Performance analysis of anti-collision protocols for RFID systems

Giuseppe Bagnato, Gaia Maselli, Chiara Petrioli, Claudio Vicari
Computer Science Department, “Sapienza” University of Rome - Italy
bagnato_giuseppe@yahoo.it, {maselli, petrioli, vicari}@di.uniroma1.it

Abstract— **Recently RFID technology has made its way into end-user applications, enabling automatic item identification without requiring line of sight. In particular passive tags provide a promising, low cost and energy-efficient solution for inventory applications. However, their large-scale adoption strictly depends on the efficiency of the identification process. A major challenge is how to arbitrate channel access so that all tags are able to answer the reader inquiries and identify themselves over time. This paper stems from the observation that a variety of anti-collision protocols for RFIDs have been proposed in the literature. However, a thorough simulation comparison among them and a clear identification of the mechanisms resulting in better end-to-end performance is lacking. The objective of our work has been to fill this gap. This paper presents the results of a detailed ns2-based comparative evaluation of representatives of all the classes of anti-collision protocols so far proposed. Simulation results show that end-to-end performance of the different classes of protocols in terms of metrics such as the time needed for tags identification differ significantly over what previously found by experiments which only focused on the number of reading cycles for tag identification. Our thorough performance evaluation has highlighted that different solutions are to be used in different application scenarios and that decreasing the collisions (rather than idle times) is the way to go to further improve anti-collision protocols performance.**

I. INTRODUCTION

A basic RFID system consists of a *reader* and a set of *tags*. The reader inquiries tags that are able to communicate on the radio channel, returning their ID. Tags are typically *passive* devices, which answer to reader's query by backscattering the received signal. One of the main objectives of a RFID system is the identification of all the tags present in the area covered by the reader. The challenges related to tags identification depend on the reference scenarios, which may include one or more readers, and a variable or stationary set of tags. The coexistence of multiple readers in the same area may cause collisions among readers interfering with each other or with tags. Moreover, collisions may occur among tags simultaneously transmitting to the same reader, independently of the presence of one or more readers. Collisions are addressed by specific solutions for the multi-reader problem (through frequency allocation mechanisms) and the single-reader problem (through collision arbitration schemes). The other distinguishing factor is given by scenario variability. In case of stationary applications scenarios (i.e., consecutive readings of the same, slightly changed, set of tags), the reader may adapt the reading procedure according to information gathered through sequential identification processes. Instead, for applications involving new or highly variable tag populations (i.e., most of or all tag IDs change in time), each identification is based on a single-reading process, that operates without any knowledge on the environment. In this paper we focus on collision arbitration in *single reader* scenarios, in which it is not possible to apply an adaptive process (that is also the case of the first reading in adaptive protocols). The main representative protocols in this class follow techniques that were studied in the past for the multiple access problem. Specifically, a first group of protocols is inspired by slotted *aloha* protocols [1], while a second group of protocols draw on serial *tree* algorithms [2] (also called walking tree algorithms). In the *aloha-based* protocols, time is slotted and slots are grouped into frames. Frame size varies over time and is communicated by the reader at the beginning of each frame in the query message. Tags then randomly select one slot in the current frame: they will answer the reader query with their ID only when such slot comes. At the end of each frame, identified tags become silent. Only colliding tags go on to the next frame. Protocols in this class differ in the way tags are grouped into frames and tag population is estimated [3][4]. A detailed description of the main representatives of aloha-based protocols is given in Section II. *Tree-based* protocols proceed more deterministically: they iteratively query a subset of tags which match a given property until all tags are identified. These protocols are called tree-based because the identification process can be represented as a tree where the root is the set of tags to be identified, intermediate nodes represent groups of colliding tags answering the same request from the reader, and the leaves correspond to single-tag responses. Tree-based protocols differ in the way tags are queried (e.g., based on a counter stored in the tags, or the binary structure of tag ID's). A detailed description is given in Section II.

Although several anti-collision protocols have been proposed in the literature, an in depth understanding of their relative performance is still lacking. In particular preliminary comparison between different solutions have failed to consider end-to-end performance metrics reflecting important aspects such as the time needed to complete the identification process. There is also the need to have a more thorough understanding of the impact of different parameter settings, tags ID distributions on the relative protocol performance. Goal of this paper is to fill this gap, performing an in-depth analysis

---

This work was partially supported by the MIUR International FIRB RBIN047MH9.

