Tree Slotted Aloha: a New Protocol for Tag Identification in RFID Networks

Maurizio A. Bonuccelli and Francesca Lonetti*
Dipartimento di Informatica, Università di Pisa
56100 Pisa, Italy
{bonucce,lonetti}@di.unipi.it

Francesca Martelli
ISTI "A. Faedo", Area della Ricerca CNR
56100 Pisa, Italy
f.martelli@isti.cnr.it

Abstract

In this paper, we approach the problem of identifying a set of objects in an RFID network. We propose a modified version of Slotted Aloha protocol to reduce the number of transmission collisions. All tags select a slot to transmit their ID by generating a random number. If there is a collision in a slot, the reader broadcasts the next identification request only to tags which collided in that slot. Simulation results show that our approach performs better than Framed Slotted Aloha and query tree based protocols, in terms of number of slots needed to identify all tags, which is a commonly used metric, strictly related to delay.

1 Introduction

Automatic object identification is very useful in many (old and new) applications. For instance, fast and reliable reading of labels (tags) attached to different objects stowed in warehouses can greatly speed up operations such as localization and retrieval. Among the countless applications, we can cite public transportation and ticketing, access control, production control, checkout speed-up in shops, secure operations in dangerous environments, localization of objects (like cars in parking lots, or books in libraries) and people. Radio frequency identification (RFID) systems offer a promisingly affordable, cheap and flexible solution for object identification [1]. An RFID system consists of radio frequency (RF) *tags* attached to the objects that need to be identified, and one (or more) networked electromagnetic *reader*. The reader is an entity with great computation power and memory, while tags have (very) limited computational resources.

In these systems there is a single communication channel, and messages are exchanged between reader and tags, which are not able to communicate each other. Typically, the reader broadcasts a message to tags, which, in case, send back an answer. If many tags simultaneously answer, a *collision* occurs, i.e. their transmissions will merge in a meaningless message, and the reader is not able to capture any answer: it can only realize that more than one tag is communicating. Instead, if only one tag at a time transmits its ID, the reader can identify it. Tags can be *active* or *passive*, according to their technology. Active tags are provided with batteries that continuously power the computing and transmitting circuits. Passive tags, instead, rely only on the RF energy induced by electromagnetic waves emitted by the reader. The energy induced by the reader is very low, so it is desirable that tags implement very simple computation procedures, so reducing as much as possible their power consumption. This is also useful for reducing tags cost.

As outlined above, concurrent tags transmissions result in a collision, that is, the reader receives a mixture of scattered waves and is not able to identify any single signal. Therefore, a mechanism for limiting and solving collisions is required. As in all multiple access systems, there are two families of protocols for approaching the contention resolution problem: probabilistic protocols, and deterministic ones. We are interested in probabilistic protocols, and in this paper we show an Aloha-based protocol which reduces the number of "recollisions", by solving the collisions as soon as they happen.

The paper is organized as follows. In Section 2, we briefly survey the prominent protocols proposed for the above problem, focusing on probabilistic protocols. In Section 3, we describe our approach, by highlighting the assumptions we made. In Section 4 we report the results of simulation experiments performed to get insights into our protocol performance. Conclusions and open problems (Section 5) terminate the paper.

2 Related work

In RFID networks, a reader recognizes objects through wireless communications with tags, and it must be able to identify all tags as quickly as possible.

*Maurizio Bonuccelli and Francesca Lonetti are also at ISTI-CNR.

Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)
0-7695-2593-8/06 $20.00 © 2006 IEEE

The problem approached in this paper is the tag collision one, that occurs when several tags try to answer at the same time to a reader query. The reader queries the tags for their ID by broadcasting a request message. Upon receiving such message, all tags send an answer back to the reader. If only one tag answers, the reader identifies the tag. If more than one tag answer, their messages will collide on the RF communication channel, and the reader cannot identify these tags. This is a special case of the medium access control problem, and has been extensively studied very recently [2, 3, 4, 5, 9]. In [6], a similar problem has been studied. In that paper, tags are assumed to have collision detection capability, which (greatly) helps in solving the MAC problem, at the cost of additional expensive and power consuming hardware in tags. Besides, in [6] it is also assumed that the reader knows the exact number of tags to be identified, an assumption quite uncommon in practice. Actual RFID applications introduce a more challenging side to the MAC problem. Given the low functional power and energy constraints in each tag, it is unrealistic to assume that tags can communicate with each other directly, and that they can notice their neighboring tags or detect collisions.

