Expowave: An RFID anti-collision algorithm for dense and lively environments

Nikolaos Konstantinou, *Member, IEEE*

***Abstract*—This paper analyzes and proposes Expowave, a distributed algorithm for the scheduling of an RFID reader network. The behavior of the algorithm is presented in detail, and its performance is evaluated through a set of simulation experiments. It is demonstrated that the algorithm constitutes an efficient approach to the reader anti-collision problem, especially in dense and lively environments.**

***Index Terms*—RFID, Reader, Anti-collision.**

I. Introduction

DURING the latest years, we are witnessing ubiquitous computing becoming more and more a part of our every day lives. RFID tags and sensors are constantly gaining popularity, to the extent that the materialization of the Internet of Things vision is constantly evolving towards its wider adoption. Research and Development in RFID has made applications possible in various domains such as person identification [1], retail stores [2], [3], asset tracking [4], etc. RFID tags are becoming cheaper and cheaper and, compared with other identification technologies such as Datamatrix and Barcodes, they seem to be gaining ground since their price, that has always been the major problem since the inception of the idea, is constantly decreasing while the benefits that occur from their use are increasing.

Since its inception, the basic idea of functioning of an RFID tag is *backscatter* [5]: An RFID reader, also referred to as an interrogator, broadcasts a signal in his interrogation zone. If an RFID tag finds itself in this interrogation zone, it backscatters (i.e. transmits back) a signal containing its unique identification number.

The signals that are transmitted and received by the readers and the tags are in a specific frequency and, as such, various problems arise when lots of RFID readers are placed close to each other, forming a densely covered area. Interference between two readers can occur even when these readers' interrogation zones do not overlap [6]–[8].

In this paper, we study this problem, we propose a novel approach for RFID reader anti-collision and we compare it experimentally, through simulation experiments, to the approaches described in the bibliography. The proposed algorithm is proved to be of higher capacity in dense RFID reader networks. Practical use cases of such algorithms include cases when an area needs to be covered with RFID readers in a way that objects moving through are correctly and efficiently identified.

Manuscript received June 13th, 2010; revised September 13th, 2010. This work was partially supported by the FP7 Project ASPIRE.

The author is with the Athens Information Technology (AIT).

The problem in these cases can be simulated as a graph, where each node represents an RFID reader and each edge represents the ability of a collision to occur between the two readers. Thus, the reader anti-collision problem is expressed as: "how can we organize the system in order to minimize node collisions and achieve the maximum throughput"?

The paper is structured as follows: Section II presents the approaches that have been proposed in the literature and Section III describes in detail the proposed algorithm. Section IV presents the experiments that were conducted in order to measure the algorithm's efficiency and Section V concludes the paper with our final observations and remarks.

II. Related Work in Reader Anti-Collision Management

First, we need to note in dense and lively RFID environments, two types of reader-to-reader interferences can occur. The first is the *frequency interference*, also called reader-reader collision and it occurs when two or more readers communicate on the same frequency at the same time.

The second type of interference is the *tag interference*, also called reader-tag collision and it occurs when two or more readers attempt to communicate with a particular RFID tag at the same time. Reader-tag collision protocols can be binary tree-based, as the work presented in [9] or aloha-based [10].

The results presented in this paper provide a solution to the former case. In particular, we do not attempt to resolve interference problems but rather to avoid them by scheduling the readers to communicate in different time periods.

The reader anti-collision problem is widely studied in the bibliography. Many approaches to its solution have been adapted from the sensor network research domain, as in [11]–[14].

Among the most notable, and one of the first reader-reader collision avoidance algorithms is Colorwave [15]. Colorwave considers an "interference graph" over the readers, where an edge between two readers means the probability of a collision when transmitting simultaneously and it tries to randomly color this graph in order for each pair of readers to have different colors, each color representing a time slot. The solution presented in this paper builds on top of Colorwave, by inheriting its key characteristics and improving its properties.

