Multiple Object Identification with Passive RFID Tags

Harald Vogt

Department of Computer Science
Swiss Federal Institute of Technology (ETH)
8092 Zürich, Switzerland
vogt@inf.ethz.ch

Abstract— **We investigate the applicability of passive RFID systems to the task of identifying multiple tagged objects simultaneously, assuming that the number of tags is not known in advance. We present a combinatorial model of the communication mechanism between the reader device and the tags, and use this model to derive the optimal parameter setting for the reading process, based on estimates for the number of tags. Some results on the performance of an implementation are presented.**

Keywords— **RFID, collision-resolution, tagging, combinatorics.**

I. Introduction

The main application of passive RFID systems so far is the identification of objects where items are presented to the antenna one at a time, or where the identification of a single tag (out of many tags attached to the same object) is sufficient to identify an item. A different problem is the identification of multiple tags where each tag determines the identity of a different object. In this case, it is not sufficient to identify a small subset of tags, but all tags should be identified robustly. As examples, consider laundry services, warehouses, or a supermarket checkout.

In this work, we investigate the applicability of passive RFID systems to the simultaneous identification of multiple objects. Advanced RFID systems support this capability by providing anti-collision techniques that address the problem of tag messages cancelling each other out. There are two basic approaches. Deterministic collision resolution works by muting subsets of tags that are involved in a collision [2], [5]. By successively muting larger subsets, only one tag will be left sending its message. Once a tag has successfully sent its message, it will not resend it, and the other tags are woken up one after another. The other approach is stochastic resolution of collisions. Since tags use a shared communication medium, it is natural to fall back upon an Aloha-like protocol that provides slots for the tags to send their data. Whenever a collision has occurred, another frame of slots is provided, and hopefully the tags will choose different slots this time, such that no collision occurs. This paper describes the analysis of a system that follows the latter approach.

For practical experiments, we have used a commercially available RFID system, "I-Code" by Philips Semiconductors [7], that offers various capabilities for communicating with tags. Tags have a 64-byte static random access memory, where a unique 8-byte identifier is stored. The simplest request from the reader device to the tags is a *read unselected* request that triggers all tags that are currently present in the antenna's field to send a message. It is noteworthy that all tags will resend their data on a second request, regardless of whether they were previously successful in sending their message or if their message was garbled by a collision (we assume no feedback channel to inform the tags about read results). This makes it possible (though very unlikely) that a tag is never recognized, if all its messages are garbled by collisions in all read cycles. Therefore, it cannot be guaranteed that all tags will be successfully identified, though the chance becomes minimal if a sufficiently large number of read cycles are performed.

We are using a combinatorial model of the communication system and devise a method that can be used to identify a large number of tags, all being present in the reader's field at the same time, assuming that the number of tags being present is not known in advance. Based on the outcomes of read cycles, the number of tags is estimated, and the frame size (a frame encloses the slots provided for tags to send their messages) is adapted according to this estimate. The model tells us, how to adapt the frame size in order to maximize the number of identifiable tags, while keeping the overhead low. By performing an appropriate number of read cycles, we are able to minimize the time required to identify all tags with a certain probability.

We assume that all messages involved in a collision are destroyed. Therefore, our model does

not take into consideration the *capture effect* that is prevalent in wireless systems. By this effect, all but one message involved in a collision are destroyed, and a single "dominating" message can be received correctly. We ignore this effect since its impact on the performance of the considered system is marginal.

Note: This paper is an abridged, and improved, presentation of work that is also described in [12].

II. System Model

The I-Code system employs a variant of slotted Aloha for access to the shared communication medium, known as *framed* Aloha [8]. A read cycle consists of two steps. In the first step, the reader broadcasts a request. The request message contains an address range, which determines what data the tags should return. It also contains a random number that is used by the tags as a seed to determine their answering slot. In the second step, a frame with a number N of slots is provided for the tags to send their messages. Each tag chooses one slot randomly and sends its data within that slot. If two or more different tags choose the same slot for sending their answer, a collision occurs and all data is lost.

For the purpose of our analysis, we are not interested in the actual data sent by the tags, but only in the quantities of empty, filled, and (due to collisions) garbled slots. Such a read result will be represented by the triple

$$c = \langle c_0, c_1, c_M \rangle .$$

The allocation of tags to slots belongs to the class of "occupancy problems" [3], [4]. Given N slots and n tags, the number r of tags in one slot is binomially distributed, and the expectation value for the number of slots that all have r tags assigned to them is given by the *occupancy number* a_u:

$$a_u = N\binom{n}{r}\left(\frac{1}{N}\right)^u\left(1-\frac{1}{N}\right)^{1-u} . \qquad (1)$$

Maximum throughput is reached when the number of slots filled with exactly one message is maximized. This is the case if $N = n$, as the analysis of slotted Aloha indicates [11]. One is, however, not necessarily only interested in optimizing throughput, but in minimizing the required overall time for identifying all tags. Therefore, the total time consumed for a read cycle must be taken into consideration. Although it can be expected that read cycle time is linearly proportional to the frame size, there might be large fixed costs attached to performing a read cycle (depending on the actual system; e.g. including the time for sending the antenna's request). Table I shows typical time requirements for read cycles for the I-Code system. A simple calculation shows that the fixed share of the time is 52 ms, which amounts to 25% of the time required for 32 slots, and is therefore not negligible. (The duration of a single slot is approximately 4.9 ms, but this number depends on the amount of data requested—in our case, only the 8-byte identifier is transmitted.)

