Hardware based Analysis of RFID Anti-Collision Protocols based on the Standard EPCglobal Class-1 Generation-2

Laura Arjona, Hugo Landaluce, Asier Perallos, Nikola Cmiljanic
Deusto Institute of Technology (DeustoTech)
University of Deusto
Bilbao, Spain
Email: {laura.arjona,hlandaluce,perallos,n.cmiljanic}@deusto.es

***Abstract*—Radio Frequency Identification (RFID) technology is undergoing a remarkable development in the last few years. In this technology, identification information is exchanged between two devices: readers and tags. If two tags attempt to transmit simultaneously, a collision is produced. This phenomena, known as the tag collision problem, is becoming increasingly important, since it leads to an increase in the number of reader transmitted bits and identification delay, in addition to a wastage of energy and bandwidth. In this context, protocols based on the EPCglobal Class-1 Generation-2 (EPC C1G2) standard arbitrate collisions by adjusting the transmission frame size. The standard presents an uncertainty in the selection of the frame size, since it does not specify the exact value. This has led to many different alternatives. This paper presents a hardware analysis of the most relevant anti-collision protocols which deal with this uncertainty. The focus of this analysis is to design and evaluate with VHDL the tag's chip design in order to extract the number clock cycles a tag employs to be identified.**

***Index Terms*—RFID, anti-collision, tag identification, EPC-global standard, VHDL**

I. Introduction

The growing concern in tracking, identification and localization systems has turn Radio Frequency Identification (RFID) technology into a mainstream in scientific research. This technology is especially attractive in areas like health care, supply chain, e-passports, and wireless sensor networks [1] [2] [3] [4]. RFID is a wireless ubiquitous technology, where a spectrum of radio frequency is used to transfer the identification information between two communication devices: tags and readers. Tags are uniquely identified with an identification code (ID), which consists of a sequence of bits. Due to the shared nature of the wireless channel used by tags, these systems are prone to transmission collisions. A collision occurs when two or more tags transmit information simultaneously. Collided tags must retransmit their IDs until they are identified, resulting in a wastage of bandwidth, energy, and an increase in identification delay and reader transmitted bits [5]. Anti-collision protocols are hence required to arbitrate these collisions.

Tree based protocols successively split collided tags into two or more subsets and the reader attempts to recognize each one of the subsets one by one [6]. Aloha-based protocols divide time into slots and tags randomly choose one slot to respond [7]. Aloha-based protocols present four main variants [8]. In Pure Aloha (PA), a tag will respond 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 only once per frame. While in FSA the frame size is fixed during the identification process, in DFSA it is variable. Finally, hybrid protocols combine tree and Aloha structures, benefiting from the advantages of Aloha-based and tree-based protocols [9].

The current standard in RFID systems is the one defined in the EPC C1G2 protocol (also included in the standard 18000-6C) [10]. EPC C1G2 employs a DFSA protocol to arbitrate collisions: the Slot Counter protocol, commonly known as the Q-protocol. In order to arbitrate the process, the reader updates the transmission frame size dynamically. In this context, a transmission frame is defined as a sequence of time slots where tags can only respond once. Conventionally, a slot comprises the time from the point that the reader sends a command to the point that the tags finish replying their temporary IDs. In this respect, the parameter Q is defined to update the frame size. This parameter is updated by adding or subtracting the value of C. Therefore, C represents a key element for the protocol performance, since it ultimately determines the transmission frame size. The standard does not specify the selection of C. It only recommends using high values if the previous Q value was low and vice versa, in the range of $[0.1, 0.5]$. This lack of definition has led to many different alternatives.

A simple solution found in the literature is setting C to a fixed value for the whole identification process [11]. A different approach is assigning C according to the current Q value [12] [13]. Additionally, a third possibility found is giving two different values to C depending on whether the current slot results in idle or collision response [11] [14] [15]. Although each protocol employs a different strategy, they all have in common the use of the parameter C to arbitrate the identification process. The study of this parameter presents a

978-1-4799-5344-8/15/$31.00 ©2015 IEEE

great relevance if the field of RFID, since it is tightly related to the frame size.

