_{FSyncRT}$	Frame Sync. duration	$T_{del} + T_{data0} + RT_{cal}$
$T_{PreambRT}$	Reader Preamble duration	$T_{del} + T_{data0} + RT_{cal} + TR_{cal}$
DR_r	Reader data rate	$1/((T_{data0} + T_{data1})/2)$
DR_t	Tag data rate	BLF/M
T_{Qc}	Qc command duration	$T_{FSyncRT} + 22/DR_r$
T_{QA}	QA command duration	$T_{PreambRT} + 9/DR_r$
T_{QR}	QR command duration	$T_{PreambRT} + 4/DR_r$
T_{ACK}	ACK command duration	$T_{FSyncRT} + 24/DR_r$
T_{Req}	Req_RN command duration	$T_{PreambRT} + 40/DR_r$
T_{Read}	Read command duration	$T_{PreambRT} + 58/DR_r$
T_{RN16}	Tag RN16 duration	$23/BLF$
T_{EPC}	Tag EPC duration	$135/BLF$
T_{handle}	Tag handle duration	$38/BLF$
T_{data}	Tag sensor data duration	$71/BLF$

tag identification time over the total time required to receive one sensor data

$$F = \frac{T_{id}}{T_{sensing}}. \quad (3)$$

Next, F is evaluated and its upper and lower bound are calculated. For this purpose, it is considered a reader with an application-specific configuration following EPC C1G2 standard, and a WISP 5.1 tag. Evaluating (3) with (1) and (2), and grouping some terms, it is obtained

$$F = \frac{T_{R1} + b_1/BLF + t}{T_{R1} + T_{R2} + (b_1 + b_2)/BLF + 2t} = \frac{BLF(T_{R1} + t) + 1.45b_2}{BLF(T_{R1} + T_{R2} + 2t) + 2.45b_2} \quad (4)$$

where $b_1=158$ bits, $b_2=109$ bits, $b_1=1.45b_2$, $t=2(T_1+T_2)$, $T_{R1}=T_{Qc}+T_{ACK}$, and $T_{R2}=T_{Read}+T_{Req}$.

The upper and lower bound of F are obtained by considering the slowest (lowest BLF) and fastest tag (highest BLF) configuration, respectively. The upper bound of F as BLF approaches 0 is

$$\lim_{BLF \rightarrow 0} F = \frac{1.45k_2}{2.45k_2} = 0.59. \quad (5)$$

And the lower bound of F as BLF approaches infinity is

$$\lim_{BLF \rightarrow \infty} F = \frac{T_{R1} + t}{T_{R1} + T_{R2} + 2t}. \quad (6)$$

The results obtained in (5) and (6) show that F is upper bounded to 59%, and the lower bound depends on the link-timing-parameters (T_1 and T_2), and on the reader application-specific configuration.

Next, (6) is evaluated for $BLF=640\text{kHz}$, the highest value allowed by EPC C1G2, considering two scenarios: i) slow reader, with $T_{ari}=25\mu\text{s}$, $T_1=77\mu\text{s}$, and $T_2=32\mu\text{s}$; and ii) fast reader, with $T_{ari}=6.25\mu\text{s}$, $T_1=16.8\mu\text{s}$, and $T_2=4.69\mu\text{s}$. For scenario i), the value obtained is $\lim_{BLF \rightarrow \infty} F = 32.4\%$, and for ii), it is obtained $\lim_{BLF \rightarrow \infty} F = 33.7\%$.

In summary, it can be concluded that the value of F is upper-limited to 59%, and its lower bound varies between 32.4% and 33.7%, depending on the reader configuration. It can be appreciated that the impact of one tag identification time over the total time required to read one sensor data from that same tag is very significant.

In an RFID scenario, there is typically more than one sensor tag inside the reader interrogation zone, and tags' $RN16$ messages often collide when responding to the reader commands. As a result, tags must re-transmit their $RN16$ in the same inventory round. In order to receive the data from a particular tag sensor, there is a very high probability that the reader will receive several $RN16$ messages from that tag. This will increase not only T_{id} but also $T_{sensing}$ and F .

In conclusion, the impact of the tags identification time over the total time required to read their sensor data is very significant. Therefore, anti-collision protocols are a key component of RFID sensor networks. The next section presents the physical implementation feasibility of the protocol used in the current standard, the Slot Counter, and a faster protocol, the FuzzyQ, in a real RFID scenario.