978-1-4244-2517-4/09/$20.00 ©2009 IEEE

of RFID anti-collision protocols, and providing an accurate performance evaluation, which investigates important aspects such as efficiency and latency of tag-reading systems. The scalability of the inventory process is also a crucial factor to analyse, as in many application scenarios the number of tags is not known prior to protocol execution. Besides identifying the best performing protocols, our study demonstrates that metrics so far used to evaluated RFID anti-collision protocols do not accurately capture protocols performance. Specifically, we show how the evaluation of protocol-execution time (measured in seconds) may not confirm results obtained with previously defined metrics, based on reading rounds. [1] The time associated to idle rounds and collision rounds is different so that a properly designed protocol should consider this aspect, favoring idle rounds over collision ones, in order to result in fast tags identification. Our findings therefore open new research perspectives, highlighting the aspects that can be improved in future work.

The paper is organized as follows. Section II presents the main representatives anti-collision protocols. Section III compares their performance under various tag densities and distributions, and draws some conclusions.

II. RFID ANTI-COLLISION PROTOCOLS

In this section we review the major representatives of aloha-based and tree-based anti-collision protocols for RFID networks, and give the theoretical complexity for each protocol. In particular, we analyze the system efficiency, defined as the ratio between the number of tags and the number of rounds needed to identify them.

A. Aloha-based protocols

Protocols in this class consider the channel to be slotted into intervals of time, whose duration is equal to the tag's ID transmission time. The reader issues consecutive groups of slots, or frames, until all tags are identified. The main problem for aloha-based protocol is frame sizing, as the reader does not know the number of tags to be identified. A common method to establish the length of the new frame is to use Chebyshev's inequality [3], exploiting the outcome of the previous frame: information such as the number of idle slots, that of slots where collision occurred, and the number of slots where tags were identified, allow to estimate the tag population and properly dimension the following frames (for further details we point the reader to [3]). Based on such estimation method, protocols in this class differ in the way tags are grouped into frames and tag population is estimated.

1) Enhanced Dynamic Framed Slotted Aloha (EDFSA) [5]: specifies a predefined set of frame sizes (varying from 8 to 256) for different ranges of estimated unread tags. Typically, the frame size is around the mid point of the range. As an example, if the estimated number of unread tags is between 82 and 176, than the selected frame size is 128, while for a range between 177 and 354 tags, the frame size grows to 256. For larger tag populations, EDFSA randomly splits tags into groups of the maximum frame size (i.e., 256 tags). In such a case, only the tags associated with one of the groups are queried in the following frame. Chebyshev's inequality is used at the end of the frame to estimate the number of tags which have participated in it. Such an estimate is then used to refine the estimate of the global tag population, possibly adjusting the number of tag groups and the size of the following frames.

2) Tree Slotted Aloha (TSA) [6]: deals with collisions more efficiently. After the first frame, a new set of frames is allocated, each devoted to solving the collisions which have occurred in a given slot of the first frame. Only the (few) tags which transmitted into that slot participate in the corresponding frame. The approach is repeated: If collisions occur in one of the frames allocated to solve collisions (say, frame i), new frames are allocated to solve such collisions (one for each collision slot in frame i). This is possible by estimating the number of tags colliding in each slot, and then allocating a properly sized frame to solve the collisions which have occurred in such slot. The cost of the TSA protocol can be estimated by referring to random hash trees [7] (the tree built by the TSA protocol is a random hash tree, if we know the number of tags participating to the protocol). The size of a random hash tree corresponds to the number of rounds required by TSA protocol to identify n tags, and has been estimated as $2.3020238n$. Hence, TSA system efficiency can be estimated as $n/2.3020238n$ (i.e., 43%).

B. Tree-based protocols

In these protocols the reader iteratively queries a subset of tags matching a given property until all tags are identified. Protocols differ in the way tags are queried: according to a binary random number in Binary Splitting[8], and tags ID in Query Tree[9].

