Influence of Managing the Number of Tag Bits Transmitted on the Query Tree RFID Collision Resolution Protocol

Hugo Landaluce, Asier Perallos and Ignacio Angulo

Abstract—**Radio Frequency Identification (RFID) technology is increasingly becoming popular, for its widespread use and more sophisticated applications. The coexistence of tags sharing the communication channel requires solutions to message collisions, which degrade bandwidth, and increase the number of transmitted bits. A new methodology called 'window' is presented to manage the number of bits transmitted by a tag. The aim is show how the query tree (QT) protocol is influenced by this feature, and how the performance of the novel protocol, query window tree (QwT), improves when the tag ID distribution is correlated. Therefore, we have performed a fair comparison of the Query Tree and the new proposed QwT protocol for various tag ID distributions. Simulations show that the QwT positively decreases the total number of bits that are transmitted by tags.**

Index terms: **RFID; QT; anti-collision; window; tag-bits.**

I. Introduction

A radio frequency identification system (RFID) is an auto identification (autoID) method that can read codes that were previously stored in small transponders/tags wirelessly. This technology uses attached tags for monitoring and identifying objects in an omnidirectional fashion. If every object in the world is tagged, everything will be identified, creating tremendous benefits in a very different kind of applications like traceability of goods, baggage management, livestock tracking, and supply chain management [1,2].

Typically, an RFID system is given by [3]:

- One or more tags. These include an IC-chip and an antenna and are attached to the objects to count or identify. Tags can be active (battery operated) or passive (no battery). Because the passive tags activate using coupled power originated from the reader, the latter has a lower coverage.
- A reader / interrogator. This device is made up of an RF module, a control unit and one or more antennas. It offers a bidirectional communication between the tags and the reader.
- A data processing subsystem. Connected to the reader, allows for the storage and further processing of the data information of identified tags into a database.

RFID is increasingly being used as an autoID technology. Unlike barcodes, RFID does not require imminent handling, no line of sight is required between the reader and the object, and tags provide greater storage (64 bits, 96 bits and 496 bits). Since passive tags are cheaper than active tags, they are becoming more common in applications such as tracking, controlling and traceability. In addition, RFID is becoming a prominent technology in supply chain management and industrial automation applications, since it perfectly evolves into the paradigm of ubiquitous computing [4]. This fact defines RFID as a unique technology that allows ubiquitous identification. Mobile readers used in most of these applications are battery operated. Information transmitted between readers and tags is relevant to preserve battery-life and minimize power consumption, whether tags are active (battery-powered) or passive. If tags are active, batteries will need more frequent replacement. In contrast, if they are passive, reader's consumption will increase.

The coexistence of various tags sharing the communication channel leads to a unique problem known as *the tag collision problem*. When various tags send messages to a reader simultaneously, a cancellation of bits is produced and the resulting message is unreadable (collision). Collisions force the reader to retransmit tag IDs, which results in a loss of bandwidth, an increase of power consumption, and a large delay in the identification process. To face this problem an anti-collision protocol is needed. In literature, several proposed protocols have been reported, and they can be classified in Aloha based, tree based and hybrid protocols [5]. Aloha based protocols are considered probabilistic because tags use random numbers to respond [6,7,8]. These protocols suffer from *the tag starvation problem*, in which a tag may not be read in a reading cycle. Tree based protocols [9]are considered deterministic and provide simple tag designs as the Query Tree (QT) [10]. These protocols theoretically read all the tags in the interrogation zone on each cycle. Hybrid protocols [11,12] are designed to avoid the problems of the Aloha and tree based protocols at the expense of complex reader and tag designs.

The standardization of the tag IDs with the Electronic Product Code (EPC) have enabled an improvement of RFID applications allowing it to access global networks. To afford these needs, RFID is increasingly demanding larger tag IDs.

EPC Header (8 bits)	General Manager Number (28 bits)	Object Class (24 bits)	Serial Number (36 bits)

Fig. 1. General Identifier (GID-96) of a tag.

Manuscript received December 14, 2012; revised February 21, 2013.
Authors are with Deusto Institute of Technology (DeustoTech), University of Deusto, Spain (e-mail: {hlandaluce, perallos, ignacio.angulo}@deusto.es).

The most used ID length, *k*, is 96 bits, enough to cover all the companies and objects in the world. Protocols that based identification on the tag ID as tree based protocols are affected causing an increase of the transmitted bits and the energy consumption. In addition, standardized tags provide correlated IDs when they belong to the same company or the same warehouse. EPCglobal has developed a standard to organize tag EPCs all over the world [13]. An example of one of the EPC structures defined is shown in Fig. 1. We present a novel method, the Query window Tree (QwT), based on the control of transmitted bits by the tag. The amount of bits transmitted by a tag is what we call the window size, w_s. This methodology is applied to the QT protocol, so that the number of transmitted bits could be controlled. The proposed protocol can perform in the same manner as a bit by bit algorithm for a small w_s value. In contrast, it performs similarly to the QT for a large w_s value close to k. Using a constant w_s value, the algorithm decreases the tag bits transmitted. Therefore when the tag ID distribution is correlated, our algorithm outperforms the number of slots and the total bits transmitted by the QT.

Subsequently, the rest of the paper is organized as follows: Section 2 provides background information and related work on anti-collision protocols. Section 3 presents the Window methodology. In Section 4 the proposed QwT protocol is presented. In Section 5 the evaluation of the QwT protocol and a comparison with the QT protocol is shown. And Section 6 closures with the conclusions and prospect research.

II. Background and Related Work

In this section, a more detailed description of the existing anti-collision protocols is presented. Afterwards, some related work is analysed.