Tags anti-collision protocols can be deterministic or probabilistic. Among the former, there are the so called *tree-based* protocols [2, 3, 9, 10], and the latter are well represented by *Aloha-based* ones [4, 5, 7].

Tree-based tag anti-collision protocols do not cause collisions, though they have a relatively long identification delay. The first work on tree search in order to solve the multiple access control problem of independent sources was presented in [8]. Among tree-based protocols there are *binary tree* protocols [10], and *query tree* protocols [3]. The features of these protocols are described next.

In the *query tree* (QT) protocol [3] the reader asks the tags to answer if their ID matches a given prefix. If there is a collision, the reader queries for one bit longer prefix until no collision occurs. Once a tag is identified, the reader starts a new round of queries with another prefix. In [3], many improvements of QT protocol have been presented, for reducing its running time. Among them, the most important is that one that we call Query Tree Improved (QTI) [3], that avoids the queries that certainly will produce collisions. Suppose that a query of prefix "q" results in a collision and the query of prefix "$q0$", results in a empty slot, then the reader skips the query prefix "$q1$" and does only the queries "$q10$" and "$q11$".

Aloha-based protocols reduce the probability of occurrence of tag collisions, since tags answer at distinct time. In *pure Aloha* protocol, tags randomly select their arbitrary response time, while in *Slotted Aloha* [7], tags can answer at the beginning of a randomly selected timeslot, to avoid partially overlapping transmissions. *Framed Slotted Aloha* [11] configures a frame with contiguous timeslots, and a tag transmits its ID only once in a frame. In [12], it is showed that classical Aloha-based multiple access protocols, namely pure and slotted Aloha, are either not energy-conserving or lead to unacceptable delays.

In [4], a variant of framed Aloha with variable frame size is presented. Tags are randomly allocated to slots in this way: at the beginning of a frame, the reader sends the frame size and a random value to the tags. Based on these values, each tag randomly chooses the timeslot during which it will transmit its ID. In the same paper, a way to set the proper frame size for each cycle, based on an estimation of the number of unidentified tags is also given. This estimation is made by using Chebyshev's inequality[1], which estimates how the real number of tags and its expected value are far. Then, the frame size is adjusted so that this difference becomes minimal. Performance is better than that of basic Framed Slotted Aloha. However, this protocol suffers the problem that the frame size is upper bounded (usually by 256), and cannot be increased at will, as the number of tags increases, thus leading to a very high number of collisions when the number of tags is larger than the upper bound.

This maximum frame size problem is solved in [5] (we call the protocol presented there AFSA) where, by introducing a *modulo* operation, it is possible to limit the number of tags transmitting in the same maximum size frame. When the reader realizes that the number of unidentified tags is larger than the maximum frame size, it sets the number of tag groups which maximizes the global system efficiency function. This function is given by the ratio between the number of slots filled with one tag and the current frame size [5]. The reader then broadcasts the number of tag groups and a random value. This last value is used by the tags as a parameter (together with the serial number) to compute another random number that is divided by the number of groups. Only tags having remainder zero can transmit in the current frame. The performance, in terms of system efficiency, namely the ratio between the number of tags and the number of time slots needed to identify all of them, is maintained between 34.6% and 36.8%, [5].

Our protocol takes a different approach and so it is able to limit the number of collisions, leading to a global average system efficiency up to 43.44%.

3 Tree Slotted Aloha protocol

In this section, we present our proposal for tag identification which we call Tree Slotted Aloha (TSA) protocol.

