IEEE International Conference on Wireless & Mobile Computing, Networking & Communication

ARPA: An Arbitration Protocol based on Advanced Channel Feedback for Radio Frequency Identification

Flaminio Borgonovo, Matteo Cesana
Dipartimento di Elettronica e Informazione
Politecnico di Milano
Email: {borgonov, cesana}@elet.polimi.it

Abstract

This paper refers to a network scenario featuring a single RFID reader (interrogator) which aims at identifying a large number of objects geared with RFID tags, through multiple and sequential interrogations. Since several tags can answer to each interrogation, a problem of collisions arbitration at the reader arises. Classical previously proposed collisions arbitration protocols based on tree-search algorithms make use of channel feedback information on the outcome of the preceding transmissions either of binary type (COLLIDED/NON COLLIDED), or ternary type (IDLE/SUCCESSFUL/COLLIDED) achieving an average resolution time per object ranging between 2.1 and 2.8 interrogation cycles.

In this paper, we show that the RFID transmission environment allows to obtain an enriched channel feedback information that can be leveraged to greatly improve the efficiency of the identification procedure. We further propose and analyze the ARbitration Protocol with Advanced feedback (ARPA), able to asymptotically provide 100% *efficiency. Finally, we show that the overhead needed to implement our protocol in practical RFID systems easily allows to achieve an average resolution time of about 1.4 interrogation cycles per object.*

1 Introduction

Radio Frequency IDentification (RFID) systems are being widely deployed by manufacturing and retail companies to automatically track and identify inventory items and raw materials [1]. General RFID implementations include several wireless transponders, *tags*, which store structured information on the items they are attached to (identifier, color, manufacturer, etc.. . .); RFID tags may further respond to interrogation of an RFID transceiver by sending the information locally stored. The RFID transceiver, often referred to as *interrogator* or *reader*, is composed by a radio frequency module and a control unit, and may eventually access an external database to integrate object recognition. The tags may either have their own power supply (active tags), or they may be activated by the electromagnetic filed generated by the reader itself (passive tags)[2].

In such scenario, since the wireless transmissions from the tags to the reader often share a common radio resource, collisions among concurrent tag transmissions may happen at the reader, which is no longer able to identify the colliding tags. As a consequence, collision may impair the accuracy and the efficiency of the whole identification process, thus effective solutions to arbitrate and resolve collisions must be deployed [1].

Collision Arbitration Protocols (CAP) originated in the past to solve the multi-access problem in communication systems such as Local Area Networks. From the seminal paper on Aloha [3] many proposals have appeared that now are being re-adapted to the RFID environment. Broadly speaking, CAPs can be classified into two general families: ALOHA-based protocols, and tree-based protocols. The former ones leverage different flavors of the ALOHA protocol [4, 5] to reduce the tag collision probability. However, since aloha-based solutions are inherently unstable, some tags can remain unresolved for long time, thus leading to the so called ”tag starvation problem” [7].

On the other side, tree-based protocol tackle the multiple access problem among tags as a tree-search problem. Namely, colliding tags are split in subgroups in such a way that, in the end, each subgroup is composed of a single tag only, thus ensuring collision-free transmission. Arbitration protocols belonging to this class are designed starting form the classical approaches to tree search problems [8, 9, 10]. Published tree-based collisions arbitration solutions for RFID system mainly differ on the way the tree is built and searched, that is, on the specific approaches to split colliding tags into subgroups to be resolved sequentially. Tags splitting may be performed randomly [11], on the basis of the tags' identifier [12], or leveraging additional information on tag numbers and distribution [13]. We refer to [6] and [7] for exhaustive overview of collisions arbitration protocols for RFID systems.

978-0-7695-3393-3/08 $25.00 © 2008 IEEE
DOI 10.1109/WiMob.2008.40

In this paper, we propose an effective tree-based collision arbitration protocol, named ARPA (ARbitration Protocol with Advanced feedback), which is based on a new signalling feedback that can be made available in RFID systems. We analytically derive the average identification time provided by the protocol, which is considerably lower (almost halved) with respect to classical tree-based approaches. The paper is organized as follows: Section 2 reviews background concepts of tree-based multiple access protocols; Section 3 describes the reference RFID scenario, and reports the implementation details of ARPA, whose performances are analytically evaluated in Section 4. Section 5 concludes the paper and comments on future extension of the proposed work.