IV. IMPLEMENTATION OF SLOT COUNTER AND FUZZYQ FOR RFID SENSOR NETWORKS

The Slot Counter and FuzzyQ anti-collision protocols have been implemented in a Software Defined Radio-Ultra High Frequency RFID (SDR-UHF RFID) system, consisting of an SDR reader, and WISP 5.1 tags. The reader uses the software modules of the system presented in [12] (source, matched filter, gate, decoder, and reader), modified to communicate with WISP 5.1 tags. These software modules are built on GNU radio 3.7.11, and extended functionalities have been added to them, highlighting the differentiation of idle and collision/success tags responses, anti-collision capabilities, and the implementation of additional reader commands. Because this reader is software-defined, FuzzyQ and Slot Counter (and many other DFSA protocols based on EPC C1G2) can be implemented by writing user-level software in C++.

A. Slot Counter implementation

This work presents an SDR RFID reader capable of communicating with accelerometer sensors included in WISP tags (version 5.1), and it is composed of an USRP N210, an SBX daughterboard, and a Linux PC. A Gigabit switch is used to communicate the USRP with the Linux PC. The transmit and receive ports of the daughterboard are connected to two

circularly polarized patch antennas of 6dBi gain. The system is presented in Fig. 3.

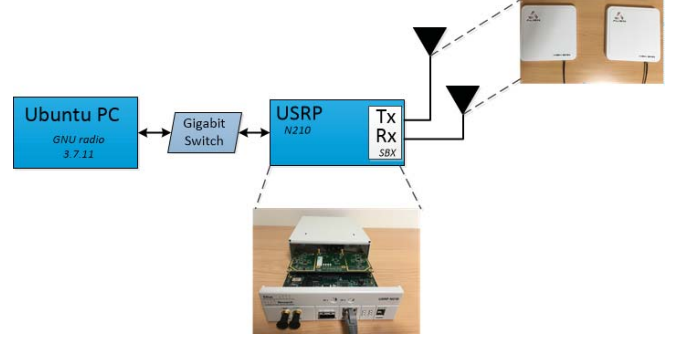


Fig. 3. A system description of the SDR-RFID reader

In order to differentiate between the three types of slots, a threshold energy value E_{th} is set for the energy of the received signal E_r to detect an idle response. Thus, an idle slot occurs when the energy of the received signal is lower than this threshold, that is, when $E_r < E_{th}$. This is shown in Fig. 4

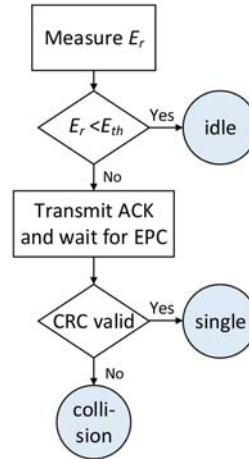


Fig. 4. Slot type differentiation in the proposed SDR-reader.

The Slot Counter protocol has been implemented using the value $C=0.2$. Regarding the tag-side implementation, WISP 5.1 tags have been used. They have been programmed to follow EPC C1G2 standard and to backscatter the information at 160kHz. Additionally, they have been re-programmed to i) transmit $RN16$ just at one slot per frame, and ii) stay idle until the reader starts a new inventory round once they are identified (transmitted their EPC).

B. FuzzyQ Implementation

FuzzyQ has been implemented in the reader presented in Fig. 3. This protocol presents a fast strategy to read RFID tags by not limiting the variation of Q to steps of one unit and considering instead a range of values, in order to achieve a more abrupt step size ($0, \pm 1, \pm 2$, or ± 3). In order to consider

these 7 different step size values, the reader QA command, as defined in the EPC C1G2 standard, needs to be modified.

According to EPC C1G2, the field $UpDn$ of QA command consists of 3 bits and it determines whether and how the tag adjusts Q , as shown in Table II. From this table, it is clear that Q can only be modified in steps of one unit.

TABLE II
QUERYADJUST COMMAND [7]

	Command	Session	UpDn
# of bits	4	2	3
Description	1001	00: S0 01: S1 10: S2 11: S3	110: $Q=Q+1$ 000: no change to Q 011: $Q=Q-1$

The protocol FuzzyQ introduces the parameter ΔQ which represents the variation of Q . This protocol takes advantage of the 3-bits-length of the field $UpDn$, and considers 7 different variations in Q : $Q=Q$, $Q=Q\pm 1$, $Q=Q\pm 2$, and $Q=Q\pm 3$. In order to consider all the possibilities, the modified QueryAdjust command is defined as mQA, by extending the values of the field $UpDn$. The specification of mQA is shown in Table III.