1) Binary Splitting (BS) [8]: recursively splits answering tags into two sub-groups until obtaining single-tag groups. Each tag maintains a counter (initially set to zero). Tags with the counter value equal to zero answer the reader query, while others remain silent until their counter decreases to zero. The value of the tag counter is modified depending on whether the query results in a collision, identification, or no-answer. In case of collision, colliding tags add a random binary value to their counter, i.e., they are split into two subsets: those whose counter value is zero and those whose counter value is one. The tags which were not involved in the collisions increase their counters by one. In case of identification or no answer, all the tags decrease their counter by one, so that those with the counter at one will answer the next query. It has been shown that the expected number of queries EQ needed to identify all tags is $2.881n - 1 \leq EQ \leq 2.887n - 1$, and hence BS system efficiency is about 34% [1].

2) Query Tree (QT) [9]: queries tags according to their ID. The reader interrogates tags sending a string, and only those tags whose IDs have a prefix matching that string respond to the query. At the beginning, the reader queries all tags (sending NULL string). If a collision occurs, then the string length is

---

[1] A round is intended as a reader request (e.g query or timeslot signaling) and relative tag response.

increased of one bit until the collision is solved and a tag is identified. The reader then starts a new query with a different string. In particular if tag identification occurred with a string $q0$ the reader will query for string $q1$. The resulting binary tree has nodes at the $i-$th level labeled with all the possible values of a prefix of length i (e.g. nodes at level 1 contain prefixes 0 and 1, nodes at level 2 prefixes 00, 01, 10, 11 and so on). The exploration of a subtree is skipped in case there is only one tag matching the prefix stored in the subtree root (i.e. if a tag identification occurs when the reader queries with the subtree root prefix). In case of uniform ID distribution, the tree induced by the query tree is analogous to the tree induced by the BS protocol. This is because a set of uniformly distributed tags splits approximately in equal parts at each query, like in the BS protocol. Hence we can assert that QT system efficiency is about 34%.

3) Query Tree Improved (QTI) [9]: is an enhancement of the QT protocol that optimizes the number of queries, avoiding those that will certainly produce collisions. As an example, consider the case in which prefix "p" produces a collision, while prefix "$p0''$ results in no tag answers. in QTI the reader will then skip prefix "$p1''$ that will certainly produce collision and query directly for "$p10''$ and "$p11''$. The expected number of queries EQ needed by QTI to identify all tags has been estimated as $2.6607n \leq EQ \leq 2.665n - 1$ [1]. Consequently, the QTI system efficiency is around 37%.

III. PERFORMANCE EVALUATION

This section reports the results of a thorough ns2 based comparative performance evaluation among all the major schemes so far proposed for single reader–tag communications. We have implemented an RFID extension of the Network Simulator $ns2$ (v. 2.30) [10] accounting for all the unique features of reader-tag communications. Using our extension we have simulated the QT, QTI, BS, TSA and EDFSA protocols. Our simulative results are in line with theoretical results discussed in Section II for what concerns system efficiency.

A. Metrics

Protocols are compared by evaluating the following metrics.

Latency It is defined as the effective duration of protocol execution in seconds.

System efficiency In terms of rounds (e.g., query or timeslot signaling) the system efficiency is measured as $SE_r = R_{id}/R_{tot}$, where R_{id} is the amount of identification rounds (which is equal to the number of tags), and R_{tot} is the total number of rounds. In terms of time, the system efficiency translates to $SE_t = T_{id}/T_{tot}$, where T_{id} is the time taken by identification rounds, and T_{tot} is the execution time.