In [16], the authors suggest coloring the interference graph using k colors, where k is the number of available channels. If the graph is not k-colorable using their suggested heuristic, then the authors suggest removal of certain edges and nodes from the interference graph using other heuristics which consider the size of the common interference regions between neighboring readers.

In [17], the authors present a CSMA-based MAC protocol for the collision avoidance in dense RFID networks. The implementation is based on mote readers.

Tanaka et al. [18] present a linear programming formulation method to obtain communication probability of the readers for a given reader deployment scenario. They also propose two algorithms based on the detect-and-abort principle for mitigating the reader-to-tag as well as the reader-to-reader interference in dense reader environments.

In [19], [20] the authors present an approach where the DCS algorithm is based on the probability p (hence Probabilistic DCS, PDCS): the probability for a node to change its color after a collision. The authors offer a theoretical study on the properties of p in the system performance.

The Pulse protocol [21] is referred to as beacon broadcast and CSMA mechanism. According to the Pulse protocol, each reader uses two separate channels in the RFID system, the data channel and the control channel. The former is used for reader to tag communication while the latter is used for reader to reader communication. Messages broadcasted in each of the channels do not interfere between them. According to the protocol, each reader continuously transmits beacon signals through the control channel while communication with the tags is made through the data channel.

HiQ [22] is an online learning algorithm, used to find dynamic solutions to the reader collision problem in RFID systems. The algorithm contains two parts: First, it allocates resources in order to maximize the number of readers communicating at a single time period and second, it attempts to minimize the number of collisions these readers experience when they are communicating. The algorithm utilizes three basic hierarchical tiers in its control structure, namely, readers, R-servers, and Q-servers. Among the limitations of this protocol is that Q-learning assumes collision detection of readers not in sensing range. This means that in the case that some collisions cannot be detected, the protocol will not operate correctly.

AC_MRFID [23] is a protocol based on DCS, especially suitable for networks with a regular deployment. However, this protocol is not fair, since it provides the readers with few neighbors in their interrogation range, with more resources. Furthermore, it introduces additional communication overhead, in order to count the neighbors.

Also, we need to mention that there is also a category of anti-collision protocols in multi-channel approaches where the readers do not only use one channel (i.e. frequency) but multiple frequencies. For RFID, international standards [24] suggest the use of frequency between 860 MHz and 960 MHz. However, despite the fact that the interference signal strength of adjacent channels is regulated by 20dBCh, a reader can still interfere with the signal of an adjacent channel when it tries to read tags.

Finally, a more detailed survey on reader anti-collision protocols can be found on [7].

III. Algorithmic description

This Section describes the proposed Expowave algorithm. First, we have to define the concepts that are used, which are contained in the following list:

1) *Time slot* or *color*. The index that identifies the time a reader has to scan its environment for new tags. A color is a reserved timeslot, assigned to an RFID reader.
2) *Birth probability (σ)*. The probability a new tag to appear in a reader's interrogation zone in a specific time slot.
3) *Iteration*. The time between two consecutive kick slots, when communication between readers occurs (see Figure 1).
4) *Transmission*. The state when the reader communicates with its environment.
5) *Attempt*. When a reader attempts to scan its area for tags. An attempt can lead to a success or a collision.
6) *Neighbors*. Two readers are considered neighbors if transmitting at the same time slot (i.e. color) can lead to a collision between them.
7) *Idle*. The state of a reader when it does not perform any kind of communication with its environment.
8) *Collision*. The interference that occurs between two readers and prevents them from successfully reading the tags in their area. A collision occurs when two adjacent readers in the same time attempt to broadcast a signal in the same timeslot and one's signal causes interference in the other one's signal.
9) *Success*. The communication of a reader and a tag.