2 Tree-Based Multiple Access Protocols: Background

The characteristics of Collision Resolution Protocols depend on the feedback the channel provides to contending users about a transmission outcome and, usually, their performance increases as the feedback information gets more accurate. For example, the well known ALOHA protocol can operate without any form of channel feedback about the activity of other transmitters; in fact, it only needs to know the outcome of its proper transmission, that is, the binary information SUCCESSFUL/NOT-SUCCESSFUL. Unfortunately it presents stability problems that can lead to very poor performance. ALOHA's stability problems can be solved if channel feedback about the outcome of a transmission period, is made available to all the involved transmitters, allowing the estimation of packet backlog and adapting the transmission probability [4].

In presence of channel feedback another protocol family have appeared in which protocols are inherently stable. Such protocols, first introduced by Capetanakis [8], are known as *splitting protocols*, or *tree-based protocols*. In the simplest case of binary splitting, the set of users that contend in the slot is split into two subsets at each collision, either by the use of a random outcome in the distributed version or by a classification operated by a central station (the reader in RFID case). The subsequent slots are used only by users that belong to the first subset, until each one of them has correctly transmitted its packet. At this point the subsequent slots are used only by users that belong to the second subset until each one of them has correctly transmitted its packet. An initial collision is then resolved by recursively applying the above rule. As a result the set of the initial users is repeatedly split into two subsets, giving rise to subsets that can be represented as the nodes of a binary tree. Figure 1 shows an example of the operation of the basic tree-based protocol with binary splitting. Time slots are represented by nodes ordered according to the superscript $1, 2, \ldots, 11$. The channel feedback distinguishes between collided slots, represented by double circled nodes, and non collided slots. For the reader's convenience we have also reported within the node the number of tags simultaneously transmitting in the slot. The protocol proceeds as follows. At the first slot, represented by the root node, five tags transmit and collide. This group is split into two subgroups composed of two and three tags, respectively. The first subgroup is explored first in the transmission of the second slot, that results in second collision. A further subdivision is attempted, but this time the first subgroup is empty and at the third slot no transmission occurs. At the fourth slot a collision occurs and a subsequent splitting is successful in isolating a tag in each subgroup. Therefore two successfully transmissions occurs in slots 5 and 6. At this time the subgroup that collided in slot 2 has been successfully explored and the protocol proceeds exploring the second subgroup of slot one. The tags of this second subgroup collide again in slot 7 and the protocol keeps splitting until in slots 10 and 11 a couple of successful transmissions signals the end of the procedure: all the contending tag have transmitted their identification successfully.

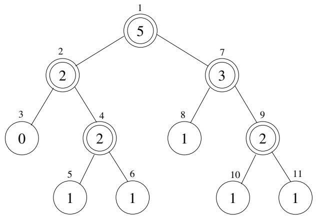

Figure 1. *Tree search with five tags colliding in the first slot.*

Clearly the primary performance figure in the identification (collision resolution) procedure is the average number of slot needed to resolve a given tag, or a given number of tags. Modifications to the binary protocols described above have been envisaged to improve this performance metric.

If the channel feedback can be extended to a ternary-type information, SUCCESSFUL, EMPTY, COLLIDED, the algorithm efficiency can be improved by skipping some nodes (slots) that can be marked as sure collisions in advance. This happens with slot 4 in Figure 1: indeed, slot 3 can be identified as an empty slot, thus the collision happened in slot 2 is sure to happen again in slot 4. In this case the exploration of slot 4 can be skipped, that is, the corresponding subgroup is further split before performing any transmission. In this way, slot 4 in the figure can be spared and the whole procedure now takes only 10 slots.

In some environments, such as satellite communications, the channel can be easily adapted to provide a more enriched channel feedback by adding to the transmission slot a signaling structure that can be leveraged to enhance the tree-based protocol performance [14, 15, 16]. In this paper, we show that even in RFID systems an advanced feedback from the channel can be obtained and used to achieve al-

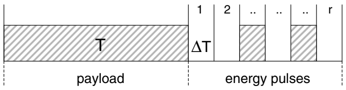

Figure 2. *Organization of a slot*

most halved average identification time with respect to the basic binary tree-based protocols.