This paper focuses on the hardware analysis of tag-side of the different solutions proposed in the state of the art. The influence of the C value on the average number of clock cycles a tag employs to be identified will be studied. To the best of our knowledge, this relationship has not been studied before. To support this analysis, the tag state machine of the Q-protocol is designed in order to extract the number of tag-clock cycles. The tag state diagram of RFID anti-collision protocols and the tag-clock cycles computation is also presented in the state of the art [16] [17] [18] [19], but the authors only analyze tree-based protocols and any Aloha-based.

Subsequently, the remainder of this paper is organized as follows: Section II describes an overview of the standard; Section III presents the comparative protocols; Section IV implements and simulates the tag state diagram of the Q-protocol; Section V shows a comparison of the previously presented protocols and the simulations results; and finally, Section VI concludes the paper.

II. Anti-collision strategy in EPCglobal Class-1 Generation-2

The fact that most RFID manufacturers currently follow the EPC C1G2 standard highlights the research relevance of this protocol [7]. The standard EPC C1G2 employs a DFSA protocol to arbitrate collisions, the Slot Counter protocol, commonly known as Q-protocol. It specifies the transmission frame size (L) as a power of two, taking the value $L = 2^Q$, where $Q \in \mathbb{N}$ and $0 \leq Q \leq 15$. It also defines the parameter Q_{fp} to update the frame size, where $Q = round(Q_{fp})$. Fig.1 shows a flow diagram of the Q-protocol employed in the standard.

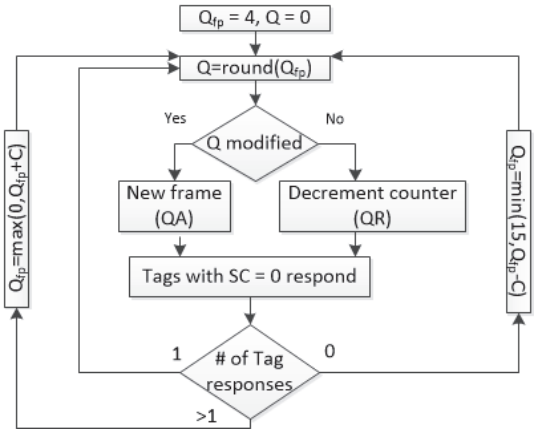

Fig. 1: Q-protocol flow chart

The reader starts the identification process by broadcasting a Query command and specifying the initial value of Q. This command is exclusively broadcast at the beginning of the identification round (first slot of the first frame). The reader then alternates between QueryAdjust (QA) and QueryRep (QR) commands to identify the set of tags. QA starts a new frame and implies tags to randomly select a slot in the frame, while QR tells tags to decrement their internal slot counter (SC). The reader starts a new frame by broadcasting QA with the corresponding Q. Then each tag computes the frame size and randomly chooses an integer from 0 to $L-1$ to update their SC with it. Those generating 0 contend the channel in the current slot by sending a randomly generated number of 16 bits length ($RN16$). Regarding the time slot occupancy, there are three possible scenarios to update the value of Q_{fp}:

- None of the tags replies. The slot is considered idle and $Q_{fp} = Q_{fp} - C$.
- Only one tag replies. The slot is considered single response and Q_{fp} remains unchanged. In this case, the reader replies the tag with an ACK command followed by the same $RN16$ received by the tag.
- More than one tag replies. The slot is considered collided and $Q_{fp} = Q_{fp} + C$.

Next, the reader readjust the frame size to $L = 2^Q$ where $Q = round(Q_{fp})$. At this point, according to [13] the reader can follow two strategies. The first one is to send a new QA and thus starting a new frame with the updated frame size only if the reader has reached the last slot of the frame. The second strategy, called *slot by slot*, consists on sending a QA every time Q differs from its previous value, independently of the number of remaining slots in the frame. In any case, if the condition for broadcasting a QA is not satisfied, a QR is sent instead, asking the tags which have not transmitted in the current frame to decrement their SC by one. Collided tags must wait for the reader to send a QA. Finally, identified tags leave the reading process, not responding to further reader queries. This procedure repeats until all tags have been identified. The *slot by slot* strategy has been followed for all the comparative protocols in this paper, since it provides a better performance according to [13].

III. Analysis of Q-protocol variants

As can be inferred, the Q parameter plays a significant role in the Q-protocol, so it should be carefully chosen. Moreover, since the value of Q is determined by the value of C, the selection of C is a key step in the protocol. According to [5] and [15], the optimal L maximizes the slots efficiency when the number of unidentified tags is equal to L. Therefore, Q-based anti-collision protocols must adopt an strategy to select C in order to dynamically adjust the frame size so that it approaches the optimal value. As shown in Table I, three main strategies to select C can be found in the literature, analyzed in the following subsections.