A. Background

Various multi-access procedures have been developed in order to separate physically the transmitters' signals [3]. They are classified into Space Division Multiple Access (SDMA), Frequency Division Multiple Access (FDMA), Code Division Multiple Access (CDMA) and Time Division Multiple Access (TDMA):

- *SDMA*. Using a controlled directional antenna on the reader, it can point the beam at different zones to be read. However, these techniques are expensive and require complex antenna designs.
- *FDMA*. Transmission channel is split up into different carrier frequencies that are simultaneously available. It requires a complex receiver at the reader.
- *CDMA*. Tag IDs are multiplied with a pseudo-random sequence before transmission. It demands elevated power consumption.
- *TDMA*. Transmission channel is divided between the participants chronologically.

In RFID systems, TDMA procedures are the most used techniques in RFID and they own the largest group of anti-collision methods. These can be categorized in: Aloha based protocols which are probabilistic, tree based protocols which are deterministic, and hybrid protocols which are a mixture of the previous ones [5].

A.1. Aloha based protocols

Aloha protocol is the origin of the Aloha based protocols. An improvement of that is the slotted-Aloha in which time is divided into slots improving its throughput [3]. Later, framed-slotted-Aloha (FSA) is developed. In FSA all nodes must respond choosing a slot into a fixed length frame (a group of slots). As the throughput of the FSA decreases with the increase of the total amount of nodes, a dynamic-framed-slotted-Aloha (DFSA) is developed [6,7]. This protocol changes the length of the frame dynamically using an estimator to adjust the frame size. Some protocols like I-Code [7] change the frame size at the end of the last frame slot, and other algorithms, as the EPC C1G2 Slot Counter [8], adjust the frame size after a slot transmission. Early cited, *the tag starvation problem* affects probabilistic algorithms, this is a tag that may not be correctly read during a reading cycle. Besides, estimation involves some disadvantages [12]: an increase in the computational cost of the reader [7] and the tag [14]; an error that degrades the efficiency; and lastly, an initial frame length cannot be set according to the estimated number of tags.

A.2. Tree based protocols

The main feature of this kind of protocols is that they are deterministic. This is that all tags in the reader's interrogation zone are going to be identified. These protocols usually have simple design tags and work well with uniform set of tags but are slower than Aloha based protocols. They can be categorized into [5]: Tree Splitting (TS), Query Tree (QT), Binary Search (BS) and Bitwise Arbitration (BTA).

A virtual tree to organize and identify each tag was firstly proposed by the authors of the TS in [9]. This algorithm splits the set of tags in B subsets (B > 1) after a collision. These subsets become increasingly smaller until they contain one tag. The TS does not need clocking circuitry but they must maintain a counter, so if a tag get discharged, it loses cycle information. Moreover, the QT is proposed in [10]. The reader of the QT sends queries and tags, whose ID match that query, respond the reader, Fig. 2. After a collision, the reader increases the query with 1 or 0, obtaining two new queries, and sending them repeatedly upon the successful response of all the tags. The process needs to go through all the possible queries to detect all the tags. QT is called memoryless because tags do not require any counter or memory. Additionally, the BS is another tree based protocol [15]. Tags compare their ID with a serial number sent by the reader. If the tag ID is equal to or lower than the serial number, the tag transmits its ID. Once the response is received at the reader, it decreases the serial

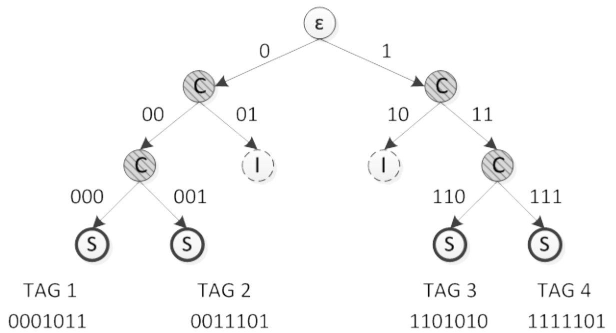


Fig. 2. Example of the QT protocol, where C is a collided node, I is an idle node, and S is a successful node.

number in case of a collision, or identifies the tag in case of a unique response. Lastly, the BTA protocols operate requesting tags to respond bit by bit. Tag responses must be synchronized in these protocols, so that identical responses could result in no collision. The Stack-based ID-Binary Tree algorithm (SIBT) [16] or de Bit Query (BQ) [17] use queries to cover a binary tree which height is the maximum tag ID.

A.3. Hybrid protocols

Tree and Aloha based protocols are combined in hybrid protocols to avoid their problems. There are mainly two kinds of this combination. One is using randomized divisions in tree-based algorithms, and another is using tree strategies after a collision in Aloha-based algorithms. The first kind of protocols such as Tree Slotted Aloha (TSA) [11] use a tree structure. Tag responses are sequenced in slots as in a FSA, and new frames are applied on collided tags. These kind of hybrid protocols require complex tags and carry the same problems as Aloha based protocols, like the tag starvation. In contrast, in the second proposed protocols such as the Binary Tree Slotted Aloha (BTSA) [12], tags choose a slot randomly after a reader command. In case of a collision, a tree based protocol is employed to identify tags. This variation of the hybrid protocols requires an initial estimation of the frame that determines the performance of the protocol.

B. Related Work

The probabilistic nature of Aloha based protocols causes the tag collision problem and the estimation error early mentioned. In contrast, Tree based protocols identify all the tags in the interrogation zone and do not need to estimate the number of tags. However, the number of transmitted bits is higher than the bits transmitted in Aloha based protocols; the reason why we have focused on decreasing the number of bits in Tree based protocols. Specifically we have focused on Tree based protocols that use queries to identify the tags.

In the literature review there are two protocols based on queries: the QT protocol [10] and the SIBT protocol [16]. Each of them represents a strategy of tag identification:

- *Large number of bits per slot*: In the QT protocol, the reader sends a query and tags, whose ID prefix match that query, respond their full ID. The main advantage of this strategy is the complete identification of a tag in one slot. However, the waste of time and energy in a collision slot is highly remarkable.