3 The ARbitration Protocol with Advanced feedback

3.1 RFID Scenario and Protocol Rationale

The reference RFID system is composed by a number n of tags and by a single interrogator. Interrogator and reader are used equivalently throughout the paper. We further assume the "interrogator talk first" paradigm according to which tags can only send their ID (and/or other local information) upon interrogation by the reader. We consider a time slotted channel which is dynamically built in the following way: at each interrogation step, the reader transmits a synchronization word that is used by the tags to synchronize their responses. We call *interrogation cycle* the time needed to perform the reader's interrogation and the tag (tags) responses, and *slot* the response period within the cycle.

Based on the synchronizing word transmitted by the reader, tag responses are all synchronized and organized as shown in Figure 2. The slot is composed of two parts: the payload part T used for the response by the tags (tag ID transmission) and a signaling part that, in turn, is subdivided into r subintervals of length ΔT. The signalling part is the one that needs to be synchronized and is used to support the effective resolution of potential collision during the payload transmission of the tags. In details, each time a tag transmits its identification in the payload part, it also transmits a short energy pulse in a randomly chosen subinterval of the signaling part. When the reader starts decoding the slot content it also checks for energy presence in each one of the signaling subintervals, thus revealing the presence/absence of transmission. In this way, the reader can:

- distinguish about COLLIDED/NON COLLIDED transmissions;
- group the colliding stations into r subgroups to be arbitrated;
- provide feedback in the subsequent interrogation phase on collision in each subgroup i, $i = 1, 2, \ldots, r$.

The information thus obtained yields a partition in the set of the transmitting tags, which can be used to implement an efficient tree search algorithm for collision arbitration among tags belonging to different subgroups.

In classical tree-based collisions resolution algorithms, the optimal number of subgroups the colliding station are split in comes from a trade-off choice between the need of resolving collisions fast (that pushes toward a high number of subgroups), and the overhead introduced by the exploration of empty subgroups which are unknown a priori (that pushes toward a low number of subgroups). The optimal subdivision in basic tree-based algorithm is therefore found to be equal to 3 subgroups [8].

In our case, the reader can easily get to know which subgroups (intervals) are empty, since no energy is perceived during the corresponding subinterval i. The exploration of corresponding subgroups can therefore be avoided, sparing the waste of slot and encouraging a number of subdivisions much greater than three. Details on the collision arbitration protocol are reported in the following section.

3.2 ARPA Operation Mode

The ARPA, as all tree protocols, can be performed by tags, in a distributed way, using the feedback provided by the reader, or it can be performed in a centralized way by having tags transmitting upon a suitable command issued by the reader. Here we refer to the latter case, easier to implement in an RFID environment.

An interrogation cycle starts with the reader issuing an interrogation command, at which all the interrogated tags reply in the following slot with their ID. Moreover, each responding tag i randomly chooses an integer number X_0^i in the range [1,r], and sends an energy pulse in the corresponding $X_0^i - th$ signalling subinterval. The chosen number X_0^i is stored in the tag as part of a provisional ID to be used in the subsequent identification steps.

If at the first interrogation more than one tag responds, a collision occurs and the collision resolution part takes place. The position of the energy pulses is used by the reader to split the contending tags into r subgroups at most. In the subsequent interrogation cycle, the reader proceeds by interrogating only those tags belonging to one of the subgroups corresponding to a non-empty signalling subinterval. Suppose the reader chooses to interrogate subgroup k at this stage. All those tags that have chosen slot k in the previous stage, that is every tag j such that $X_0^j = k$, will transmit their IDs and will further generate an energy pulse in one of the subintervals randomly chosen.

If the interrogation of subset k is responded by only one transmission, meaning that only one tag is present in the queried subset, the corresponding tag is identified and the reader proceeds querying the other non-empty subgroup at stage 0. On the other hand, if the query causes a collision, the reader will issue an interrogation at the subsequent step of type $X_0 = k, X_1 = l$, to which only those tags having chosen subinterval k at the first response, and subinterval l at the second one will answer. The procedure is iterated until all tags are correctly identified by the reader.