Keeping these in mind, for the operation of the algorithm, we make the following assumptions:

- Time is divided in discrete time slots. During these time slots, each node can either scan its interrogation area for RFID tags (this time slot is referred to as a color) or communicate with its neighbors (kick slot). Thus, there is no need for a distinct communication channel.
- Nodes are synchronized. They do not have to know necessarily the iteration number but they need to know when a timeslot starts and ends.
- Each node has the capability of detecting a collision.
- Each node can communicate with its neighbors. This happens during the kick slot, as displayed in Figure 1. Collisions during the kick slot are "*kick collisions*", as opposed to the previously mentioned "*collisions*".
- Each node possesses one of the three states: idle, transmitting, or collided.

Expowave algorithm builds on top of Colorwave. As such, it is a distributed (or on-line) algorithm in the sense that each node operates based on local information. There is no need of a central entity maintaining system-wide information and each algorithm execution has knowledge only of a local cluster of readers.

As depicted in Figure 1, time is divided into discrete timeslots. Kick slots (t_k) are the timeslots during which kick signals are sent while colors (c_1, c_2, …) represent time periods over which the readers consecutively attempt to read their respective interrogation zones. Although similar to Colorwave, the Expowave algorithm introduces the following modifications and additions.

- First, the algorithm introduces an upper bound to the colors (i.e. time slots) a period can have. This happens in

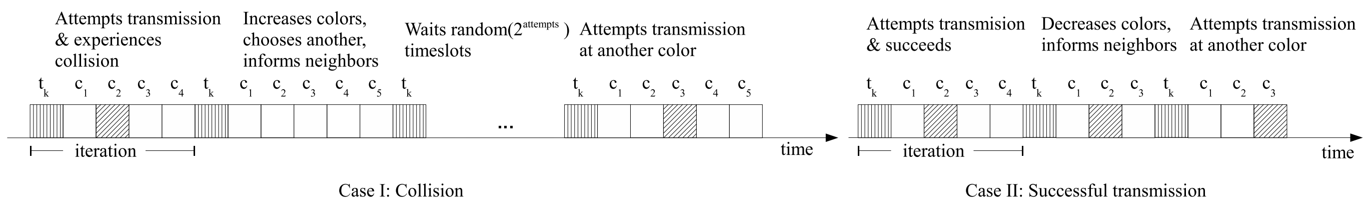


Fig. 1: Expowave behavior

order to prevent the number of colors from increasing to unpredictable – even unacceptable – levels. As it can be seen in Figure 1, a high number of colors means more time between algorithm iterations. Expowave assures an upper time bound (proportional to the `color upper bound` variable) between two consecutive iterations.

- After each attempt, if a collision is experienced, the reader will wait for a random amount of time between zero and $2^{attempts}$, in a fashion similar to slotted Aloha. As demonstrated in the experiments Section, the introduction of the exponential backoff in the Colorwave algorithm provides significantly greater throughput compared to the delay it introduces to the system.

A. *From simulation to practice*

We need to note that physical readers that operate at distinct timeslots, at a practical level, can be configured with the use of a middleware. With the use of a middleware solution, such as Fosstrak[1], Rifidi[2], or AspireRfid[3], we can configure adjacent physical readers to operate at distinct timeslots.

Note that as far as RFID readers are concerned, the suggested approach cannot be applied everywhere. According to the EPCglobal[4] specifications that regulate to a large extent the RFID-related implementations, an Event Cycle is the smallest unit of interaction between an ALE (Application Level Events) client and an ALE implementation through the ALE Reading API [25]. The API itself, however, does not provide a client with explicit control over the frequency at which reader cycles are completed and leaves implementation details up to the client. Therefore, in practice, there is no standard way provided by ALE for clients to control reader cycle timing. Implementations *may* provide different means for this through configuration files, administrative interfaces and so on.

IV. Experiments

The measurable efficiency parameters investigated in the algorithm are the throughput and the delay. These are defined as follows:

- Throughput = $\dfrac{\text{total queries sent successfully by a reader}}{\text{total time in iterations}}$
- Delay = $\dfrac{\text{time in iterations required for an appearing tag to be identified by a reader}}{\text{total time in iterations}}$

[1]Fosstrak: http://www.fosstrak.org
[2]Rifidi: http://www.rifidi.org
[3]AspireRfid: http://wiki.aspire.ow2.org
[4]EPCglobal: http://www.epcglobalinc.org

System-wise, throughput (resp. delay) is the mean average of the throughput (resp. delay) of all the RFID readers (or nodes) in the graph.