TABLE III
MODIFIED QUERYADJUST COMMAND

	Command	Session	UpDn
# of bits	4	2	3
Description	1001	00: S0 01: S1 10: S2 11: S3	000: no change to Q 001: $Q = Q-3$ 010: $Q = Q-2$ 011: $Q = Q-1$ 100: $Q = Q+3$ 101: $Q = Q+2$ 110: $Q = Q+1$

Regarding the tag-side, WISP 5.1 tags have been used to validate FuzzyQ in a real scenario under EPC C1G2 requirements. They have been programmed to backscatter the information at 160kHz, and their firmware has been updated to interpret mQA. Additionally, they have been re-programmed to i) transmit $RN16$ just at one slot per frame, and ii) stay idle until the reader starts a new inventory round once they are identified.

V. EXPERIMENTAL RESULTS

This section evaluates and compares the performance of Slot Counter and FuzzyQ anti-collision protocols running in the proposed SDR-UHF RFID system. This system consists of the reader shown in Fig. 3, and 4 WISP tags placed 1m away from the reader's antennas. The transmission model for the proposed RFID hardware, meeting EPC C1G2 requirements, is shown in Fig. 5. The duration of each type of slot, single, collision, and idle, is referred to as T_s , T_k , and T_i , respectively.

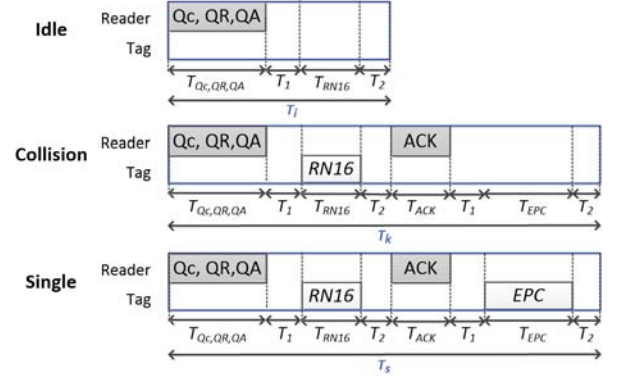


Fig. 5. Transmission model of the proposed SDR system

It is appreciated that a single and collision slot have the same duration. The duration of each type of slot is obtained as:

$$T_k = T_s = T_{Qc,QR,QA} + T_1 + T_{RN16} + T_2 + T_{ACK} + T_{EPC} + T_2, \quad (7)$$

and

$$T_i = T_{Qc,QR,QA} + T_1 + T_{RN16} + T_2. \quad (8)$$

The parameters of the implemented RFID system are presented in Table IV. First, the tag identification rate TIR is evaluated, and the experimental results are compared with simulated results for denser tag sets. Then, the parameter F is evaluated for both anti-collision protocols via simulation under different sizes of tag sets, from 4 to 1024 tags.

TABLE IV
PARAMETERS OF THE IMPLEMENTED RFID SYSTEM

Parameter	Value
BLF	160 kHz
T_{ari}	12.5 μs
$T_{delimiter}$	12.5 μs
RT_{cal}	36.6 μs
TR_{cal}	50 μs
T_1	56 μs
T_{2RN16}	253 μs
T_{2epc}	352 μs
Divide Ratio (DR)	8
Num. of subcarrier cycles per symbol (M)	1
Initial Q	1
Transmission frequency	915 MHz
Transmission power	180 mWatts

A. Analysis of T_2

Considering $BLF=160kHz$, the maximum value of T_2 allowed by EPC C1G2 is $T_{2max}=120\mu s$. After T_{2max} , a commercial tag ignores the next reader command. In the presented system, the reader needs an average of T_{2RN16} after the reception of $RN16$ to process the tag's response and to transmit ACK . Also, the reader needs an average T_{2epc} after the reception of EPC to process the tag's response and to

transmit the next command. The experiments performed in this work showed that $T_{EPC} > T_{max}$, and $T_{RN16} > T_{max}$. This occurs because the USRP and GNU radio require a higher processing time than a commercial reader. However, a WISP tag will ignore this T_{max} requirement imposed by EPC C1G2, and it will keep waiting for the next reader command after T_{max} . In conclusion, the values T_{RN16} and T_{epc} will be used in the protocols evaluation metrics.