- *Small number of bits per slot*: The reader sends a query and a bit position. Once tags receive the command, they respond their next bit position. This procedure is repeated until the tag sends its last ID bit. Therefore this strategy persuades a better use of each ID bit. Although a in a collided slot two ID bits are identified simultaneously, long ID tags cause an increase in power consumption.

Focused on reducing the number of transmitted bits are [10] and [18]. In [10] an improvement of the QT is proposed using two types of queries. The reader sends a short query in order to receive 1 bit response from the tag. Otherwise, it sends a long query when it knows that only one tag will match the prefix expecting to receive the full ID of the tag. In [18] each tag generates a *k* bit random number prefix that is used to respond the reader instead of sending the full ID. When a tag matches its generated prefix with the one sent by the reader it responds its ID. If more than one tag chooses the same prefix, they respond their ID and a collision occurs. Tags increase their prefix with a new random bit and wait for a new reader query. This method is not very efficient because after each identified prefix the tag should send its full ID which is a waste of resource.

There are some surveys on how the QT protocol can be improved to handle tag IDs which could have some common prefixes [19,20,21]. The work in [19] takes advantage of the statistical information or other features and improves the QT protocol in terms of slots needed. An estimation of the number of tags and a complex hardware is required. The authors of [20] present a study of how the QT protocol can be improved to handle correlated prefixes in tag IDs. It saves the most used prefixes and uses them in subsequent read cycles. Therefore this method needs various read cycles to show improvements. Authors in [21] propose an algorithm that tries to exploit the GID-96 structure of tag IDs. It starts the identification over the LSB instead of the MSB since the main differences in IDs will be in the right part of it. The reader uses the query tree to generate prefixes that should match the right part of the tag ID and when a tag matches its prefix it send the rest of the ID. If a collision occurs sending the full ID, the reader extends the prefix sent. This algorithm only works well if the tag IDs vary on their right part whereas the proposed method in this paper exploits the common parts of the tag IDs on any part of the full ID.

III. WINDOW METHODOLOGY

In most of the tree based protocols, tags respond their full ID when the query sent by the reader matches the tag ID prefix. In a reading cycle there are lots of tag responses that end up in collision and on each of those collision slots the whole ID bits are wasted. Tree based collision resolution protocols are very sensitive to tag ID lengths or how the tag IDs are distributed.

We propose a methodology in order to restrict the bits sent by the tag. A constant amount of bits are established and that is what we call 'window'. Tags supposed to respond will send synchronously the amount of bits specified by the window, w_s, instead of their full ID, as it is depicted in Fig. 3. Fading problems are not considered in this work. This methodology considerably decreases the amount of bits transmitted by a tag to be identified. It also contributes to transform possible collisions into partial successes and decrease the number of idle slots. However, the reader must interrogate tags until they send their last part of their ID. Four situations can occur on a tag response:

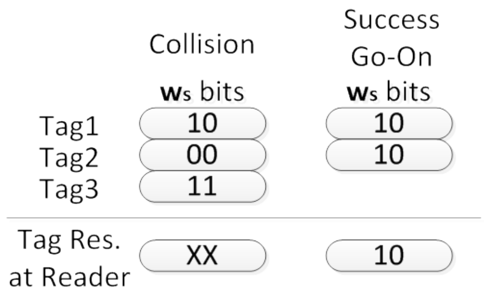


Fig. 3. Synchronized window answers.

- *Idle slot*: when there is no response upon the reader's request, an idle slot is performed. The reader rejects that request and continues sending the next query.
- *Collision slot*: when various tags respond and the windows responded are different, a collision occurs, Fig. 3. The reader is unable to understand tag responses and it creates new requests following the protocol guidelines.
- *Go-On slot*: when at least one tag responds its window bits and the reader is able to understand them, Fig. 3. If the ID is not completed, it is assumed a go-On slot. The reader stores the window bits received and uses them to send the next request for the tags.
- *Success slot*: it is a go-On slot where the window bit received is the end of the tag ID, Fig. 3. Afterwards, the reader identifies the tag and stores it in its memory. Then it continues with the next request.

The main features of this procedure are:

- Decline the number of unnecessary bits. Low window values will contribute to waste less bits per slot in case of collision. This feature will cause the increase in the number of slots.
- Take advantage of the similarity of the distribution of tag IDs. When various tags respond the same information at the same time, the reader receives it as a unique response and completes a new query with the received bits, w_s.
- Higher window values causes more wasted bits on each collision, but gives a faster performance to the algorithm.
- Idle and collision slots are reduced, but a new type of slot is produced. Go-On slots are partial success slots that are used to complete the tag ID so that the reader could differentiate tags.

IV. QUERY WINDOW TREE PROTOCOL (QWT)

The proposed Query window Tree (QwT) protocol is a QT based protocol that has adopted the window methodology. It is

Fig. 4. Structure of a tag ID.

also a memoryless protocol since tags do not need to store information to be identified. The main contribution of the window methodology to the QT protocol is the decrease in the number of tag bits transmitted. That will preserve active tags battery-life, and will decrease passive tags complexity. Besides, as they are powered by the reader, the amount of energy spent on powering the tags will decrease. The variations made by the window to the QT protocol are compatible with most of the proposed protocols in the literature improving not only the transmitted tag bits but the feature proposed by each modification too.