A. Fixed C

A simple solution found in the literature is fixing the value of C during the complete identification process [11]. In this paper the value $C = 0.3$ has been chosen for the comparison, since it falls about the middle of the allowed range.

TABLE I: CLASSIFICATION OF Q-PROTOCOL-BASED ALGORITHMS

Strategy	Protocol
C fixed	$C = 0.3$
$C = f(Q)$	optimalC $C = 0.8/Q$
C = f(slot state)	fastQ Q+ SCS

B. C as a function of Q

A different strategy to update the frame size is setting C according to the current Q value. A protocol which outputs different C values within a range (optimalC) is presented in [12]. They claim to obtain C from the simulation results for different frame sizes (or Q values) in the sense of minimizing the identification time. For this purpose $C \in [0.1, 0.5]$, and it can only take values in steps of 0.1 (5 different possible values). The value of C is chosen regarding that the higher the value of Q, the lower the value of C. However, their results are not conclusive, since the authors only compare optimalC with the Q-protocol with fixed frame size (Q does not change during the process of tag identification).

In [13] it is suggested to establish $C = 0.8/Q$ based on empirical results, although the authors claim to obtain poor results with this strategy.

C. C as a function of the current slot state

A different approach to select C is based on the current slot state (collision, idle or success). The Fast Q protocol (fastQ) presented in [14] introduces two different values for C to avoid unnecessary collided or idle slots. If the reader detects a collision C takes the value C_{col} while if the reader detects no response, C takes the value C_{idle}. They claim that C_{col} and C_{idle} should be proportional to Tr and Pr, where Tr represents the ratio between the time duration of a collided slot and idle slot and Pr represents the ratio between the probability of a collided slot and an idle slot. This solution presents a higher complexity, since it involves slots probability calculations.

Similarly, the protocol Q+, presented in [15], also separates C into two values, C_c for collided slots and C_i for idle slots. Both values are defined in order to optimize the efficiency of tag identification, setting the optimal ratio as $C_c/C_i = e - 2$. The authors also provide a formula to obtain the two variables regardless of knowing the number of tags, although they do not indicate the tag estimation procedure.

In [11] C is also replaced by two new variables: $c1$ for collided slots and $c2$ for idle slots. The presented protocol (SCS) obtains $c1$ and $c2$ slot by slot as a function of other parameters that mainly depend on *reader-to-tag* and *tag-to-reader* data rates. It is suggested to set $c2 \in [0.1, 1]$ and $c1 = 0.1$ claiming that it greatly improves the performance in comparison to the Q-protocol with fixed C value.

IV. HARDWARE ANALYSIS OF THE Q-PROTOCOL

The hardware design of tag-side of the Q-protocol is modeled with a block diagram, shown in Fig.2. Seven main blocks define the tag's chip: Control Unit, Finite State Machine, Outputs, Random Number Generator, ROM memory, Internal Counter and Transceiver. The first two blocks are synchronous, controlled by the clock signal *clk*, while the rest are asynchronous. The brain of the chip is the Control Unit. According to its inputs (*reader commands*, *SC* and *power*), this unit determines the value of the *control* signal. The tag state machine is controlled by the mentioned *control* signal. The signal *state*, representing the current state of the state machine, determines the actions which the tag must perform at any instant inside the Outputs block. Additionally, the system needs a random number generator to generate the RN16 and the initial value of *SC*. The signal *generate* is added to determine which of the two random numbers must be generated. If the generated number is the slot selection, it goes to the tag's counter (SC). Otherwise, the RN16 goes to the Transceiver block which sends the data to the reader. Lastly, whenever the signal *read* is active, the ID of the tag is transmitted to the Transceiver block and the ID is sent to the reader.