A. *Experimental setup*

In order to initiate the experiments, three random graphs of 250 nodes each were created: sparse, medium and dense. The nodes in the first case have a 10% probability of being connected to each other, 50% in the second and 90% in the third. The mean average of the neighbor nodes in the sparse, medium, and dense graph is 25.08, 123.79, and 224.45, respectively.

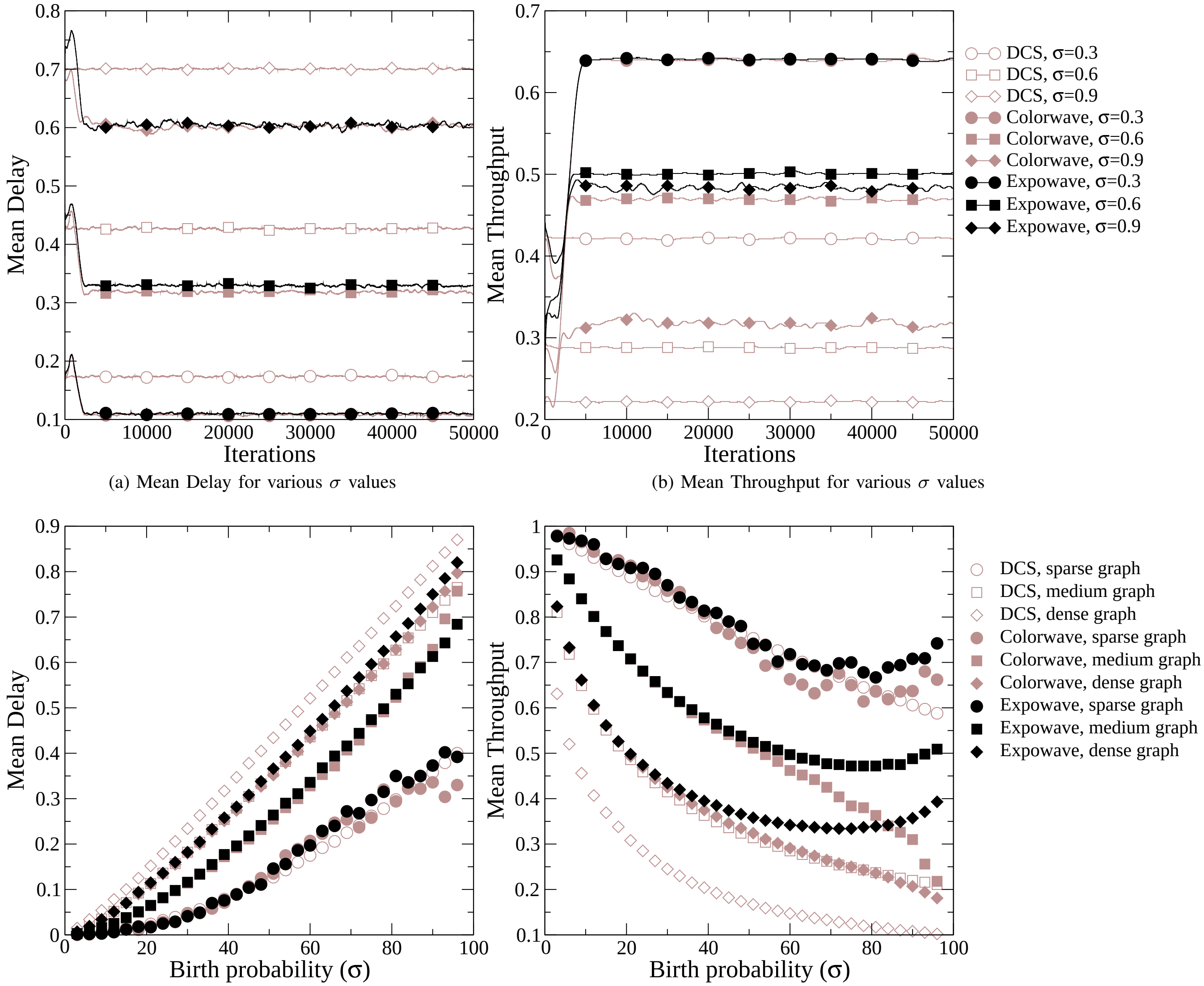
In order to simulate the experiments, we run DCS, Colorwave and Expowave on all the graphs. In order to run one experiment, for each iteration, according to a probability σ, it is decided whether an RFID tag will appear in each node's interrogation zone. We refer to σ as the birth (appearance of a new tag) probability in each iteration.