TABLE I

Execution time for *read unselected*. Figures shown are for a 57600 baud connection over RS-232

N slots	t_N (ms)
1	56
4	71
8	90
16	128
32	207
64	364
128	676
256	1304

III. Multiple Tag Identification

The scenario considered is based upon the assumptions that (a) an application is trying to identify all tags that are present in the field of the reader device, (b) the tag set is more or less static (i.e. tags don't move in and out frequently), and (c) the number of tags is not known in advance. This poses basically two challenges to the reading process. First, it cannot choose the optimal frame size, since the number of tags is unknown. However, the more read cycles are performed, the better estimates of this number can be obtained and the frame size can be adapted. The second problem is to know when to stop reading. Again, the number of tags is not known and the application cannot be sure when it has identified all of them. In this section, we devise a procedure for solving these problems.

A. Estimation of Tag Set Size

We are using a simple guess on the number of tags actually present from the outcome of a read cycle $c = \langle c_0, c_1, c_M \rangle$. Since in every collision, at least two tags are involved, and every tag chooses one and only one slot for sending its message, a lower bound on the number of tags is

$$\gamma : \langle c_0, c_1, c_M \rangle \mapsto c_1 + 2c_M . \qquad (2)$$

We can use this lower bound as an estimate for the number of tags. The expected error of the estimate can be obtained by summing up the weighted errors

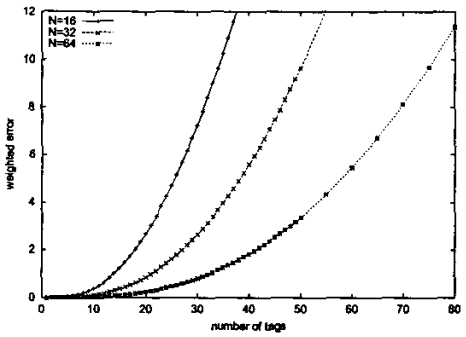

Fig. 1. Error of tag set size estimation

over all possible outcomes of a read cycle:

$$\varepsilon = \sum_{t} |\gamma(c) - n| P(\mu = c) \; . \tag{3}$$

Fig. 1 shows the expected error for various frame sizes N. Note that, from a certain point on, the error is growing very fast, but since we adapt N to the number of tags, we are normally using an estimate for which the error is acceptable.

B. Tag Reading as a Markov Process

The process of reading tags can be modelled as a homogeneous Markov process $\{X_i\}$, where a state X_i denotes the number of identified tags after s read cycles. The state X_{i+1} depends only on the previous state X_i and the outcome of the read cycle. The discrete, finite state space of the Markov process is $\{0, 1, \ldots, n\}$. The transition probabilities are given by the transition matrix $Q = (q_{pe})$ defined as

$$q_{pe} = \begin{cases} 0 & \text{if } j < i \\ \sum_{o=0}^{p} P(\mu_1 = k) \frac{\binom{i}{k}}{\binom{n}{k}} & \text{if } j = i \\ \sum_{o=e-p}^{l} P(\mu_1 = k) \frac{\binom{n-i}{j-i}\binom{i}{k-j+i}}{\binom{n}{k}} & \text{if } j > i \end{cases} , \tag{4}$$

where $P(\mu_u = m_u)$ denotes the probability distribution of the slots filled with exactly $r$tags ($r = 0, 1, 2, \ldots, n$). This distribution is given by [12]:

$$P(\mu_u = m_u) = \frac{\binom{b}{j\,r} \prod_{o=0}^{j-r-1} \binom{l-o}{u} \; G(N - m_u, n - r m_u)}{N^l} , \tag{5}$$

where

$$G(M, m) = M^j + \sum_{o=1}^{\lfloor \frac{m}{r} \rfloor} \left[(-1)^o \cdot \prod_{e=0}^{o-1} \left\{ \binom{m - jr}{r} (M - j) \right\} (M - k)^{j - o_u} \frac{1}{k!} \right] . \tag{6}$$

The rationale for the transition probabilities q_{pe} is the following. In the first case, the number of identified tags decreases, which is not possible and has therefore zero probability. In the second case, no new tags are found, which means that up to $i$tags are found, but all of them already have been identified earlier. The third case happens if $(j - i)$ tags are found that have not been identified yet (and possibly some more that are already known).

The assumption that at the beginning of the process, none of the tags is known, is reflected by the following initial distribution $q(0)$:

$$q(0) = (1, \overbrace{0, \ldots, 0}^{l}) \; . \tag{7}$$

The Markov process computes probabilities for the identification of a number of tags after a number of steps (i.e. read cycles). This yields the following probability of identifying the full set of tags present in the field, after s steps:

$$Q^i \, q(0)[n] \; . \tag{8}$$

This is useful for applications that want to stop reading when they have identified all tags with sufficient probability.