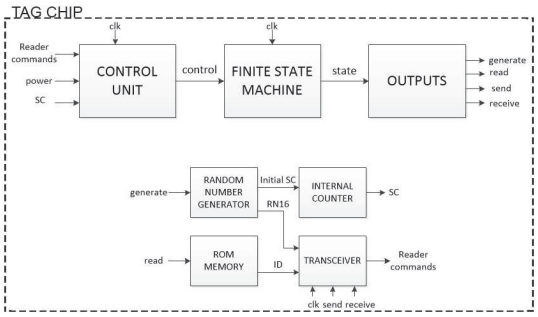

Fig. 2: Tag hardware block design of the Q-protocol

A special attention has to be paid to the tag state machine, since it is the core of the tag's chip. The tag-side of the Q protocol has been modeled with ten finite states, as shown in Fig.3. State transitions are based on the reader commands and the tags' operations results. Table II describes the main function which the tag must develop in each state. According to the standard, once the tag is identified, the reader can perform other operations with it, such as reading or writing into memory, opening files, updating security configurations or killing the tag. This work will only cover the identification commands. The presented tag's chip has been implemented in VHDL. In order to verify the design, the identification process of one tag has been simulated using ModelSim software. The reader's command are manually set inside the simulation profile. Moreover, it is assumed that the reader's and tag's clock are perfectly synchronized.

State transitions are activated with the positive edge of the clock. The number of clock cycles the tag must remain in the same state depends on the length of the commands sent by

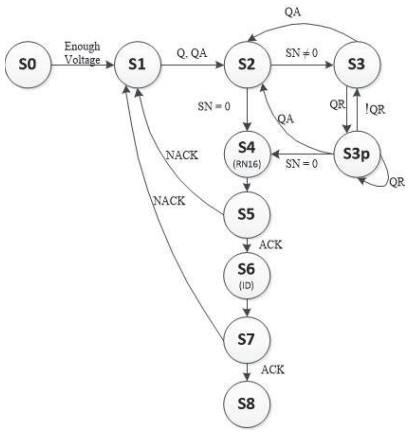

Fig. 3: Q-protocol tag state diagram

TABLE II: STATES' FUNCTION

State	Function
S0	Low power mode
S1	Receiving Q, QA
S2	SC selection
S3	Receiving QR
S3p	Decrementing SC
S4	Sending RN16
S5	Receiving nack/nack
S6	Sending ID
S7	Receiving nack/nack
S8	Other operations

the reader and the length of the data transmitted by the tag, assuming one clock cycle per symbol transmitted (data-0 or data-1). However, in order to visually observe the complete identification process of one tag in the example of Fig.4, it is assumed that any transmission or reception of data as well as SC selection involve only one clock cycle. It is important to note here that this simplification will no longer be considered in the next section, when the number of tag-clock cycles will be extracted.

Simulation results are shown in Fig.4. At any point of the identification process, if the tag does not receive enough power voltage, it restarts to S0 from any current state. In this work only passive tags, which are powered by the reader, are being considered. The tag is initially in S0, and after being powered by the reader, it jumps to S1. The reader initiates the identification round by sending a Q command, containing the frame size. The tag then transits to S2 and chooses the slot 1. Since it is not equal to 0, it goes to the S3. After receiving a QR, it goes to S3p and decrements its SC, taking the value 0. Then, it jumps to S4 and sends the RN16. When the tag finishes sending the 16 bits, it transits to S5 and waits for the reader's response. In this example, it receives a NACK meaning that the reader has detected a collision. Therefore, the tag returns to S1 and waits for a QA. When a QA is received, the previous process is repeated up to S5. In this case, the reader detects a single tag response and sends an ACK followed by the same RN16 received. Therefore, the tag jumps to S6 and transmits its ID together with the CRC. When transmission is finished, the tag goes to S7 and waits for the reader's response. Next, the reader acknowledges the ID transmission, so the tag is successfully identified. Finally, the tag goes to S8 and waits for further commands. In S8 the tag has already been identified.

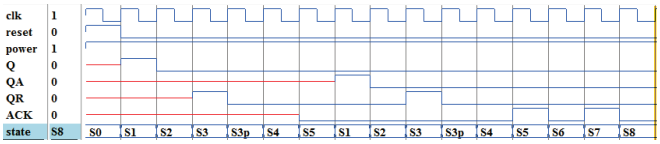

Fig. 4: Identification process for 1 tag

V. Performance Evaluation

This section presents the results of the simulation experiments using Matlab R2013a software. The performance of the protocols presented in Section III has been evaluated. The different strategies to update the frame size with C have been employed considering the state machine presented in the previous section.