As it is shown in Fig. 4, the proposed QwT protocol sends a query of q bits to all the tags in the interrogation zone. Tags respond if their ID prefix (Query in Fig. 4) matches the query sent by the reader. If there is a successful match, the tag responds the next adjacent bits (w_s) of the ID. An example of identification of 6 tags using QT and QwT is shown in Fig. 5. The w_s used is 2 bits in QwT. And the k is assumed 8 bits. The reader starts with a query *0*. Tag 1,2,3, and 4 respond and a collision occurs. Two new queries are created adding a *0* and a *1* to the query sent (0*0*, 0*1*). The reader sends the new query (0*0*) and a collision occurs again. The same procedure is followed and two new queries are created (00*0*, 00*1*). After sending the new query (00*0*), only tag 1 responds. Were it to respond using the QT protocol, tag 1 would respond the full ID and the tag would be fully identified. However, tag 1 only responds the window bit in the QwT protocol, and the tag, therefore, is not fully identified. A new query is performed adding the received bits to the query sent previously (000*10*). It sends it again and another go-On slot occurs repeating the same procedure. After this last query, tag 1 is fully identified. The procedure followed to identify tag 2 is similar to tag 1. Afterwards, the QwT protocol goes back to the last known collision and chooses the next query (0*1*). In this case, Tag 3

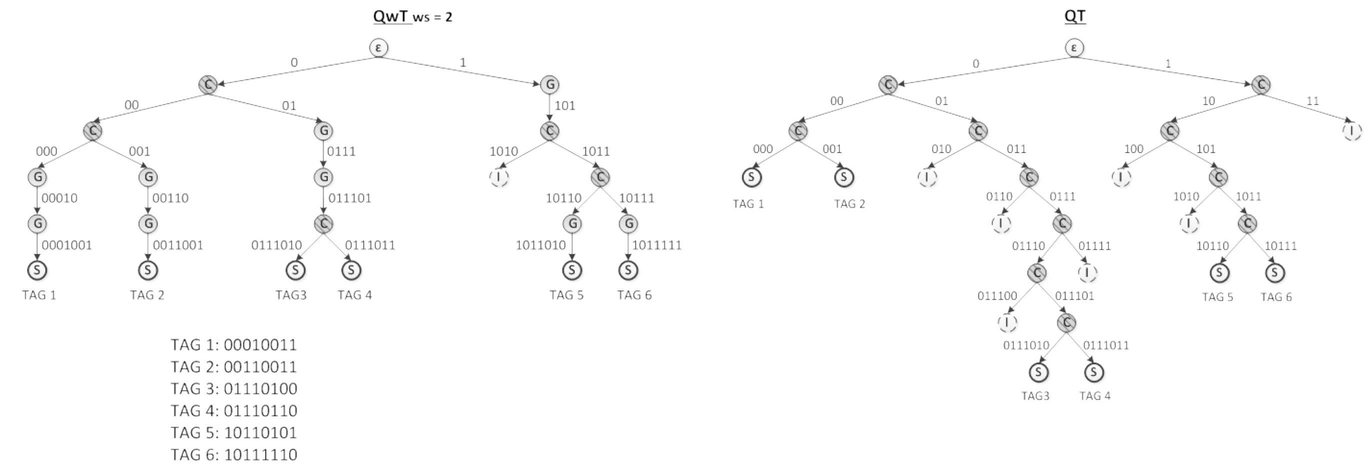
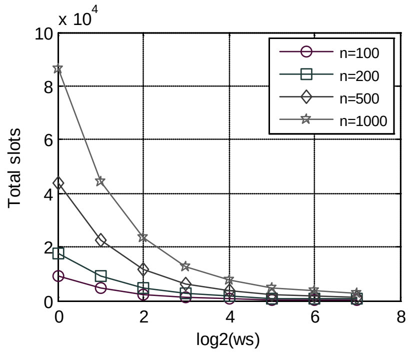
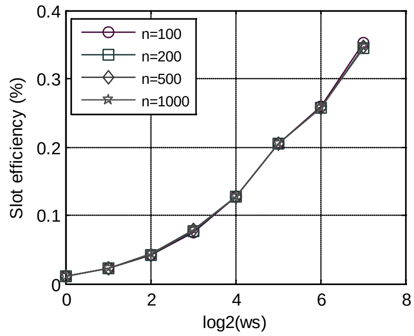


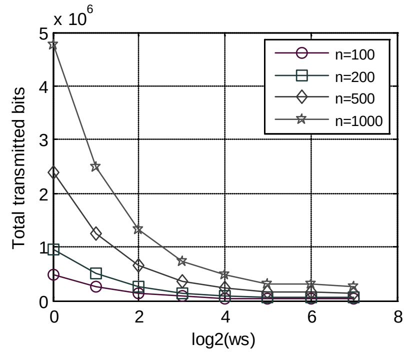
Fig. 5. Example of the proposed QwT with a w_s = 2 and the QT protocol. *C* represents a collision slot, *I* an idle slot, *S* a succesful slot, and *G* a go-On slot.



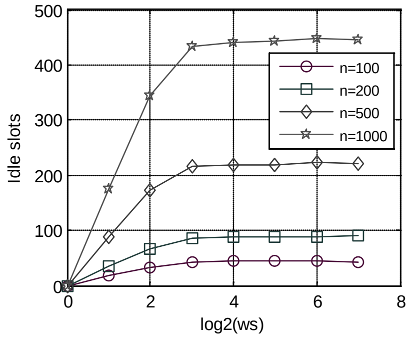
(a) Total number of slots used



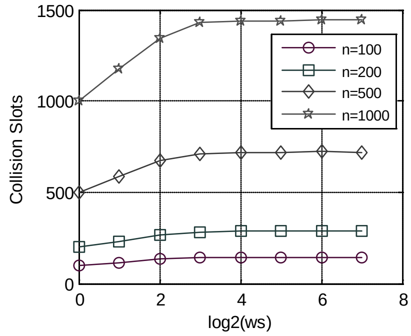
(b) Reader's efficiency



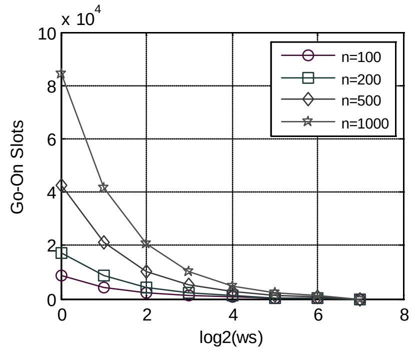
(c) Total bits send by reader and tags



(d) Idle slots



(e) Collision slots



(f) Go-On slots

Fig. 6. Influence of the window in the QwT protocol.

and 4 respond, and as both tags provide the same window bit value, the reader understands the response. That is added to the old query (01*11*) and the new query produced is sent again. Another go-On slot is performed (0111*01*) and after extending the query a collision occurs. The next 2 queries (011101*0*, 011101*1*) perform the identification of both tags 3 and 4. Unlike the QT protocol, the QwT does not use any idle slot. Later, the other side of the tree is requested using fewer collisions and idle slots than the QT.