B. Transmission time model

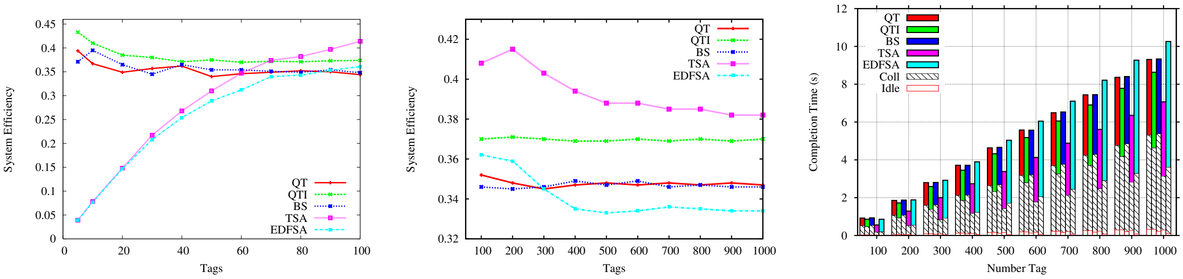
To perform a realistic evaluation of protocols temporal aspects, we refer to the transmission model defined by the EPCglobal Specification Class-1 Gen-2 [11], that specifies time requirements for reader-to-tag and tag-to-reader communications, for a passive-backscatter, interrogator-talks-first system. In the specified link timing, whenever the reader sends a message to tags (i.e., query or timeslot signaling), the reader transmission arrives to tags after a propagation delay (depending on the transmission range and typically fixed around a fraction of $1\mu s$) and lasts TX_I time (depending on datarate). After reception, tags need a $R1$ time to react and send back an answer. The reaction time $R1$ lasts from the end of reception of interrogator transmission, to the start of transmission of tag's response, and is fixed to $10 * T1$ (where $T1 = 1/datarate$) according to EPC specification [11]. Again a propagation delay is needed before the reader starts receiving tag responses, that will last TX_R time. After receiving tag responses, the reader needs a reaction time $R2$ before being able to issue a new command (new query or timeslot signaling). Analogously to $R1$, the time $R2$ is defined for the reader side, lasting from the end of reception of tag's transmission to the start of the new reader transmission, and is fixed equal to $T1$. In case the reader does not get any response (idle round), the reader realizes that no transmission is coming back from tags after a RX threshold that is the time at which the reader should receive the first bit of tag transmissions. This means that in case of no response by the tags, the round ends when the RX threshold elapses, and the reader issues a new command after a reaction time $R2$. The model highlights that the duration of a round strictly depends on the amount of bits transmitted in the reader's message and tags' response. As a consequence, the duration of an idle round is shorter than the time taken by an identification or collision round, that involves the transmission of tag's ID.

C. Scenarios

We consider an RFID system where a single reader communicate with $n = 10, ..., 1000$ tags. The channel data rate is 40 Kbps, reader–tags communications occur at a frequency 866 Mhz as specified by the EPCglobal standard [11]. Tag IDs have length equal to $k = 96$ bits (the most common value). The initial frame size for aloha-based protocols is set to 128 slots, according to previous studies [6]. In terms of distribution of tags IDs we have first considered tags with uniformly distributed IDs. Then we have studied the impact of specific IDs distributions on the performance of query-tree protocols, which are sensible to changes in the tags IDs distribution. Results have been obtained by averaging over 100 runs.

D. Results with uniform distribution of tag IDs

Results on the protocols system efficiency and time system efficiency are reported in Figures 1(a), 1(b), 2(a) and 2(b) for both sparse ($n = 10, ..., 100$) and dense ($n = 100, ..., 1000$) networks. In terms of system efficiency tree-based protocols significantly outperform aloha-based schemes when $n \leq 60$. The trend changes in denser networks ($n = 100$ or higher) where TSA is by far the best performing protocol, while tree-based protocols experience a similar system efficiency (around



(a) System efficiency: 10-100 tags.

(b) System efficiency: 100-1000 tags.

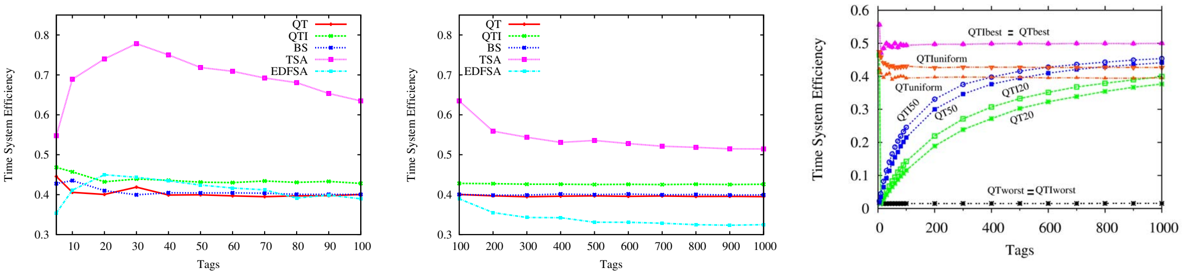
(c) Protocol execution time.

Fig. 1. Round-based system efficiency and protocols latency.