The proposed simulation defines a scenario with one reader and a varying number of tags, n, from 8 to 256 tags with a step size of 8. These numbers have been chosen providing that real systems do not usually deal with very high amounts of tags. It is assumed that the transmission channel is ideal and the tags' slot selection is uniformly distributed along the frame [7] [11] [14]. The simulation responses are averaged over 1000 iterations for accuracy in the results. For all the simulated protocols in the comparison, it is assumed that the identification procedure ends when all tags have been identified. Table III shows the values of the parameters used for the simulations according to the standard specifications [10].

TABLE III: TYPICAL SYSTEM PARAMETERS OF EPC C1G2

Parameter	Value
Length of Query (lq)	22 bits
Length of QueryAdjust (lqa)	9 bits
Length of QueryRep (lqr)	4 bits
Length of ACK/NACK ($lack$)	18 bits
Length of ID (lid)	64 bits
Length of CRC ($lcrc$)	16 bits
Length of RN16 ($lrn16$)	16 bits

A. Simulations assumptions

Some algorithms leave some parameters unspecified. In those cases, the following is assumed:

- Q+ protocol: the computing methodology for C_c is not accurately specified in [15]. The authors only state that $C_c \in [0.1 - 0.5]$. Therefore, it is assumed $C_c = 0.35$, leading to $C_i = (e-2) * C_c = 0.2514$.
- SCS protocol: $c1$ and $c2$ are computed assuming the typical values of the standard specified in [20]. Herein, the values $c2 = 0.1253$ and $c1 = 0.1$ are employed.

They show that Q updates faster when a collided reply occurs than when there is no reply

B. Average number of clock cycles a tag employs to be identified

As mentioned in section IV, state machine transitions are activated with the rising edge of the tag's clock. In the designed model, one clock cycle represents one transmission symbol, data-1 or data-0. The standard EPC C1G2 specifies different duration for each symbol. However, for the purpose of the analysis in this work, equiprobable 0 and 1 symbol is assumed [21]. Moreover, it is assumed that one symbol corresponds to one bit. In order to obtain more realistic results, the previously made assumption that each state transition implies only one clock cycle is no longer considered. When the tag is in any of the four receiving states (S1, S3, S5, S7), it must wait in the same state as many clock cycles as the length of the reader command, specified in Table II. Additionally, when the tag is in any of the sending state (S4, S6), it must wait in the same state as many clock cycles as the length of the transmitted data. The power signal is not considered, since it is assumed that at the beginning of the reading process all tags are being powered. The analysis also considers that the tag employs one clock cycle to select SC and also one clock cycle to decrement it.

The minimum number of clock cycles a tag needs to be identified has been obtained in (1), according to the values given in Table III. This calculation considers that the tag selects slot 0 as the response to Q and experiments no collision. Thus, the tag is identified in the first slot of the first frame.

$$clk_{min} = lq + 1(SC = 0) + lrn16 + lack + \\ + lrn16 + lid + lcrc + lack = 171\ clks \quad (1)$$

After receiving any of the reader's queries (Q, QA or QR) the increase produced in the number of clock cycles is analyzed in (2), (3), and (4), according to whether the current slot results in successful, idle or collided response, respectively. The clock cycle introduced by SC in the equations is due to setting a new value to SC (if Q or QA is received) or decrementing it (if QR is received).

$$clk_{success} = 1(SC) + lrn16 + lack + \\ + lrn16 + lid + lcrc + lack = 149\ clks \quad (2)$$

$$clk_{iddle} = 1(SC)\ clks \quad (3)$$

$$clk_{collision} = 1(SC) + lrn16 + lnack = 35\ clks \quad (4)$$

With this, simulation results are shown in Fig.5. It can be appreciated that the protocols fastQ, Q+ and SCS present the lowest number of tag-clock cycles. A considerable increase in the tag-clock cycles with the tag population is also observed for all the comparative protocols. Moreover, the number of tag-clock cycles is always much higher than the minimum obtained in (1), meaning that it is very likely that the tag experiments several collisions before being identified. This