It is clear that the protocol operates by splitting colliding subgroups of tags and exploring these subsets sequen-

Table 1. *Example of ARPA operation in case $r = 2$ and 3 tags to be identified.*

	I_0={ALL}		I_1={S_1}		I_2={S_1,S_1}		I_3={S_1,S_2}		I_4={S_2}	
	Transmit	SubInt.	Transmit	SubInt.	Transmit	SubInt.	Transmit	SubInt.	Transmit	SubInt.
TAG_1	√	S_1	√	S_1	√	ANY				
TAG_2	√	S_1	√	S_2			√	ANY		
TAG_3	√	S_2							√	ANY

tially. In other words, a tree structure is created by the reader whose nodes represent the responses to the reader's interrogations and the branches represent the splitting into subgroups operated through the signalling subintervals. Table 1 reports the operation of ARPA in case r =2 and n =3 tags need to be identified by the reader. Each column represents an interrogation cycle, specifying the interrogated subgroups, which tags do transmit in the cycle, and which subinterval each transmitting tag chooses. At the first interrogation cycle, the reader issues a generic interrogation ($I_0 = ALL$) to which all the tags answer leading to collision. TAG_1 and TAG_2 further choose subinterval S_1 ($X_0^1 = X_0^2 = S_1$) to send the energy pulse, whereas TAG_3 picks subinterval S_2 ($X_0^3 = S_2$). During the following interrogation cycle, the reader explores the subgroup corresponding to subinterval S_1, by issuing the interrogation $I_1 = \{S_1\}$. Having chosen subinterval S_1 during the previous cycle, TAG_1 and TAG_2 both respond causing a collision again; they further choose at this time subinterval S_1 ($X_1^1 = S_1$) and S_2 ($X_1^2 = S_2$) respectively. During the following cycles, the reader issues subsequent interrogations of type $I_2 = \{S_1, S_1\}$, $I_3 = \{S_1, S_2\}$ and $I_4 = \{S_2\}$, to which only TAG_1, TAG_2 and TAG_3 answer, respectively, all the tags being identified. We observe that the signaling mechanism and the feedback usage adopted in ARPA share similar ideas as the one proposed in [15, 16]. However, differently from ARPA, the protocols proposed in [15, 16] aim at achieving throughput maximization, rather than the minimization of the identification time. This leads to substantial differences in the arbitration phase, as well as in the efficiency derivation approach.

4 Analysis and performance evaluation

In this section we derive the expression for the average identification time needed by the protocol to explore all the tree and identify all the tags, assuming that the energy pulse is transmitted in a subinterval uniformly chosen among r available subintervals.

We start in Section 4.1 by deriving the gross efficiency, that is, the average number L_n of interrogation cycles needed to achieve full identification. In Section 4.2 we calculate the interrogation efficiency depurated of the overhead introduced by the protocol.

4.1 Average Number of Interrogation Cycles

We approach the calculation by first deriving L'_n, which is the average number of interrogation cycles needed to explore the tree with the basic protocol, and then finding the relation with L_n. To this purpose we resort to the derivation techniques proposed in [11] and [8].

We observe first that L'_n coincides with the average number of nodes in the tree visited by the algorithm. Since a node is visited only if its parent node suffered a collision, the average number of nodes visited at level $l + 1$ is given by

$$\sum_{i=0}^{r^l-1} r\beta_n(l,i) \tag{1}$$

having denoted by $\beta_n(l,i)$ the probability of a collision at the i-th node at level l, $l = 0, 1, 2, \ldots$. Considering that the root node is always visited we have

$$L'_n = 1 + \sum_{l=0}^{\infty} \sum_{i=0}^{r^l-1} r\beta_n(l,i). \tag{2}$$

Exploiting the fact that $\beta_n(0,0) = 1$, Eq. (2) changes into

$$L'_n = 1 + r + \sum_{l=1}^{\infty} \sum_{i=0}^{r^l-1} r\beta_n(l,i) \tag{3}$$

The tags transmitting at a generic node (l,i) are those that have selected the path from the root node to node (l,i). Since at each node a specific branch is chosen with probability $1/r$, the probability that a tag transmits at node (l,i) is $(1/r)^l$. Therefore, the corresponding collision probability, i.e., the probability that two or more transmissions occur at such node is derived by the binomial distribution as:

$$\beta_n(l,i) = 1 - (1-(1/r)^l)^n - n(1/r)^l(1-(1/r)^l)^{n-1} \tag{4}$$

Substituting Eq. (4) into Eq. (3) provides the solution.