34%), independently of n. The number of deployed tags has instead a strong impact on the performance of aloha-based anti-collision schemes. In such schemes system efficiency is extremely low (as low as 4%) when only a few tags are deployed. It then steadily increases with n till a peak value (which occurs at $n = 100$ in EDFSA and at $n = 200$ in TSA). After this peak, system efficiency first quickly degrades and then almost flattens (at $n = 4 - 500$). The rationale behind these different trends is quite clear. Since the number of tags is not known, it is challenging to properly tune the frame size in aloha-based protocols. Based on previous results [6] we have set the initial frame size to 128 bits. Such frame size is over-estimated for small number of tags, resulting in low system efficiency at low n. Tree-based protocols instead only need a few collision rounds to identify all tags when n is small. This motivates tree-based anti-collision protocols superior system efficiency for $n \leq 60$. Out of the tree-based protocols, QTI highlights its best performance (and confirms theoretical results), maintaining an average efficiency around 37% independently of n. QT and BS also performs as estimated by theoretical analysis, experiencing similar, slightly worse, performance (around 35%). When we increase the number of tags, TSA and EDFSA performance improves, as the number of idle slots reduces. TSA and EDFSA achieve a peak of system efficiency when n is between 100 and 200, which is when these protocols make the best use of the initial frame (and hence protocols performance converges to the upper bound). In fact, many tags are successfully identified during the first frame and possible collisions incurring during such frame are quickly solved in a few (short) additional frames which follow. TSA peak system efficiency is quite high (42% against a theoretical estimation of 43%). TSA also performs quite well for denser networks, as system efficiency never drops below 38% for $n \leq 1000$. EDFSA performance is not as good. System efficiency never tops $36,5\%$ and quickly falls below 34% when n increases beyond 400. This is due to the fact that when the tag population increases, EDFSA is not as efficient as TSA in solving collisions, as it remixes all tags at each new frame.

Protocols performance change considerably if we switch our attention from the system efficiency, which is an indirect measure of the effectiveness of the identification process, to the protocol execution time, which instead directly reflects performance experienced by the user (see Figure 1(c)). When focusing on latency, TSA is consistently faster than any other protocol, for all the considered n. All other protocols performance is comparable when the number of tags is below 400. When n further increases EDFSA performance slightly degrade with respect to the tree-based protocols. This is quite unexpected and different from the protocols relative performance predicted by previous works which focused only on system efficiency. The problem is that system efficiency does not account for the different time a protocol needs to complete idle or collision/identification rounds. Also, the same type of round may have different durations in different protocols (the number of per round transmitted bits is protocol-dependent). Therefore considering only the number of identification rounds over the overall number of rounds (i.e., the system efficiency) does not provide an accurate estimate of how effectively and fast tags will be identified. Insights on the reasons behind protocols different execution times is provided by results shown in Figure 1(c). The bars in the figure show the execution time of each protocol, displaying with different colors the time spent performing idle rounds (lower white part of the bar), the time spent performing collision rounds (grey mid section of the bar) and finally the time spent for identification rounds (upper section of the bar, colored with different colors depending on the protocol). Idle rounds have a negligible impact on overall protocol execution time. This has a twofold reason: collision rounds tend to occur more frequently than idle rounds, and idle rounds have a shorter duration. Aloha-based protocols experience much less collisions that tree-based protocols, which motivates good TSA performance.

Time system efficiency provides a useful metric that complements latency. More than a way to fairly compare different protocols, time system efficiency is able to provide insights of the inefficiency (and possible margins of improvement) of a given class of anti-collision protocols. It gives a clear idea of how much time is devoted to tag identification and how much is wasted in collisions or in idle rounds. Results are displayed in Figure 2(a) and Figure 2(b). They show that TSA very effectively reduces the time spent performing idle and collision rounds, resulting in a time system efficiency which is over 55% (with a peak at 78%) for low density networks. TSA time system efficiency remains quite remarkable (between 52% and 64%) also at high density. All other protocols have



(a) Time system efficiency: 10-100 tags.

(b) Time system efficiency: 100-1000 tags.

(c) Time system efficiency for different tag ID distributions in QT and QTI protocols.

Fig. 2. Time system efficiency.

a much lower time system efficiency (the gap with TSA is 100% for low density scenarios and as high as 50% at high density). Tree-based protocols suffer from many more collisions. EDFSA results in longer identification times, due to overhead introduced by an explicit mechanism to mute tags after their identification.