TABLE I. 1
PSEUDO-CODE OF QWT

Reader procedure	**Tag procedure**

QwT(0);
QwT(1);
 	
QwT(char [] *query*);
k = *ID*.length;
q = *query*.length;
 	
broadcast(*query*)
winMatch = receiveResponse();
 	
if *matchedTags* > 1
nColls += 1;
query0 = query + 0;
QwT(*query0*);
query1 = *query* + 1;
QwT(*query1*);
elseif *matchedTags* == 1
ws = *winMatch*.length;
if *q* + *ws* < *k*
nGoOns += 1
query += *winMatch*;
QwT(*query*);
else
nSucc +=1;
end
elseif *matchedTags* == 0
nIdles += 1;
end
	receive(*query*);
	q = *query*.length;
	k = *ID*.length;
	ws = constantValue;
	
	if *query* == *ID*[0:*q*-1]
	if *q* + *ws* > *k*
	ws = *k* - *q*;
	end
	backscatter(*ID*[*q*:*ws*-1])
	end

Although the branch of the tag 1 and 2 needs more slots, other branches of the tree (tags 3, 4 ,5 and 6) use less slots and bits than the QT protocol. A comparison table between the QwT and the QT is also shown for this set of tags in Fig. 5. The performance of the QwT protocol is more efficient than the QT with this set of tags due to the common parts of the IDs. These common parts cause the proposed QwT, to aggressively advance through identification. QwT performs faster and more efficient than the QT. The pseudo-codes of the reader and the tag of the QwT protocols are shown in Table 1. Reader procedure shows a recursive function that needs the query as a string parameter. And the tag procedure shows the backscattering of the number of bits specified by w_s as a constant value. Table 1 also shows a limitation if the number of query bits plus w_s bits is bigger than k, resizing w_s.

V. SIMULATIONS

This section presents the evaluation of the simulation results of the proposed QwT protocol using Matlab R2012b. Simulation proposed defines a scenario with one reader and a varying number of tags, n. The tags are uniformly distributed and k is assumed 96 bits. The simulated responses were averaged over 100 iterations for accuracy in the results.

Fig. 6 shows how the QwT is influenced by the variation of w_s. The total number of slots performed by the QwT protocol and the slots efficiency; the number of idle, collision, and go-On slots; and the total number of bits transmitted between the reader and the tags, are depicted varying w_s under certain n

values. Simulated results in Fig. 6.a presents the decrease of the total number of slots used in the identification process with the increase of w_s. If a single tag matches a query sent by the reader, it will send the number of bits specified by w_s. Therefore the larger the w_s is, the less slots that are required to obtain the full tag ID. Also, the slots efficiency in Fig. 6.b, increases with the increase of w_s. For a high w_s value fewer slots are used to identify each tag than for a low w_s value. In Fig. 6.c the total number of bits transmitted between the reader and the tags are shown. It is calculated as the number of bits sent by the reader in a slot plus the number of bits received from the tags at the reader. These number of bits decrease with the increase of w_s. Although a small w_s decreases the number of bits transmitted by a tag, the improvement obtained is overwhelmed by the number of bits sent by the reader. A small w_s demands long queries to identify the full ID. Therefore, the number of bits transmitted by the reader increase, which increase the total number of bits transmitted too. In contrast, the use of small w_s values decreases the number of collision and idle slots. In Fig. 6.d and e and in Fig. 7, the number of collision and idle slots are shown for a homogeneous tag ID distribution. Not only does a small w_s reduces collisions, Fig 7.a, but also idles, Fig. 7.b. The new generated queries are highly likely to match at least two tags, and also to avoid inexistent queries. The number of collision and idle slots increase with the increase of w_s. And it should be noted that when w_s is 1 there are no idle slots, Fig. 7.b. Go-On slots are critical to finish the identification cycle as soon as possible. Graphics in Fig 6.f show that the smaller the w_s, the larger the number of go-On slots that are needed to cover the full ID of the tag. Although a large w_s provides few go-On slots, a lot of bits are wasted on previous collisions.

Summing up, small w_s values provide a great reduction of collision and idle slots, which reduces the number of slots used in the identification cycle. Also, the number of bits transmitted by a tag are nearer the optimal value, k. In contrast, a great number of go-On slots are required to accomplish the identification increasing the total number of bits. On the other hand, high w_s values cause a higher waste of bits transmitted by a tag, which increases the number of bits transmitted per tag. On the contrary fewer slots are required to identify the set of tags.

A. *Comparison of QwT schemes with the QT protocol for different ID distributions*

Early mentioned in section I, the standardization of RFID results in heterogeneous tag ID distributions. In Fig. 1, a GID-96 is shown but it is not the only standard to organize information in the tag ID. Fields contained in the ID cause correlated distributions of tags IDs, since at least the first fields are assigned by EPCglobal. This fact forces the reader to have to descend the binary tree until it reaches the end of the common ID to begin distinguishing tag responses. Window methodology enables the protocol to aggressively advance through the common parts of the IDs. That causes a decrease in the number of slots and bits used in the identification process. Three tag ID distributions are considered in the simulations:

- *Homogeneous distribution or 100% variable ID*: the likelihood of obtaining ‘0’ or ‘1’ when generating the tag ID of this distribution is the same.