Now, if we denote C'_n and Z'_n respectively the average number of slots with collisions and zero replies encountered during the exploration of the tree, we have:

$$L'_n = C'_n + Z'_n + m, \tag{5}$$

where m accounts for the non-collided transmissions, The expression for C'_n is only slightly different from (3). In fact, it is obtained by summing $\beta_n(l,i)$ over all nodes at all levels of the tree, which yields:

$$C'_n = 1 + \sum_{l=1}^{\infty} \sum_{i=0}^{r^l-1} \beta_n(l,i). \tag{6}$$

Checking with Eq. (3), Eq. (6) may be written as:

$$C'_n = \frac{1}{r}(L'_n - 1). \tag{7}$$

From Eq. (5) and Eq. (7) we finally derive Z'_n as

$$Z'_n = \frac{r-1}{r} L'_n - n + \frac{1}{r} \tag{8}$$

Turning to the analysis of ARPA, the average identification time L_n can be promptly derived by simply observing that the exploration of empty subset is avoided through the use of the signalling subintervals. Therefore, we have:

$$L_n = L'_n - Z'_n \tag{9}$$

which becomes:

$$L_n = \frac{1}{r} L'_n + n - \frac{1}{r}. \tag{10}$$

This is the expression that relates the identification efficiency of the two protocols (traditional tree search vs ARPA) allowing a direct comparison. The explicit expression for L_n is obtained by substituting Eq. (3) into Eq. (10):

$$L_n = 1 + n + \sum_{l=1}^{\infty} \sum_{i=0}^{r^l-1} \beta_n(l,i). \tag{11}$$

A closed form expression for L'_n is well known to be [17]:

$$L'_n = 1 + r \sum_{i=2}^{n} \binom{n}{i} (-1)^i \frac{r^{i-1}(i-1)}{r^{i-1}-1} \quad n \geq 2, \tag{12}$$

that, leveraging Eq. (10), provides:

$$L_n = n + \sum_{i=2}^{n} \binom{n}{i} (-1)^i \frac{r^{i-1}(i-1)}{r^{i-1}-1} \quad n \geq 2. \tag{13}$$

It is well known [17, 18] that an approximated asymptotic expression for L'_n, very accurate even for low values of n, is:

$$L'_n \simeq \frac{r}{\ln r} n. \tag{14}$$

The corresponding expression for L_n is derived from Eq. (10) as:

$$L_n \simeq \frac{1 + \ln r}{\ln r} n - \frac{1}{r} \tag{15}$$

The above expressions clearly shows the difference between the two protocols. The basic one is optimized by $r = 3$, as shown in Table 2, whereas ARPA is optimized by $r = \infty$, where the absolute maximum efficiency equal to one can be reached. The opposite behavior of the two protocols must be ascribed to the different policies for the exploration of empty slots, which always increases with r. In the basic protocol an empty slot must be explored, wasting the corresponding slot, while in ARPA the exploration is skipped and the corresponding slot is saved. We further see that for moderate values of r, such as $r = 10$, the efficiency of the new protocol doubles that of the basic one. Table 3 reports the value L_n/n for low values of n and r, where the first line for $r_b = 2$ represents the basic algorithm with binary splitting, which is optimal for small values of n.

Table 2. *Linear coefficients of L_n and L'_n as function of the subintervals number, r.*

r	2	3	4	5	10	∞
$r/\ln r$	2.885	2.730	2.885	3.107	4.343	∞
$(1+\ln r)/\ln r$	2.443	1.910	1.721	1.621	1.434	1

Table 3. *Values of L_n/n as function of n for different values of r. The first line for $r_b = 2$ represents the basic algorithm with binary splitting*

	$n=2$	$n=3$	$n=4$	$n=5$	$n=10$
$r_b=2$	2,5	2,556	2,631	2,684	2,785
$r=2$	2	2,111	2,190	2,242	2,343
$r=3$	1,75	1,75	1,779	1,805	1,861
$r=5$	1,625	1,556	1,548	1,556	1,601
$r=10$	1,556	1,438	1,397	1,385	1,412