The basic idea of our TSA identification protocol, is to solve a collision as soon as it happens. In Framed Slotted Aloha protocols, two tags not colliding in a frame, can collide in the next frame. In our approach, the above situation

[1] We use the same estimation function, but in a different way, so we postpone its definition to Section 3.

Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)
0-7695-2593-8/06 $20.00 © 2006 IEEE

is avoided, since when a collision occurs in a slot, only the tags that generate such collision are queried in the next read cycle. This results in a better performance.

We consider an RFID system consisting of a reader and a set of n passive tags. Each tag $t \in \{0, ..., n-1\}$ has a unique ID string $t_{id} \in \{0,1\}^k$, where k is the length of the ID strings. We assume that the reader does not know the exact number n of tags present in its communication range, but it can estimate it. Such an estimation l_0 is the starting frame size. In this paper, we show that such initial estimation does not affect the performance in considerable manner: our protocol has a good efficiency even if the initial estimation is far from the actual number of tags in the reader communication range, as shown in Section 4.

The protocol is performed in several tag reading cycles. A reading cycle consists of two steps: in the first step, the reader broadcasts a request for data by specifying the frame size l_i, in the second step each tag in the communication range of the reader, selects its response slot by generating a random number in the range $[1, ..., l_i]$ and transmits its ID in such a slot. The reader identifies a tag when it receives the tag ID without collisions. The behavior of the protocol follows a tree structure. The root node is the frame in the first reading cycle. Let l_0 be the size of such a frame, N_i being the number of tags transmitting their ID in slot i, with $i \leq l_0$, $N_i \geq 0$, $\sum_i N_i = n$. If $N_i \geq 2$, there is a collision in slot i. At the end of each reading cycle, if the reader realizes that collisions occurred, it starts a new reading cycle for each slot where there was a collision. This corresponds to adding new nodes in the tree, as sons of the node representing the above reading cycle, one son for each slot with collisions. The size of such new cycles is defined as described later, and the reader broadcasts such a size, together with the slot number of the previous frame (to address only the tags colliding in that slot), and the level of the tree. In each reading cycle, tags store the generated random number (i.e. the slot in which they transmitted their ID) and increase by one their own tree level counter, so that they can realize when are involved in later communications. Obviously, in each new reading cycle, collisions can occur. Each time a collision is sensed, a new node (son of the node representing the previous cycle) is inserted in the tree, and another reading cycle is started. The whole process is recursively repeated until no collisions are detected in a cycle. The reader and tags procedures are shown in Table 1, and an example of protocol execution is shown in Figure 1.

Like in Framed Slotted Aloha, TSA is not memoryless, since each tag has to remember the random number generated in the previous cycle, and the level of the tree, as said before. Notice that the amount of memory needed by each tag is very small, namely few bits to remember the tree level and the slot number of the previous reading cycle. In Section 4, we show that TSA performs better than AFSA

Table 1. Reader and Tag procedures in TSA.

Reader procedure:
$level = 0$;
$l_0 =$ initialEstimation;
$s = -1$;
collisionResolution(level,s,l_0*)*.

collisionResolution(level,slot,l_i*)*:
broadcast($level$,$slot$,l_i);
for $s = 1$ **to** l_i **do** receiveIDs;
$l_{i+1} =$*tagsEstimation(*l_i*)*;
for $s = 1$ **to** l_i **do**
if (collision[s]$= 1$) **then**
identify tag and sendAck;
else if (collision[s]> 1) **then**
$level++$;
collisionResolution(level,s,l_{i+1}*)*;

int tagsEstimation(l_i*)*:
compute c_0, c_1, c_k, obtained in frame of length l_i;
compute tag number estimation n_i by using equation (1);
$n_i = n_i - c_1$;
return $\left\lfloor \frac{n_i}{c_k} \right\rfloor$;