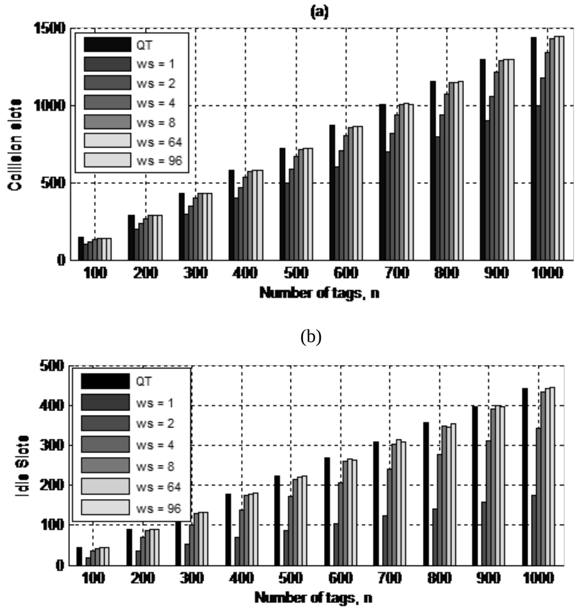


Fig. 7. (a) Collision slots and (b) idle slots in QT and QwT with an homogeneous tag ID distribution.

- *50% variable ID*: in this distribution, 50% of the tag ID is randomly generated at the beginning of an iteration, and fixed for all the set of tags. The rest of the ID is randomly generated for each tag.

- *10% variable ID*: in this case, 90% of the tag ID is randomly generated at the beginning of the iteration, and fixed for all the set of tags while the 10% remaining part of the ID is randomly generated for

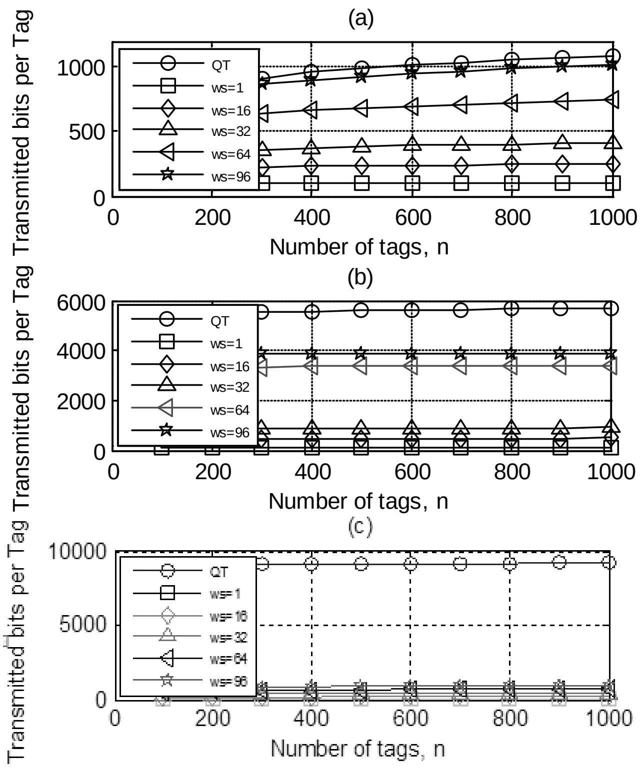
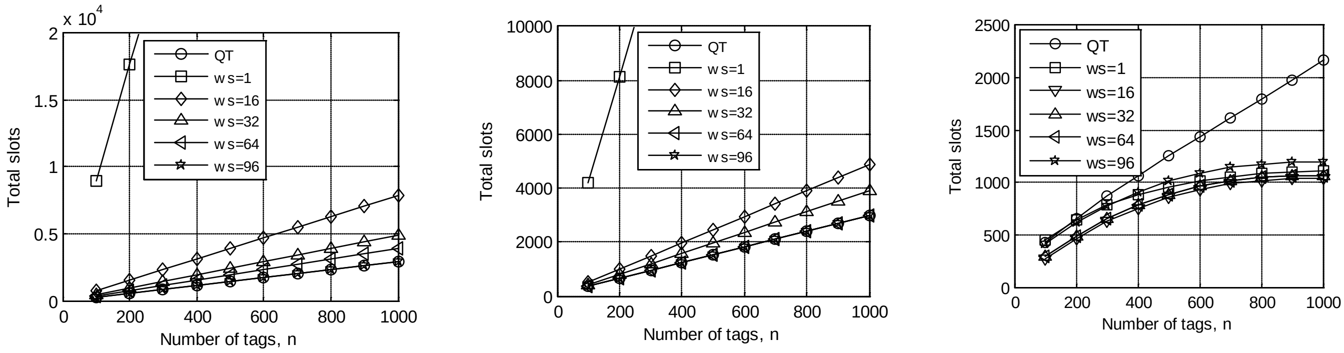
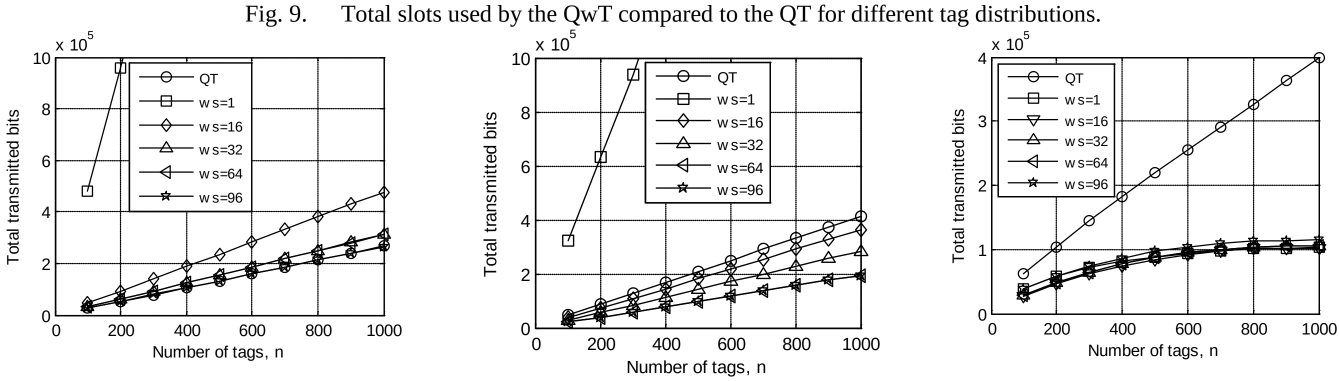


Fig. 8. Transmitted bits per tag. (a) ID 100% variable, (b) ID 50% variable, and (c) ID 10% variable.