4.2 ARPA Overhead

We observe here that the efficiency of ARPA must be balanced against the overhead introduced, that causes an increase of the slot time by an amount of $r\Delta T$ with respect to tree-based arbitration protocol without signalling feedback. To this purpose, we note that ΔT must be composed of two parts: a time ΔX needed to safely detect energy presence, and a guard time $\Delta\tau$ needed to avoid the overlapping of adjacent subintervals, as seen by different tags. The minimum guard time can be estimated as: $\Delta\tau = \max_{i,j} 2|\tau_i - \tau_j|$, where $2\tau_i$ is the round trip propagation time between tag i and the reader, including the tag reaction time.

Usually, typical RFID systems feature propagation delay values far below 1 μs [19]; moreover, turn-around time values to switch from reception to transmission typically amount to hundreds of microseconds, so that the uncertainty in the reaction time can also be well below 1 μs. Accurate energy detection should also be feasible within a few bits period, say 5 bit period, which, with 100 kb/s transmission rate, amounts to $50\mu s$, largely the dominant figure. Therefore, a value of $\Delta T = 50\mu s$, corresponding to 5 bits should represent a realistic figure.

Let now β be the subinterval length normalized to the payload length; the number of interrogation cycles, including signalling overhead, becomes:

$$L_n^{net} = L_n \frac{1}{1 + r\beta}. \tag{16}$$

By substituting Eq. (15) into Eq. (16), we get an estimate of the actual identification time of ARPA including the signalling overhead. Figure 3 reports the identification cycle duration when varying the number of subinterval used for signalling purposes, in case n =50 and n =100 tags have to be identified and $\beta = 1/100$, which corresponds to a case where the slot payload is 500 bits long. The figure reports both L_n (dashed curves) and L_n^{net} (continuous curves) and clearly shows the tradeoff involved in the choice of the number of subinterval r between the efficiency of the split-

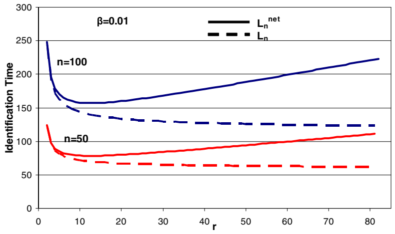

Figure 3. *Identification time versus the number of subintervals r including the signalling overhead.*

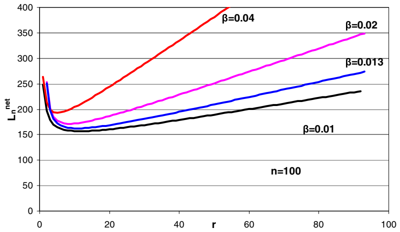

Figure 4. *Effect of the overhead parameter β on the identification when varying the number of subintervals r.*

ting procedures (high values of r required) and the need of limiting the overhead (low values of r to be preferred). Indeed, the curves representing the identification time present a minimum in r that is still very pronounced.

The behavior of L_n^{net} obviously depends on the parameter β representing the length of each subinterval normalized to the slot payload length. Intuitively, as β increases, the signalling overhead becomes predominant. This is readily confirmed by Figure 4 which reports the very same L_n^{net} curves of Figure 3 for various value of β. As clear from the figure, as the normalized subinterval length increases (β increases), the impact of the signalling overhead becomes relevant also for low values of r.

5 Conclusions

Radio Frequency Identification systems are being massively deployed to support a vast range of applications to track and identify inventory items and raw materials equipped with RFID tags. To increase the efficiency of these systems (identification/tracking time of the items), it is of utmost importance to implementing effective mechanism to detect and resolve collisions among multiple and concurrent tag transmissions.

To this extent, we have proposed the ARbitration Protocol based and Advanced feedback (ARPA) which leverages a tree-based multiple access mechanism to arbitrate tag transmissions. ARPA implements an efficient splitting mechanism of colliding tags which is based on advanced signalling information provided by the channel. We have shown how such information can be provided in RFID systems, and we have derived analytical formulas for the average identification time under ARPA. Numerical results show that, by exploiting the feedback information, ARPA can provide an average identification time almost halved with respect to classical tree-based approaches.