B. *Performance evaluation*

Figure 2 demonstrates the proposed algorithmic approach compared with Colorwave and DCS. It needs to be noted that Colorwave has been proven to be more efficient than DCS because of the former's dynamic nature [15].

Graph-wise, the algorithm behavior is displayed in Figures 2a and 2b where the values displayed are the mean values of all the graph nodes. The graph used in 2a and 2b is the graph of 250 nodes of medium density, as mentioned above. Parameters `upTrig`, `upSafe`, `dnTrig` and `dnSafe` are set to 0.9, 0.93, 0.99 and 0.98, respectively. In both Figures 2a and 2b, the experiments were run with a value of σ of 0.3, 0.6, 0.9, for the DCS, Colorwave and Expowave algorithms, all starting at 12 colors – and DCS staying on 12 colors because of its static nature. We can deduce in Figure 2b that the respective throughput does not behave the same for all cases of σ. For small values of σ (0.3), Colorwave achieves similar throughput to Expowave, but for values 0.6 and 0.9 the difference in the throughput between Colorwave and Expowave increases, with the latter being at all cases the higher. The DCS results are poor at all cases, compared to the other algorithms.

The conclusion that can be drawn by Figures 2a and 2b is that the delay in Expowave is in general slightly greater than the delay in Colorwave. This happens because of the introduction of the exponential backoff in case of a collision. However, the difference in the respective throughput is greater, which means that with a little increase in the delay we can achieve higher throughputs: In the case when σ is 0.9, Expowave



(a) Mean Delay for various σ values

(b) Mean Throughput for various σ values

(c) Mean Delay (`color upper bound` = 30)

(d) Mean Throughput (`color upper bound` = 30)

Fig. 2: Expowave performance evaluation

achieves a 52.84% greater throughput than Colorwave, by introducing a 0.33% greater delay. Also, in this case, compared to the static DCS algorithm, the delay in Expowave decreases by 13.71% while the throughput increases by 118.55%. This renders Expowave an anti-collision algorithm mostly suitable for dense and lively environments.

The next set of experiments illustrated in Figures 2c and 2d was taken as follows: The system was set to run for a number of 50000 iterations, for the sparse, medium, and dense graph with the parameters `upTrig`, `upSafe`, `dnTrig` and `dnSafe` set as in the previous set of experiments. Its final values of delay and throughput were recorded, while the value of σ increased from 0.01 to 0.99, the initial number of colors was again 12 for all algorithms, and the `color upper bound` variable was set to 30.

In this case we notice that for small σ values, Expowave behaves almost the same as Colorwave, while DCS is a poorer choice in the medium and dense graphs. As σ increases above 50%, we notice the throughput in Expowave increases with respect to Colorwave and DCS, while in the same time presenting an increase at the respective additional delay; note especially the dense graph. As it can be seen in Figures 2c and 2d, when σ is 0.99, Expowave compared to Colorwave gives a 13.83% greater throughput with a 28.61% greater delay in the sparse graph, a 188.77% greater throughput with a 12.23% less delay in the medium density graph, while for the dense graph, the respective numbers are 155.42% greater throughput and 2.78% greater delay. This strengthens the point that Expowave will find better usage in more dense and more lively environments.