(a) Homogeneous distrib. (100% ID var.) (b) 50% ID variable (c) 10% ID variable

Fig. 9. Total slots used by the QwT compared to the QT for different tag distributions.



(a) Homogeneous distrib. (100% ID var.) (b) 50% ID variable (c) 10% ID variable

Fig. 10. Total bits used by the QwT compared to the QT for different tag distributions.

each tag.

Again, the simulated responses were averaged over 100 iterations. In Fig. 9 and Fig. 10, simulation results of the performance of the proposed QwT and the QT protocols are shown for these tag ID distributions. The total number of slots and the total number of bits transmitted are shown varying n, under certain w_s values. Note that in the homogeneous distribution, Fig 9.a, the larger the w_s the more similar to the QT is the performance of the QwT. The largest w_s uses the least amount of slots as the QT does so. The lowest w_s presents the highest number of slots used and also the highest number of total bits used, Fig. 10.a.

Besides, Fig.8 shows promising results in number of transmitted bits per tag, where low w_s values provide transmissions of very few tag bits. In Fig. 8.a for a homogeneous distribution of IDs, low w_s values outperform the QT protocol in transmitted bits per tag. Low values provide optimal transmissions, where few bits are wasted. The worst case is for the biggest w_s value, which shows a similar performance to the QT protocol. Furthermore, the more correlated the distribution is, the bigger the outperform between the proposed QwT and the QT in transmitted bits per tag. Fig. 8 shows that the proposed QwT tags transmit less bits, which will decrease the speed and the power consumption of the protocol.

Results shown in Fig. 9.b, have been simulated with the second proposed distribution with the 50% of the ID variable. That shows that the number of slots used by the QwT has been drastically reduced for all w_s values in reference to the previous distribution in Fig. 9.a. The performance of the QwT protocol is improved due to the correlation of the tag IDs. The real improvement comes in terms of total bits transmitted between tags and the reader. It is shown in Fig 10.b that when w_s is bigger than 16, the proposed QwT performs the identification using less bits than the QT. And also in Fig 8.b, the number of bits used per tag in the QwT protocol is considerably lower than the number of bits per tag in the QT. This fact provides an improvement of the total number of transmitted bits in the QwT, which makes it more energy aware.

Finally, the last proposed distribution with only 10% of the ID variable presents results in Fig 9.c and 10.c. These are even better than the results of the second proposed distribution. QwT outperforms the QT in terms of slots used and slots efficiency for all w_s values. Also in Fig. 8.c the number of bits transmitted by a tag is lower than in other tag ID distributions. Thus, it can be concluded that the QwT protocol works better when there are common part IDs in the set of tags.

B. Selection of w_s

At this point, a proper value of w_s can be selected to face the identification process of a set of tags with an unknown tag ID distribution. Thanks to the results obtained, it is known that the more correlated the tag IDs are the better the performance of the QwT is in terms of slots and total bits transmitted. For the homogeneous distribution of the tag IDs, the best result is for w_s = 96. Using this w_s value the best performance is obtained and it is similar to the performance of the QT. However, the use of that value for another not uniformly tag ID distribution provides no improvements over the QT. Therefore, a smaller window is preferred.

The improvements in terms of bits transmitted are very evident when tag IDs are partially correlated. The smallest w_s obtain the best performance in slots, transmitted bits per tag, and total bits transmitted. However, the results shown in an homogeneous tag ID distribution are not very promising. For the reasons previously mentioned, medium w_s are preferred to obtain a better performance of the algorithm. We have chosen a medium value like $w_s = 64$, which outperforms the QT in terms of transmitted bits by a tag whatever the tag ID distribution is. Results for correlated distributions are better than the QT, especially in terms of total transmitted bits. Besides, the perfornabce of the algorithm present quite good results in an homogeneous tag ID distribution.

VI. Conclusions

A new methodology has been presented in this paper. The window methodology controls the number of bits transmitted by a tag. It has been applied to the most popular tree based protocol, the QT, and has been compared under certain conditions. The resulting algorithm keeps the memoryless feature of the QT. An analytical framework has been designed to compare the performance of both algorithms. Results obtained show that the bigger the w_s, the more similar to the QT is the proposed QwT algorithm. However, the number of bits transmitted per tag is reduced for all values of w_s. Three tag ID distributions have been proposed too, varying the amount of constant bits in the tag IDs. Simulation results show that the window feature exploits the correlated parts of the IDs, decreasing the number of slots and total transmitted bits.

A. Future Work

This work has been released to obtain some conclusions of the performance of a tree based protocol with batched responses of tags. A new anti-collision protocol based on the QT and the window methodology is going to be designed. Bearing in mind the conclusions obtained, a new QwT with dynamic window is expected to be designed. The window dynamic methodology will try to decrease the number of collisions and idle slots. But the main purpose of this feature will be the decrease in the total number of wasted bits transmitted between the reader and the tag, which will decrease the total number of transmitted bits between the reader and tags and the energy consumed by the RFID system. Moreover, the algorithm will exploit correlated sets of tags decreasing the number of slots and improving the efficiency.

References

[1] S.F. Wamba and A. Wicks, "RFID deployment and use in the dairy value chain: Applications, current issues and future research directions," in IEEE International Symposium on Technology and Society (ISTAS), Wollongong, NSW, 2010 , pp. 172 - 179.