B. Evaluation of the Tag Identification Rate

The performance of FuzzyQ is validated and compared with Slot Counter by measuring the Tag Identification Rate TIR metric, defined as the total number of tags identified per second. Experimental results have been obtained using 4 WISPs, following two different approaches:

- *analytical*: TIR is measured as the total number of tags identified divided by the total identification time. This time is calculated by measuring the number of transmitted reader commands (Qc , QA , QR , ACK , Req_{RN} , and $Read$), and the received tags' responses ($RN16$, EPC), and multiplying them by their corresponding duration, which the reader knows a priori.
- *timer*: TIR is measured as the total number of tags identified divided by the total identification time obtained with a software timer. For this purpose, a timer is started in the reader program when the reader transmits the first command Qc , and it is stopped when the reader correctly identifies the four tags present inside its interrogation zone. With this approach, the experimental results also provide information about the reader processing time.

Experimental results are shown in Fig. 6 and compared with simulation results for $n=4$. Clearly, FuzzyQ improves TIR in relation to Slot Counter. Additionally, experimental results obtained by calculating TIR with the timer are higher than the values calculated analytically, as expected, because the SDR system requires some time for the processing and calculations tasks. Analytical TIR is higher than simulation TIR for both protocols, because capture effect¹ contributes to a reduction in the number of collision slots c_k .

Next, both protocols have been compared by simulation for higher tag set sizes up to $n=1024$. Simulation results have been obtained with Matlab R2017a. In the simulation scenario, tags are assumed to be uniformly distributed and physical-layer effects are not considered. In order to get a fair comparison between the experimental and simulation results, the transmission model for the proposed SDR reader, shown in Fig. 5, has been used for the simulation scenario. The simulation responses have been averaged over 1000 iterations for accuracy in the results. The results are shown in Fig. 7, where it is appreciated a small increase in TIR with increasing n for the two protocols evaluated. Also, the improvement of

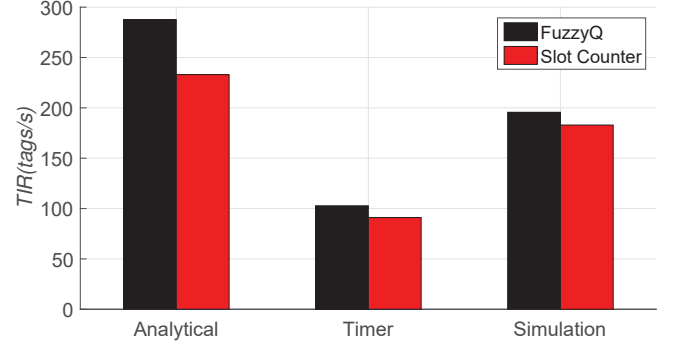


Fig. 6. Comparison of simulated and experimental (analytical and timer) TIR of FuzzyQ and Slot Counter for $n=4$.

FuzzyQ with respect to Slot Counter is maintained for n from 4 to 1024.

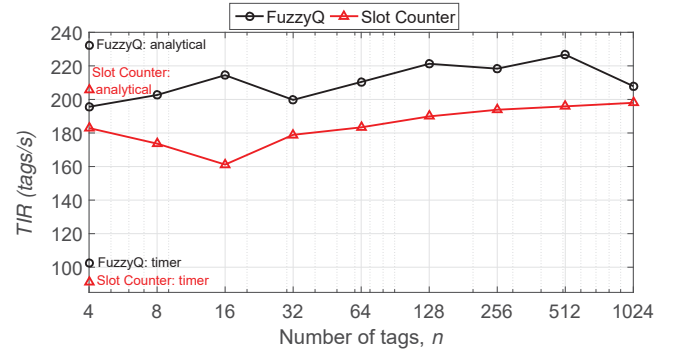


Fig. 7. Comparison of simulated TIR as a function of n for FuzzyQ and Slot Counter. Experimental results are included for $n=4$.

Next, the total number of collision and idle slots are evaluated for $n=4$. Experimental and simulation results are compared in Fig. 8. This figure shows the ability of FuzzyQ to reduce the number of collision slots by using mQA, at the expense of a higher number of idle slots. Additionally, this figure shows that the capture effect contributes to a reduction in experimental c_k with respect to the simulation c_k , for the two protocols evaluated.