Tag procedure:
$identified = false$;
$myLevel = 0$;
$previousValue = -1$;
while (**not** $identified$) **do**
receive($level$,$slot$,l_i);
if (($level = myLevel$) $\wedge$ ($previousValue = slot$)) **then**
$s =$ randomNumber $mod\ l_i$;
$myLevel++$;
$previousValue = s$;
send $myID$ in slot s;
if ($receivedAck$) **then** $identified = true$;

in terms of number of slots needed to identify all tags. As we shall see, our experimental outcomes are confirmed by the analytical result presented in [13] about the average size of a random hash tree, which is asymptotic to $2.3020238n$, where n is the number of uniformly distributed random entries to be placed in the tree. The tree built by TSA protocol is identical to such random hash tree, if we would know the exact number of tags. So, if this is the case, we can assert that the average number of slots needed to identify n tags is $2.3020238n$. In other words, if we define the system efficiency as the ratio between the number of tags and the number of slots needed to identify them like in [5], it results that the average system efficiency of TSA protocol is given by $n/2.3020238n = 0.4344$. In the same paper [13], the average height of a random hash tree is also shown. Such an height is proved to grow as $\lg_2 n$, in probability, with uniform distribution. So, in TSA the amount of bits needed to represent the tree level is equal to $\lg_2 \lg_2 n$. Such amount is actually very small: for instance, up to $n = 65536$, only $\lg_2 \lg_2 2^{16} = \lg_2 16 = 4$ bits are needed to represent the

Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)
0-7695-2593-8/06 $20.00 © 2006 IEEE

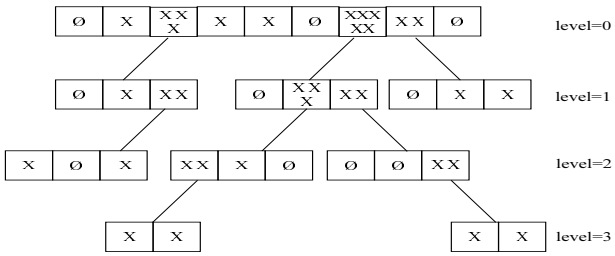

Figure 1. Example of TSA protocol execution

level of TSA tree (in addition to that ones needed to store the slot number).

The size of the frame in each reading cycle is computed by using a particular estimation function, similar to that used in [4], and defined as follows. At each reading cycle, we obtain a triple $< c_0, c_1, c_k >$ quantifying the empty slots, slots filled with one tag and slots with collisions, respectively. We use Chebyshev's inequality asserting that the outcome of a random experiment involving a random variable X, is most likely somewhere near the expected value of X. We use this property to compute the distance between the effective result $< c_0, c_1, c_k >$ and the expected result $< a_0, a_1, a_k >$ of a reading cycle. By minimizing such a distance, defined in equation (1), it is possible to estimate the number n of tags transmitting in such a cycle.

$$\epsilon(N, c_0, c_1, c_k) = \min_n \left| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a_{\geq 2}^{N,n} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right| \quad (1)$$

The triple $< a_0, a_1, a_k >$ entries indicate the expected number of empty slots, slots filled with one tag, and slots with collision, respectively. N and n denote the frame size in the reading cycle and the number of tags, respectively. When the reader uses a frame size equal to N, and the number of responding tags is n, the expected value of the number of slots with r responding tags is given by

$$a_r^{N,n} = N \times \binom{n}{r} \left(\frac{1}{N}\right)^r \left(1 - \frac{1}{N}\right)^{n-r}.$$

We assume that the ϵ value in equation (1) is searched by varying n in the range $[c_1 + 2c_k, ..., 2(c_1 + 2c_k)]$, where $c_1 + 2c_k$ represents the minimum possible value of n. That is, since c_1 tags have been identified, and if there are c_k collisions, *at least* $2c_k$ tags collided [4]. We set the upper bound of the range equal to $2(c_1+2c_k)$, since by simulation we saw that there is no further accuracy in the estimation, if we set it to higher values. Our tag estimation function computes the frame size l_{i+1} of reading cycle $i+1$ as

$$l_{i+1} = \left\lfloor \frac{n^i - c_1^i}{c_k^i} \right\rfloor$$

where n^i, c_1^i and c_k^i are the estimation, the number of identified tags, and the number of slots with collisions related to reading cycle i respectively. In our tag estimation function, we assume that if n^i is the number of transmitting tags in reading cycle i, the number of tags transmitting in reading cycle $i+1$ will be that one, minus the number of identified tags (c_1^i) in reading cycle i. Besides, since we assume uniform distribution of collisions, the number of unread tags in cycle $i+1$ will be the estimated number of colliding tags in cycle i divided by the number of slots with collision c_k^i. This function is applied for each reading cycle.