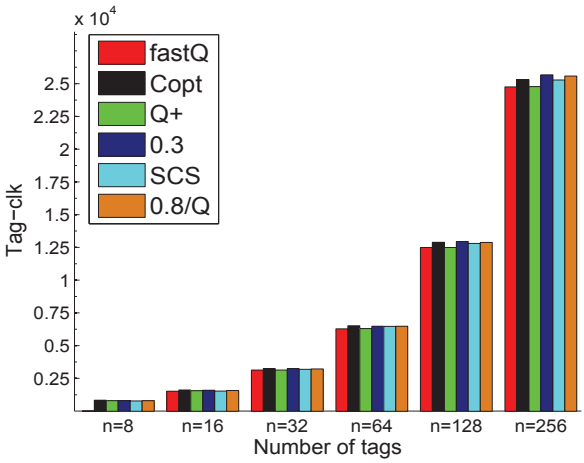

Fig. 5: Clock cycles to identify one tag

behavior is greatly influenced by the initial frame size. The number of collision slots has also been obtained in order to extract a relation between this number and the number of tag-clock cycles. Results are shown in Fig.6. A correlation between the number of collision slots and the number of clock cycles can be appreciated, since the protocols which present fewer collided slots (fastQ, Q+, SCS) also present the lower number of tag-clock cycles. Moreover, the number of collision slots increase with the tag population, justifying the increase of the the tag-clock cycles with the number of tags.

From Fig.5 it can also be observed that the strategy C = f(slot state) presents better results in relation to the other two strategies. It can also be appreciated that for this strategy, the higher the ratio $C_{collision}/C_{idle}$, the lower the number of clock cycles. To corroborate this fact, this ratio has been obtained for the three protocols inside this category:

- fastQ: $C_{collision}/C_{idle} = 1.4122$
- Q+: $C_{collision}/C_{idle} = 1.3922$
- SCS: $C_{collision}/C_{idle} = 1.2530$

A higher ratio means that the frame is being widened at a higher pace than it is being narrowed, since a higher weight is given to collided slots in relation to idle slots. As this ratio increases, a decrease is found in the number of collision slots leading to a lower number of tag-clock cycles.

VI. Conclusion

Relevant results have been achieved regarding current DFSA proposals: hardware analysis and direct comparison of the most important strategies used in the Q-protocol to update the frame size with the parameter C. RFID technology is widely being employed nowadays, so the physical implementation of the anti-collision protocols can not be left apart. For this reason, this work has covered the study of the current standard in this technology. Moreover, the advantages of performing a hardware-based analysis instead of software-based are clear regarding a possible future physical implementation. By em-

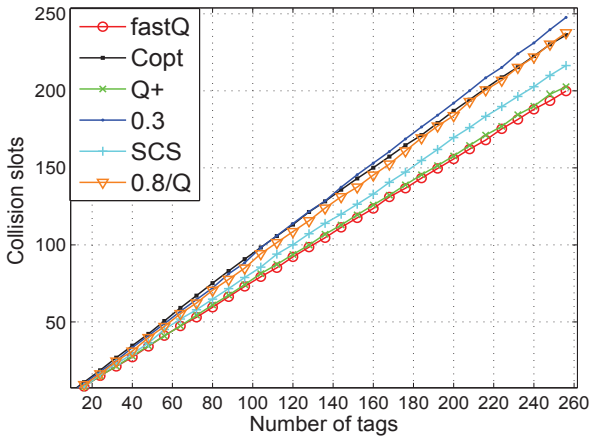

Fig. 6: Total collision slots

ploying VHDL for the design, the protocol implementation on a FPGA is almost straightforward.

To conclude, some outcomes are presented. Three main strategies are found in the literature to determine the value of C in the Q-protocol: fixing C to a constant value, setting C as a function of Q and setting C as a function of the current slot state. The strategy where *C = f(slot state)* presents the lowest number of clock cycles a tag needs to be identified in addition to the lowest number of collided slots. Inside this strategy, the higher the ratio $C_{collision}/C_{idle}$ the lower the number of tag-clock cycles. Therefore, if the scope of the RFID system is to minimize the number of clock cycles, the protocol fastQ is highly recommended to be employed.

As a concluding remark, some future work is presented. The standard suggests $L = 4$ as the initial frame size. However, this parameter greatly influences the protocols performance. Future work could lead to the study of the effect of the initial frame size over the number of tag-clock cycles. Furthermore, the hardware design and implementation of the reader's side of the Q-protocol would also be a relevant study.

REFERENCES

[1] A. Shirehjini, A. Yassine, and S. Shirmohammadi, "Equipment location in hospitals using RFID-based positioning system," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, pp. 1058–1069, Nov 2012.

[2] Y. Li and X. Ding, "Protecting RFID communications in supply chains," in *Proceedings of the 2Nd ACM Symposium on Information, Computer and Communications Security*, ASIACCS '07, (New York, NY, USA), pp. 234–241, ACM, 2007.