Finally, this section compares and evaluates the total number of reader commands QA (mQA for FuzzyQ) and QR in one inventory round, referred to as tQA and tQR , respectively. Experimental and simulation results for $n=4$ are shown in Fig. 9. Clearly, FuzzyQ reduces the number of QR commands, at the expense of a higher number of mQA commands.

C. Evaluation of F

This section evaluates the parameter F (3) considering the RFID system presented in Section IV, using the parameters of Table IV.

1) *Single tag environment*: First, F is evaluated for the single tag environment, considering one WISP 5.1 tag with an accelerometer sensor incorporated. Evaluating (3), it is obtained $F=43.89\%$. This result shows that about half of

¹Physical layer effect which is very common in RFID systems. Tags response amplitudes can vary due to different distances from the reader antenna, or due to multi-path fading present in the channel [13]. As a result, the reader successfully resolves one tag reply in a collided slot.

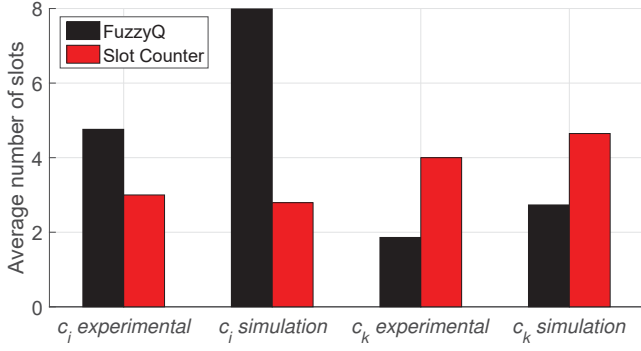


Fig. 8. Comparison of experimental and simulated c_i and c_k of FuzzyQ and Slot Counter for $n=4$.

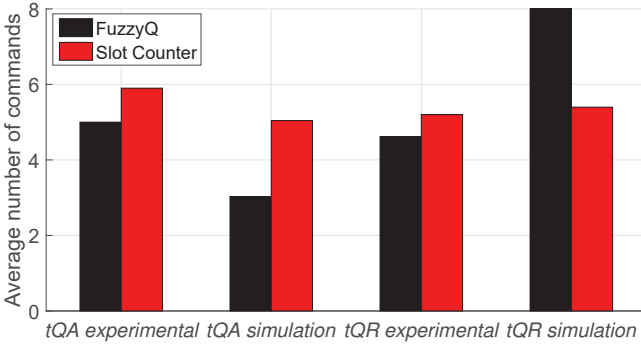


Fig. 9. Comparison of experimental and simulated tQA and tQR of FuzzyQ and Slot Counter for $n=4$.

the time required to read one accelerometer sensor data is employed in identifying the tag (receiving the EPC). Furthermore, this value falls between the lower bounds and upper bound of F obtained in Section III.

2) *Multiple-tag-environment*: Next, F is evaluated for the multiple tag environment. For this purpose, it is defined the factor F_n as total time required to identify n sensor tags (T_{id_n}) divided by the total time to read one sensor data from each one of the n sensors ($nT_{sensing}$)

$$F_n = \frac{T_{id_n}}{nT_{sensing}}. \quad (9)$$

The parameter T_{id_n} is obtained by simulation, by measuring the total time required by the protocol to identify n tags. The parameter $T_{sensing}$ is also obtained by simulation. Evaluation results are obtained with Matlab R2017a, defining a scenario where tags are assumed to be uniformly distributed. Simulation results are obtained for n from 4 to 1024 tags. The simulation responses have been averaged over 1000 iterations for accuracy in the results. Results are shown in Fig. 10.

This figure shows that FuzzyQ presents a lower F_n than Slot Counter, due to the ability of FuzzyQ to reduce the number of collision slots per tag with mQA command, as shown in Fig. 8. Finally, it is appreciated that most of the values of F_n in Fig. 10 overpass the upper limit of F (59%) because tags

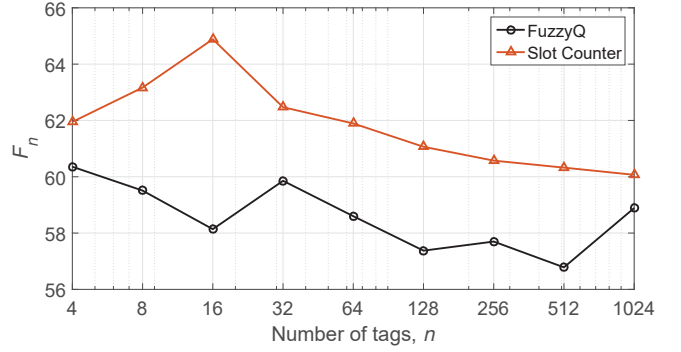


Fig. 10. Comparison of simulated F_n as a function of n for FuzzyQ and Slot Counter.

experience collisions in a multiple tag identification scenario, increasing the tags identification time.