Initially, there was a problem in comparing the behavior between Colorwave and Expowave, because of the fact that the former, under heavy load will increase the number of `max_colors` of each node. This increase could lead to non-comparable results since, between two cases that present the same throughput, the one with the less `max_colors` is optimal: less real time will intercede between two consecutive kick slots in this case (see Figure 1). With the introduction of the `upper color bound` variable, we assure that measurements are realized on an equal basis. Most importantly, we assure that the number of colors per node will not be left free to increase uncontrollably since high `max_colors` values entail large time intervals between consecutive kick slots.

We notice in Figure 2d that in the throughput in Expowave slightly increases for values of σ greater than 80%. This is caused by the values of `upTrig` and `upSafe` which determine when will a node increase its `max_colors` variable. With these values being 0.9 and 0.93 respectively, the number of nodes that will increase their `max_colors` will be higher as σ increases.

V. Conclusions

In this paper we defined the main concepts regarding the anti-collision problem in dense RFID networks, we presented and analyzed the Expowave algorithm, a novel approach for anti-collision in a multiple access protocol.

Nevertheless, we cannot claim that the proposed algorithm constitutes a single panacea for the anti-collision problem. As demonstrated and analyzed, the increase in the throughput is associated with a delay. In cases when new tags appear in high rates and the environment is covered by a dense RFID reader network, the proposed algorithm outperforms Colorwave and DCS since the throughput achieved is substantially higher than the delay introduced.

Acknowledgment

Part of this work has been carried out in the scope of the ASPIRE project (FP7-215417) (http://www.fp7-aspire.eu). The author acknowledges help and contributions from all partners of the project.

References

[1] S. Garfinkel and B. Rosenberg, *RFID: Applications, Security, and Privacy*. Addison-Wesley, 2005.

[2] G. Roussos, "Enabling RFID in Retail," *Computer*, vol. 39, no. 3, pp. 25–30, 2006.

[3] G. Roussos, P. Kourouthanassis, and T. Moussouri, "Appliance design for mobile commerce and retailteinment," *Personal and Ubiquitous Computing*, vol. 7, no. 3–4, pp. 203–209, 2003.

[4] F. Thiesse, E. Fleisch, and M. Dierkes, "LotTrack: RFID-based process control in the semiconductor industry," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 47–53, 2006.

[5] H. Stockman, "Communication by means of reflected power," *Proceedings of the IRE*, vol. 36, pp. 1196–1204, October 1948.

[6] G. P. Joshi, K. M. A. Mamun, and S. W. Kim, "A Reader Anti-collision MAC Protocol for Dense Reader RFID System," in *IEEE International Conference on Communications and Mobile Computing*, 2009, pp. 313–316.

[7] G. P. Joshi and S. W. Kim, "Survey, Nomenclature and Comparison of Reader Anti-Collision Protocols in RFID," *IETE Tech Rev*, vol. 25, no. 5, pp. 285–292, 2008.

[8] S. Kim, S. Lee, and S. An, "Reader collision avoidance mechanism in ubiquitous sensor and RFID networks," in *International Conference on Mobile Computing and Networking, Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization*, Los Angeles, CA, USA, September 2006, pp. 101–102.

[9] J. Myung, W. Lee, and J. Srivastava, "Adaptive binary splitting for efficient RFID tag anti-collision," *IEEE Communications Letters*, vol. 10, no. 3, March 2006.

[10] J. Zhai and G.-N. Wang, "An Anti-collision Algorithm Using Two-Functioned Estimation for RFID Tags," in *Computational Science and Its Applications ICCSA 2005, Optimization: Theories and Applications (OTA) 2005 Workshop*, ser. Lecture Notes in Computer Science, vol. 3483/2005. Springer Berlin / Heidelberg, May 2005, pp. 702–711.