E. Results with optimal distribution of tag IDs

We now identify the best IDs distribution, that maximizes the performance of query-tree protocols, reducing the number of queries needed to identify all tags to $2n - 1$ (while for uniform ID distribution is $2.881n - 1 \leq EQ \leq 2.887n - 1$ [9]). The idea is to minimize the number of collisions. Let us consider the identification process of two tags: if their IDs differ for the most significant bit (e.g., $< \mathbf{0}0000 >$, and $< \mathbf{1}0000 >$) the inventory will result in only one collision (which is the minimum number of collisions to identify two tags). Such collision occurs when the reader queries for the NULL prefix; both tags will be identified when the reader queries for the 1-bit prefix. Extending the reasoning to n tags, we can label them by making different the first $log_2 n$ most significant bits, for example by assigning to the $i - th$ tag ($0 \leq i \leq n - 1$) the unsigned integer representation of i in the first ($log_2 n$) most significant bits. The remaining least significant bits of its ID can be given any value. In this way, common prefixes of tag IDs are as short as possible, and collisions are reduced to the first $log_2 n$ bits. Thus, the number of queries needed for the inventory is minimized (i.e., $2n - 1$). The worst ID distribution instead is characterized by IDs having the longest common prefixes. If we consider two tags differing for the least significant bit (e.g., $< 0000\mathbf{0} >$, and $< 0000\mathbf{1} >$), then tags are sibling leaves in the identification tree, and the inventory will result in as many collisions as the common (prefix) bits in the IDs. Figure 2(c) shows the time system efficiency of query-tree protocols for different ID distributions. Also the cases in which tags IDs are clustered in groups of h consequent IDs [6] (displayed in Figure 2(c) as QT_h) have been simulated. When the worst case IDs distribution is applied, system efficiency drops close to 0%, while in case of the best distribution, protocols time system efficiency is nearly 50%, becoming comparable to TSA system efficiency. Investigating protocol latencies we found that QTI (best case) and TSA employ a comparable amount of time to identify all tags, especially for dense networks. Another important aspect to be noticed is that such distributions nullify the improvements obtained by QTI, as the number of collisions is minimum, and hence there are no branches that can be pruned by QTI.

F. Concluding remarks

Our study highlights the importance of a realistic (temporal) performance evaluation, and suggests that in case of application scenarios that allow to write tag IDs, QTI may be the solution of choice, as it combines good performance, scalability, fast and easy implementation (the protocol is memoryless, since the set of tags answering a query only depends on the current prefix included in the query, not on the past history). On the other hand, the TSA protocol requires a more complex tag device (able to maintain state information regarding frames and slots), but it is more efficient whenever it is not possible to use a specific ID distribution.

REFERENCES

[1] J. Massey, "Collision resolution algorithms and random-access communication," *Multi-Users Communication Networks, CISM Courses and Lectures*, no. 256, pp. 73–137, 1981.

[2] J. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 505 – 515, 1979.

[3] H. Vogt, "Efficient object identification with passive rfid tags," in *Pervasive '02*. London, UK: Springer-Verlag, 2002.

[4] M. Bolic, M. Latteux, and D. Simplot-Ryl, "Framed aloha based anti-collision protocol for rfid tags," in *Proceedings of ACM SenseID 2007*, Sydney, Australia, 2007.

[5] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhanced dynamic framed slotted aloha algorithm for rfid tag identification," in *MOBIQUITOUS '05*, Washington, DC, USA, 2005, pp. 166–174.

[6] M. Bonuccelli, F. Lonetti, and F. Martelli, "Instant collision resolution for tag identification in RFID networks," *Elsevier, Ad Hoc Networks*, no. 5, pp. 1220–1232, 2007.

[7] L. Devroye, "The height and size of random hash trees and random pebbled hash trees," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1215–1224, 1999.

[8] J. Myung and W. Lee, "Adaptive binary splitting: a rfid tag collision arbitration protocol for tag identification," *Mob. Netw. Appl.*, vol. 11, no. 5, pp. 711–722, 2006.

[9] C. Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification," in *ACM DIALM '00*, New York, USA, 2000.

[10] "The Network Simulator - ns-2," http://www.isi.edu/nsnam/ns/.

[11] "EPC Radio-Frequency Identification Protocols Class-1 Gen-2 UHF RFID Protocol for Communications at 860MHz-960MHz, EPCglobal," http://www.epcglobalinc.org/standards/uhfc1g2, Dec. 2004.