D. Identified limitations and Future work

Given the capabilities of the hardware used, several problems have been found to fluently identify more than 4 WISP tags in the reader interrogation range. As a result, it was not possible to obtain a strong set of data streamed from bigger sets of WISP tags. The main reason for this behaviour was the low transmission power of the reader, which provided a weak transmission channel prone to detection errors and reading transmission errors. A possible solution to strengthen the communication among the reader and the WISP tags is to include a power amplifier before the transmission antenna in order to increase the antenna range. This modification will increase the power transmitted allowing the reader to interrogate a higher number of tags at a higher distance and it is proposed as future work.

VI. CONCLUSION

The experimentation of two anti-collision protocols, the Slot Counter and the FuzzyQ, has been performed and analyzed using an SDR-UHF RFID system. The aim of the experimentation was to compare two anti-collision protocols from a physical implementation perspective and to show the relevance of the identification phase of an RFID sensor network with respect to the sensing phase. For this last step, several simulations have been provided showing that in a multiple tag scenario of an RFID sensor network, the identification time represents more than 59% of the sensing time. In other words, the time required to identify one tag represents more than half of the total time to read one sensor data. To conclude, anti-collision protocols constitute an important and relevant part of the process of reading RFID sensors.

REFERENCES

- [1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. New York, NY, USA: John Wiley & Sons, Inc., 2 ed., 2003.
- [2] A. P. Sample, D. J. Yeager, P. S. Powlidge, A. V. Mamishev, and J. R. Smith, "Design of an RFID-Based Battery-Free Programmable Sensing Platform," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, pp. 2608–2615, Nov 2008.

- [3] (2016) UMich Moo. [Online]. Available: <http://spqr.cs.umass.edu/moo>.
- [4] H. Aantjes, A. Y. Majid, P. Pawełczak, J. Tan, A. Parks, and J. R. Smith, "Fast downstream to many (computational) RFIDs," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, May 2017.
- [5] D. Klair, C. Kwan-Wu, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," *Communications Surveys Tutorials, IEEE*, vol. 12, pp. 400–421, Third 2010.
- [6] H. Landaluze, A. Perallos, E. Onieva, L. Arjona, and L. Bengtsson, "An Energy and Identification Time Decreasing Procedure for Memoryless RFID Tag Anticollision Protocols," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 4234–4247, June 2016.
- [7] "Radio Frequency Identity Protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz–960 MHz," April 2015. Version 2.0.1.
- [8] C. Angerer, R. Langwieser, and M. Rupp, "RFID Reader Receivers for Physical Layer Collision Recovery," *IEEE Transactions on Communications*, vol. 58, pp. 3526–3537, December 2010.
- [9] Y. Chen and F. h. Zhang, "Study on Anti-collision Q Algorithm for UHF RFID," in *2010 International Conference on Communications and Mobile Computing*, vol. 3, pp. 168–170, April 2010.
- [10] L. Arjona, H. Landaluze, A. Perallos, and E. Onieva, "Fast fuzzy anti-collision protocol for the RFID standard EPC Gen-2," *IET Electronics Letters*, vol. 52, no. 8, pp. 663–665, 2016.
- [11] J. V. Sobral, J. J. Rodrigues, R. A. Rabelo, J. C. L. Filho, N. Sousa, H. S. Araujo, and R. H. Filho, "A framework for enhancing the performance of Internet of Things applications based on RFID and WSNs," *Journal of Network and Computer Applications*, vol. 107, pp. 56 – 68, 2018.
- [12] N. Kargas, F. Mavromatis, and A. Bletsas, "Fully-Coherent Reader With Commodity SDR for Gen2 FM0 and Computational RFID," *IEEE Wireless Communications Letters*, vol. 4, pp. 617–620, Dec 2015.
- [13] P. Solic, J. Maras, J. Radic, and Z. Blazevic, "Comparing Theoretical and Experimental Results in Gen2 RFID Throughput," *IEEE Transactions on Automation Science and Engineering*, vol. 14, pp. 349–357, Jan 2017.