Tag identification protocols are usually compared according to two main performance metrics: *time complexity* and *bit complexity*. The former is the number of slots issued by the reader for identifying all tags. The latter is the amount of transmitted bits by the reader and/or by the tags, and represents the energy spent in communication. We analyze more in depth the time complexity, since this is the most used metric. Besides, as we will show in the next section, by reducing the total number of collisions, our protocol reduces also the bit complexity (with respect to the Framed Slotted Aloha protocol). Notice that, since in TSA tags store the number of the last slot in which they transmitted, it is possible to reduce the bit complexity by changing the protocol in this way: instead of always sending the whole serial ID number, tags can answer the reader by sending only one bit in each (micro)slot, and later the reader can query only the not colliding tags by asking for their ID. This is true if we assume that the reader can detect collisions by signal strength inspection.

4 Simulations

We implemented the Tree Slotted Aloha (TSA) protocol described before, together with Advanced Framed Slotted Aloha (AFSA) [5], Query Tree (QT) [3], and Query Tree Improved (QTI) [3]. We implemented also a TSA version, called Limited Tree Slotted Aloha (LTSA), which differs from basic TSA in this: after computing the expected number of tags transmitting in the previous reading cycle with the estimation function ϵ presented in Section 3, LTSA adjusts the frame size according to the values given in Table 2. The same frame adjustment is done also by AFSA, as it is reported in [5]. We implemented also another Framed Slotted Aloha, where frame size does not have limitations: in other words, the frame size is exactly equal to the estimation $\epsilon - c_1$, where ϵ is obtained by means of equation (1). We call this protocol Dynamic Framed Slotted Aloha (DFSA).

We considered the following simulation parameters: the number n of tags in the reader communication range; the initial frame size l_0; the length t_{id} of ID strings, which implies the total number of tags (the "universe" size is $2^{t_{id}}$); the tag IDs distribution: the uniform one, and another dis-

Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)
0-7695-2593-8/06 $20.00 © 2006 IEEE

Table 2. Frame size in LTSA and AFSA.

Estimated unread tags (N_i)	Frame size (l_i)	Groups
1 - 11	8	1
12 - 19	16	1
20 - 40	32	1
41 - 81	64	1
82 - 176	128	1
177 - 354	256	1
355 - 707	256	2
708 - 1416	256	4
...	...	...

tribution where groups of tag IDs are consecutive. In some applications, a distribution of tags IDs in groups, with consecutive numbered IDs in each group, is more realistic than purely uniform distribution. Consider, for instance, a shop inventory: a block of tags are initially (bought and) attached to all objects present at that moment. Very likely, such tags will have consecutive IDs, since have been produced in sequence. Then, some objects will be sold, and some new will be inserted in the RFID system, thus creating groups of blocks.

We implemented the protocols in C language, compiled with *gcc* tool; all tests ran on Pentium III, 800 Mhz, 256 RAM, with Fedora Linux distribution as operating system. Each test was defined by tuning the above cited parameters. For each test, we ran 100 cases by randomly generating the IDs of tags to be identified. The definition of one ID is done by composing a string of randomly generated bits. In case of IDs distributed in groups, the maximum group size g was set equal to 10% of the number of tags n to be identified, and it was used in the ID generation in the following way. Together with the beginning of a group, which was uniformly generated, the size was set by generating another integer random number uniformly distributed in the range $[0..g-1]$. The group definition procedure checked also possible overlappings with already generated groups. In such a case, another ID for a group beginning was generated, until no overlapping was achieved.