[11] L. Schmidt, N. Mitton, and D. Simplot-Ryl, "Towards Unified Tag Data Translation for the Internet of Things," in *Wireless Communication Society, Vehicular Technology, Information Theory and Aerospace & Electronics Systems Technology (VITAE'09)*, Aalborg, Denmark, 2009.

[12] A. Gallais, F. Ingelrest, J. Carle, and D. Simplot-Ryl, "Preserving Area Coverage in Sensor Networks with a Realistic Physical Layer," in *Proc. IEEE INFOCOM'07*, Anchorage, Alaska, May 2007.

[13] A. Gallais, J. Carle, D. Simplot-Ryl, and I. Stojmenovic, "Ensuring K-Coverage in Wireless Sensor Networks with Realistic Physical Layers," in *Proc. IEEE Sensors'06*, Daegu, Korea, October 2006.

[14] M. Cardei, M. Thai, Y. Li, and W. Wu, "IEEE INFOCOM 2005," in *Energy-Efficient Target Coverage in Wireless Sensor Networks*, Miami, FL, United States, March 2005.

[15] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: An Anticollision Algorithm for the Reader Collision Problem," in *IEEE International Conference on Communications*, vol. 2, May 2003, pp. 1206–1210.

[16] V. Deolalikar, M. Mesarina, J. Recker, D. Das, and S. Pradhan, "Perturbative Time and Frequency Allocations for RFID Reader Networks," HP Labs, Tech. Rep., 2005.

[17] S. Jain and S. Das, "Collision Avoidance in a Dense RFID Network," in *WiNTECH'06*, Los Angeles, California, USA, September 2006.

[18] Y. Tanaka and I. Sasase, "Interference Avoidance Algorithms for Passive RFID Systems Using Contention-Based Transmit Abortion," *IEICE TRANSACTION COMMUNICATIONS E SERIES B*, vol. 90, no. 11, pp. 3170–3180, 2007.

[19] F. Gandino, R. Ferrero, B. Montrucchio, and M. Rebaudengo, "Introducing Probability in RFID Reader-to-Reader Anti-collision," in *2009 Eighth IEEE International Symposium on Network Computing and Applications*, Cambridge, Massachusetts, July 2009, pp. 250–257.

[20] ——, "Probabilistic DCS: An RFID reader-to-reader anti-collision protocol," *Journal of Network and Computer Applications*, 2010.

[21] S. Birari and S. Iyer, "Mitigating the reader collision problem in RFID networks with mobile readers," in *13th IEEE International Conference on Networks*, November 2005.

[22] J. Ho, D. W. Engels, and S. E. Sarma, "HiQ: A Hierarchical Q-Learning Algorithm to Solve the Reader Collision Problem," in *Proceedings of the International Symposium on Applications on Internet Workshops*, Phoenix, Arizona, January 2006, pp. 88–91.

[23] K. C. Shin, S. B. Park, and G. S. Jo, "Enhanced TDMA Based Anti-Collision Algorithm with a Dynamic Frame Size Adjustment Strategy for Mobile RFID Readers," *Sensors*, vol. 9, no. 2, pp. 845–858, 2009.

[24] EPCglobal, "Class 1 Generation 2 UHF Air Interface Protocol Standard "Gen 2" Version 1.2.0 Specification," Available online at http://www.epcglobalinc.org/standards/uhfc1g2, October 2008.

[25] ——, "The Application Level Events (ALE) Specification, Version 1.1.1," Available online at http://www.epcglobalinc.org/standards/ale, March 2009.

Nikolaos Konstantinou received his Dipl-Ing degree from the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Greece, in 2004 and his Ph.D. in Electrical and Computer Engineering from NTUA in 2009. Since February 2010, he joined the Autonomic and Grid Computing (AGC) group in Athens Information Technology (AIT). His research interests include knowledge management, context-sensitive systems and information flow in the internet of things. He is a member of the Technical Chamber of Greece.