[2] A. Moreno, I. Angulo, A. Perallos, H. Landaluce, I.L.García, L. Azpilicueta, J.J. Astrain, F. Falcone, J. Villadangos, "IVAN: Intelligent Van for the Distribution of Pharmaceutical Drugs," Sensors, vol. 12, no. 5, pp. 6587-6609, May 2012.

[3] K. Finkenzeller, RFID Handbook: fundamentals and applications in contactless smart cards identification., 2003.

[4] V. Stanford, "Pervasive Computing Goes the Last Hundred Feet with RFID Systems," IEEE Pervasive Computing, vol. 2, no. 2, pp. 9-14, April- June 2003.

[5] D.K. Klair, K.W. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," IEEE Communications Surveys & Tutorials, vol. 12, no. 3, pp. 400-421, August 2010.

[6] F. Schoute, "Dynamic Frame Length ALOHA," in IEEE Transactions on communications, vol. COM-31, NO 4, 1983, pp. 565-568.

[7] H. Vogt, "Efficient object identificacition with passive RFID tags," in Proceedings International Conference on Pervasive Computing, Zurich, Switzerland, April, 2002, pp. 98-113, vol. 2414.

[8] EPC Global, EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz. Version 1.2.0, Octubre 2008.

[9] D. R. Hush and C. Wood, "IEEE International Symposium on Information Theory," in IEEE Symposium on Information Theory, Cambridge, MA, USA, August 1998.

[10] C. Law, K. Lee, and K. Y. Siu, "Efficient memory-less protocol for tag identification," in Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Boston, Massachussetts, USA, August, 2000, pp. 75-84.

[11] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Tree Slotted Aloha: a New Protocol for Tag Identification in RFID Networks," in International Symposium on a World of Wireless, Mobile and Multimedia Networks, Buffalo-Niagara Falls, NY, 2006, pp. 1-6.

[12] H. Wu, Y. Zeng, J. Feng, and Y. Gu, "Binary Tree Slotted ALOHA for Passive RFID Tag Anti-Collision," IEEE Transactions on Parallel and Distributed Systems, vol. PP, no. 99, p. 1, April 2012.

[13] GS1 and EPCglobal, "GS1 EPC Tag Data Standard 1.6," GS1, EPCglobal, Sept. 2011.

[14] C. Qian, Y. Liu, H. Ngan, and L. Ni, "ASAP: Scalable Collision Arbitration for Large RFID Systems," IEEE Transactions on Parallel and Distributed Systems, vol. PP, no. 99, p. 1, February 2012.

[15] H. Vogt, "Efficient object identificacition with passive RFID tags," in Proceedings International Conference on Pervasive Computing, April, 2002, pp. 98-113.

[16] F. Bo, L. Jin-Tao, G. Jun-Bo, and D. Zhen-Hua, "ID-Binary Tree Stack Anticollision Algorithm for RFID," in IEEE Symposium on Computers and Communications, June 2006, pp. 207 - 212.

[17] S. H. Kim and P. G. Park, "An Efficient Tree-Based Tag Anti-Collision Protocol for RFID Systems," IEEE Commun. Letters, vol. 11, pp. 449-451, 2007.

[18] K.W. Chiang, C. Hua, and T.P. Yum, "Prefix-Randomized Query-Tree Protocol for RFID Systems," in IEEE International Conference on Communications (ICC '06), Istanbul, Turkey, June 2006, pp. 1653-1657.

[19] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Exploiting id knowledge for tag identification in rfid networks," in Proceedings of the 4th ACM workshop on Performance evaluation of wireless ad hoc, sensor,and ubiquitous networks (PE-WASUN '07), Chania, Greece, 2007, pp. 70-77.

[20] N. Bhandari, A. Sahoo, and S. Iryer, "Intelligent Query Tree

(IQT) Protocol to Improve RFID Tag Read Efficiency," in 9th International Conference on Information Technology (ICIT'06), Bhubaneswar, India, Dec. 2006, pp. 46-51.

[21] Woon-Young Yeo and Gyung-Ho Hwang, "Efficient anti-collision algorithm using variale length ID in RFID systems," IEICE Electronics Express, vol. 7, no. 23, pp. 1735-1740, December 2010.

Hugo Landaluce is a Ph.D candidate in the Mobility Research team in the University of Deusto, Spain. He received his bachelor's degree in Industrial Control and Electronics Engineering from the University of Deusto in 2007 and his master's degree in Advanced Electronic Systems from the University of the Basque Country, Spain in 2012. He was with Miesa Ingeniería S.L. from 2007 to 2011 working on the implementation of industrial applications. His main research interests include RFID technology, anti-collision protocols, algorithm analysis and optimization.

Asier Perallos BSc in Computer Engineering and Ph.D from the University of Deusto. MSc in Software Engineering. He has over 10 years of experience as a lecturer in the Computer Engineering Department of the Faculty of Engineering of the University of Deusto. His academic background has been focused on teaching software engineering, distributed systems and web quality evaluation. Director of Master's in Development and Integration of Software Solutions at the University of Deusto. Principal Researcher of DeustoTech Mobility research team in Deusto Foundation. More than a decade of experience developing and managing R&D projects, with dozens of projects and technology transfer actions led.

Ignacio Angulo Graduated in Computer Engineering at the University of Deusto in 1997. Founding partner of the company "Ingeniería de Microsistemas Programados SL" working on the design and manufacture of development systems based on microprocessors. Since October 2002 is working at the University of Deusto, where he teaches courses integrated in the Department of Industrial Technologies focused on the development of embedded systems. He is currently studying PhD inside the Remote Laboratories Research Line working to facilitate the deployment and integration of remote experiments in educational centers. He has participated in projects in the area of ITS and remote control and tele-maintenance. He has also collaborated in writing 6 books focusing on the design of microcontroller-based systems published by McGraw Hill or Thomson Learning Publishing. In March 2008 he joined Deusto Intitute of Technology within the Mobility research team.