[3] M. Meingast, J. King, and D. Mulligan, "Embedded RFID and everyday things: A case study of the security and privacy risks of the U.S. e-Passport," in *IEEE International Conference on RFID, 2007.*, pp. 7–14, March 2007.

[4] V.-T. Hoang, N. Julien, and P. Berruet, "Cluster-head selection algorithm to enhance energy-efficiency and reliability of wireless sensor networks," in *20th European Wireless (EW) Conference.*, (Barcelona, Spain), May 2014.

[5] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. New York, NY, USA: John Wiley & Sons, Inc., 2 ed., 2003.

[6] D. Hush and C. Wood, "Analysis of tree algorithms for rfid arbitration," in *Information Theory, 1998. Proceedings. 1998 IEEE International Symposium on*, pp. 107–, Aug 1998. 3.

[7] C. Wang, M. Daneshmand, K. Sohraby, and B. Li, "Performance analysis of RFID Generation-2 protocol," *IEEE Transactions on Wireless Communications*, vol. 8, pp. 2592–2601, May 2009.

[8] D. Klair, K.-W. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," *Communications Surveys Tutorials, IEEE*, vol. 12, pp. 400–421, Third 2010.

[9] H. Wu, Y. Zeng, J. Feng, and Y. Gu, "Binary tree slotted aloha for passive RFID tag anticollision," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, pp. 19–31, Jan 2013.

[10] "Radio Frecuency Identity Protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz-960 MHz," November 2013.

[11] M. Daneshmand, C. Wang, and K. Sohraby, "A New Slot-Count Selection Algorithm for RFID Protocol," in *Second International Conference on Communications and Networking in China, 2007. CHINACOM '07.*, pp. 926–930, Aug 2007.

[12] I. Joe and J. Lee, "A novel anti-collision algorithm with optimal frame size for RFID system," in *5th ACIS International Conference on Software Engineering Research, Management Applications, 2007. SERA 2007.*, pp. 424–428, Aug 2007.

[13] C. Floerkemeier and M. Wille, "Comparison of transmission schemes for framed aloha based RFID protocols," in *SAINT Workshops 2006. International Symposium on Applications and the Internet Workshops, 2006.*, pp. 4 pp.–97, Jan 2006.

[14] J. Teng, X. Xuan, and Y. Bai, "A Fast Q algorithm based on EPC Generation-2 RFID protocol," in *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), 2010*, pp. 1–4, Sept 2010.

[15] K. K. Donghwan Lee and W. Lee, "Q+-algorithm: An enhanced RFID tag collision arbitration algorithm," *4th International Conference on Ubiquitous Intelligence and Computing, Hong Kong, China*, July 11-13, 2007.

[16] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, "Evaluating and optimizing power consumption of anti-collision protocols for applications in rfid systems," in *Proceedings of the 2004 international symposium on Low power electronics and design*, pp. 357–362, ACM, 2004. 15.

[17] S. M. A. Motakabber, M. A. Mohd Ali, and N. Amin, "VLSI Design of an Anti-Collision Protocol for RFID Tags," *European Journal of Scientific Research*, vol. 28, p. 560, March 2009.

[18] M. O. Jahariah Sampe and I. Baharin, "Hardware Implementation of Fast Detection Anti-Collision Algorithm For RFID System," in *International Conference on Electrical Engineering and Informatics. Institut Teknologi Bandung, Indonesia.*, June 17-19. 2007.

[19] S. S. A. S. Joyashree Bag, k. Senthill Kumar and P. S. K. Sarkar, "VLSI Implementation of priority selection Algorithm to select tags using RFID system," in *International Conference of Sustainable Energy and Intelligent Systme (SEISCON). DR. M.G.R. University Maduravoyal, Chennai, Tamil Nadu, India.*, July 20-22 2011.

[20] W.-T. Chen, "A feasible and easy-to-implement anticollision algorithm for the EPCglobal UHF Class-1 Generation-2 RFID Protocol," *IEEE Transactions on Automation Science and Engineering*, vol. 11, pp. 485–491, April 2014.

[21] F. Hessar and S. Roy, "Energy based performance evaluation of passive epc gen 2 class 1 rfid systems," *Communications, IEEE Transactions on*, vol. 61, pp. 1337–1348, April 2013. 46.