References

[1] Y. Xiao, S. Yu, K. Wu, Q. Ni, C. Janecek, J. Nordst, *Radio frequency identification: technologies, applications, and research issues*, Wirel. Commun. Mob. Comput, 2007, Vol. 7, Page(s): 457-472.

[2] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*, John Wiley & Sons, 2003.

[3] N. Abramson, *The Aloha SystemAnother Alternative for Computer Communications,* Proc. Fall Joint Computer Conf., Am. Federation of Information Processing Soc. Conf., Nov. 1970, Vol. 37, Page(s): 281-285.

[4] F. Schoute, *Dynamic Frame Length ALOHA*, IEEE Trans. Comm., Apr. 1983, Vol. 31, No. 4, Page(s): 565-568.

[5] J. Wieselthier, A. Ephremides, and L. Michaels, *An Exact Analysis and Performance Evaluation of Framed ALOHA with Capture*, IEEE Trans. Comm., Feb. 1989, Vol. 38, No. 2, Page(s): 125-137.

[6] Dong-Her Shih, Po-Ling Sun, David C. Yen, Shi-Ming Huang, *Taxonomy and survey of RFID anti-collision protocols*, Comp. Comm., 2006, Vol. 29, No. 11, Page(s): 2150-2166.

[7] J. Myung, W. Lee, J. Srivastava, T. K. Shih, *Tag-splitting: Adaptive collision arbitration protocols for RFID tag identification*, IEEE Trans. on Paral. and Distr. Systems, June 2007, Vol. 18, No. 6, Page(s): 763-775.

[8] J.I. Capetanakis, *Tree Algorithms for Packet Broadcast Channels*, IEEE Trans. on Inform. Theory, September 1979, Vol. IT-25, No. 5, Page(s): 505-516.

[9] R.G. Gallager, *Conflict resolution in random access broadcast networks*, in proc. of AFOSR Workshop in Communication Theory and Applications, Provincetown, MA, USA, Sept. 1978.

[10] J. L. Massey, *Collision resolution algorithms and random-access communications* Tech. Rep. UCLA-Eng-8016, Univ. of California at Los Angeles, Los Angeles, CA, April 1980.

[11] D.R. Hush, C. Wood, *Analysis of tree algorithms for RFID arbitration*, IEEE International Symposium on Inform. Theory, Cambridge, MA, USA, 16-21 August, 1998, Page 107.

[12] H. Vogt, *Efficient object identification with passive RFID tags*, in proc. of the International Conf. on Pervasive Comp., 2002, Page(s): 98-113.

[13] M. A. Bonuccelli, F. Lonetti, F. Martelli, *Exploiting id knowledge for tag identification in rfid networks*, in proc. of ACM PE-WASUN 2007, October 22-26, 2007, Crete Island, Greece, Page(s): 70-77.

[14] F. Borgonovo, L. Fratta, *A collision resolution algorithm for random-access channels with echo*, ACM SIGCOMM Computer Communication Review, Jan. 1983, Vol. 13, No. 1.

[15] D. Towsley, P. O. Vales, *Announced Arrival Random Access Protocols*, IEEE Trans. on Comm., May 1987, Vol. COM-35, No. 5, Page(s): 513-521.

[16] Y. Oie, T. Suda, H. Miyahara, T. Hasegawa, *Throughput and delay analysis of free access tree algorithm with minislots*, IEEE Trans. on Comm., Feb. 1990, Vol. COM-38, No. 2, Page(s): 137-141.

[17] P. Mathys, P. Flajolet, *Q-ary Collision Resolution Algorithms in Random-Access Systems with Free or Blocked Channel Access*, IEEE Trans. on Inform. Theory, March 1985, Vol. IT-31, No. 2, Page(s): 217-243.

[18] M. A. Kaplan, E. Gulko, *Analytic Properties of Multiple-Access Trees*, IEEE Trans. on Inform. Theory, March 1985, Vol. IT-31, No. 2, Page(s): 255-263.

[19] *Information Technology Automatic Identification and Data Capture TechniquesRadio Frequency Identification for Item Management Air InterfacePart 6: Parameters for Air Interface Communications at 860-960 MHz*, Intl Standard ISO 18000-6, Nov. 2003.