As performance measure, we use the time complexity represented by the total number of slots required to identify all tags. For QT and QTI protocols, we consider the number of queries needed to identify all tags, and we assumed that the time spent in one query is equivalent to one time slot. We represent the time complexity by means of the *system efficiency* defined as the ratio between n and the total number of slots required to identify all tags. We compared the performances of TSA and LTSA, with those of AFSA, DFSA, QT and QTI. The results of our simulations are presented in Figures 2 and 3, by varying n value from 5 to 100 and from 100 to 3000, respectively, when IDs are uniformly distributed. As we can see, our protocol performs better than the others when the number n of tags to be identified exceeds 50. For a smaller number of tags, query tree protocols are faster in identifying the tags. In our opinion, this is due to the choice of initial frame size in Slotted Aloha based protocols. In fact, in these test cases, the frame size was initially set to 128, which is too large for small number of tags. As a result, we have at least 128 slots even if n is very small. We performed a set of simulation tests in which the initial frame size l_0 was let to vary from 2 to 256. In Figure 4 we can see that the choice of initial frame size does not significantly affect the performance of TSA, when the number of tags is far from l_0. The best performance is obtained when l_0 is close to the number n of tags to be identified. The same outcome is turned out also for the other protocols.

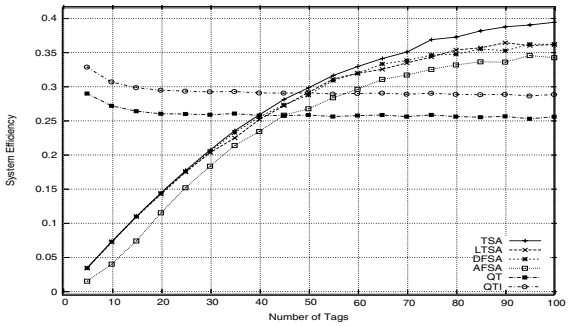

Figure 2. System efficiency vs. Number of tags n, $5 \leq n \leq 100$, $l_0 = 128$, $t_{id} = 48$.

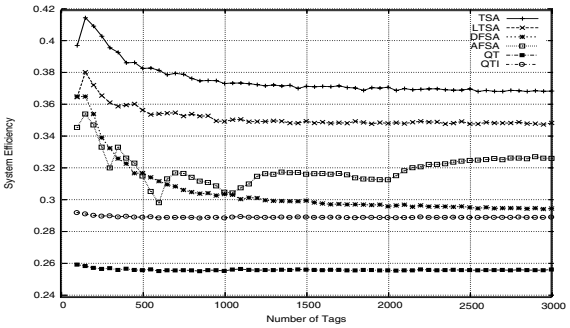

Figure 3. System efficiency vs. Number of tags n, $100 \leq n \leq 3000$, $l_0 = 128$, $t_{id} = 48$.

Figure 5 shows the system efficiency of TSA compared to query tree protocols, when IDs are distributed in groups, as previously described. In such a case, TSA, for its nature, is not influenced by the IDs distribution, while it happens for QT and QTI protocols. This is because query tree protocols need to execute prefix queries until tag IDs are different, and for consecutive IDs this happens only at the last bit. For this phenomenon, the performance of QT and QTI degrades also by increasing the ID length.

Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)
0-7695-2593-8/06 $20.00 © 2006 **IEEE**

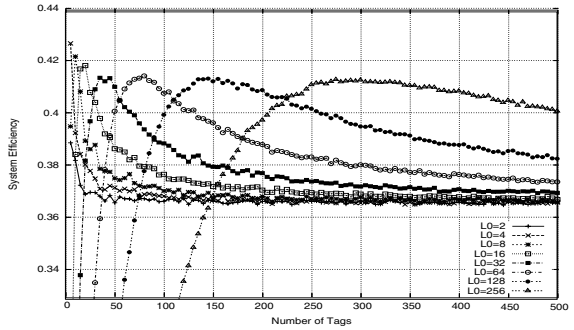

Figure 4. System efficiency of TSA, by varying the initial frame size l_0.

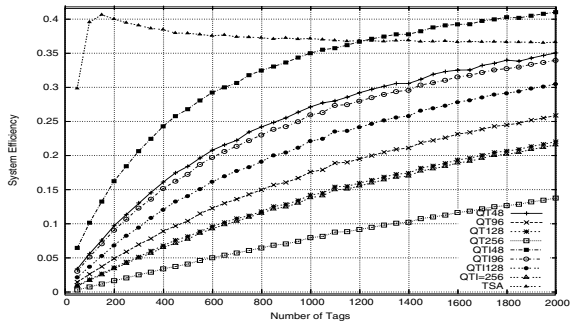

Figure 5. System efficiency of QT and QTI vs. ID length, IDs distributed in groups.

5 Conclusions

In this paper, we investigated the tag identification problem in RFID systems. Firstly, we surveyed the existing proposals for this problem, in particular the probabilistic protocols. Then, we proposed a new probabilistic protocol based on a modified version of Slotted Aloha protocol to reduce the number of transmission collisions. All tags select a slot to transmit their ID by generating a random number. If there is a collision in a slot, the reader broadcasts the next identification request only to tags which collided in that slot. We evaluated its performance in terms of number of slots, that is strictly related to the delay for identifying all tags. We compared our protocol performance with that of other Framed Slotted Aloha based and query tree based protocols by simulation. The results of the simulation experiments show that our protocol behaves better than the others. Specifically, TSA achieved a performance of about 37%, with peaks of over 41%, while all the others did not exceed 36%.

Some problems need further investigation. For instance, it would be interesting to assess the performance of our protocol when the reader moves, namely when the set of tags to be identified changes over time. Another issue is a simplification of the tags procedure and of its hardware requirements.

References

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, 2^{nd} ed. John Wiley & Sons, March 2003.

[2] J. Myung and W. Lee, "An adaptive memoryless tag anti-collision protocol for RFID networks,". *Poster paper at IEEE INFOCOM 2005*, Miami, FL, March 2005.

[3] C. Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification (extended abstract)," *Proc. of DIALM '00*, New York, NY, 2000, pp. 75–84.

[4] H. Vogt, "Efficient object identification with passive RFID tags," *Proc. of Pervasive 2002*, pp. 98–113.

[5] S. Lee, S.D. Joo, and C.W. Lee, "An enhanced dynamic framed slotted aloha algorithm for RFID tag identification," *Proc. of Mobiquitous 2005*, pp. 166-172.

[6] A. Micic, A. Nayak, D. Simplot-Ryl, and I. Stojmenovic, "A hybrid randomized protocol for RFID tag identification," *Proc. of WoNGeN '05*.

[7] L. G. Roberts, "Extensions of packet communication technology to a hand held personal terminal," *Proc. of Spring Joint Computer Conference, AFIPS Conference, 1972*, vol. 40, 1972, pp. 295–298.

[8] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. on Information Theory*, vol. IT-25, pp. 505–515, September 1979.

[9] D. R. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," *Proc. IEEE Intern. Symposium on Information Theory, Boston, USA, August 1998.*

[10] M. A.-I. Center, "Draft protocol specification for a 900 MHz class 0 radio frequency identification tag," http://www.epcglobalinc.org, February, 23^{rd}, 2003.

[11] F. C. Schoute, "Dynamic frame length aloha," *IEEE Trans. on Communications*, vol. COM-31, no. 4, pp. 565–568, April 1983.

[12] I. Chlamtac, C. Petrioli, and J. Redi, "Energy-conserving access protocols for identification networks," *IEEE/ACM Trans. on Networking*, vol. 7, pp. 51–59, February 1999.

[13] L. Devroye, "The height and size of random hash trees and random pebbled hash trees," *SIAM Journal on Computing*, vol. 28, no. 4, pp. 1215–1224, 1999.

Proceedings of the 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)
0-7695-2593-8/06 $20.00 © 2006 **IEEE**