C. Full Tag Set Identification

After a certain number of read cycles, the probability of having identified all tags that are in the field has reached a level, which is sufficient for a certain kind of application. A value of 0.95, e.g., would indicate that all tags have been identified with a probability of 95%. That means, out of one hundred runs, there are five runs that miss one or—very rarely—more tags.[1]

Now, we have the answer to the question of how many read cycles to perform. But what frame size should be chosen? The frame size depends on the number of tags in the field (or, as this number is not known, an estimate thereof). The frame size to be chosen is the one that minimizes the time T_c

[1]With $N = 64$ and $n = 40$, after 9 read cycles we have a probability of about 96.45% of identifying all 40 tags. The chance to miss one tag is 3.48%, leaving less than 0.07% to miss more than one.

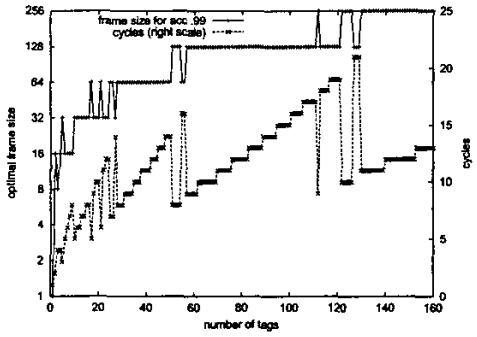

Fig. 2. Optimal frame sizes and numbers of read cycles to perform

TABLE II
TOTAL TIME FOR 64 AND 128 SLOTS

n	N	# read cycles	t_b	total time
51	64	15	364	5460
51	128	8	676	5408
56	64	16	364	5824
56	128	9	676	6084

TABLE III
OPTIMALITY INTERVALS FOR FRAME SIZES

N slots	1	4	8	16	32	64	128	256
n_{Idn}	–	–	–	1	10	17	51	112
n_{fipafi}	–	–	–	9	27	56	129	∞

required to identify the full tag set with probability α. T_c is obtained by

$$T_c = s_0 \cdot t_b \ , \tag{9}$$

where s_0 is the minimum number of read cycles required to identify all tags in the field with probability α. Therefore, s_0 is the minimum value of s for which the condition

$$Q^i q(0)[n] \geq \alpha \tag{10}$$

holds. Figure 2, for up to 160 tags, the optimal frame size and the number of read cycles to perform for a given number of tags in order to achieve $\alpha = 0.99$. The graph is based on the values in Table I.

IV. PRACTICAL ADAPTIVE TAG IDENTIFICATION

In Figure 2 it can be seen that, the function that determines the optimal frame size is not monotonic in the number of tags. Within certain "transition periods", there is no unique optimal frame size; e.g. for $n \in [51, 54]$, a frame size of 128 is suggested, while for $n = 55$ or 56, the optimal size is 64, and for n greater than 57, it becomes again 128. So it seems reasonable that a frame size of 64 can be chosen for values of n up to 56, while 128 is fine with values greater or equal 51. Indeed, this can be done without much overhead, as Table II shows. Table III summarizes this consideration and gives intervals for n for which a certain frame size is reasonable.

A. A Procedure for Tag Set Identification

Assume there is a process waiting for tags to appear in the field, trying to identify them as soon as they arrive. In order to detect their appearance, the tag reading process is continuously performing read cycles with a small frame size, e.g. 16, as this is the smallest reasonable value from Table III. As soon as the first read result $c = \langle c_0, c_1, c_M \rangle$ with $c_1 + c_M > 0$ is encountered, the procedure shown in Figure 3 is started.

The procedure begins with an initial frame size of 16 and an estimate of zero for the number of tags `n_est`. Variable `stepN` holds a counter for the number of read cycles performed with the current setting of N. When this counter reaches its maximum value (the optimal number of cycles to perform), the loop terminates. `stepN` is increased in every step, and reset to zero whenever a new frame size is set. This guarantees that the optimal number of cycles is performed with the (hopefully optimal) setting of N. However, this renders all steps performed up to this point irrelevant, and introduces a time penalty.

Note that the estimate for n, held in variable

```
identifyStatic() {
  N = 16; n_est = 0; stepN = 0;
  do {
    stepN++;
    c = performReadCycle(N);
    t = estimateTags(N, c);
    if (t > n_est) {
      n_est = t;
      NO = adaptFrameSize(N, n_est);
      if (NO > N) {
      // restart with new frame size
        stepN = 0;
        N = NO;
      }
    }
  } while (stepN < maxStep(N, n_est));
}
```

Fig. 3. Procedure to adaptively read a static set of tags

`n_est`, is only updated if a new estimate excels the old one. The same is true for the frame size, held in `N`. By this monotony, we can guarantee termination of the procedure, since there is a maximum value for `N`. The auxiliary procedure `adaptFrameSize` can be implemented as a simple lookup in Table III; `maxStep` yields the optimal number of cycles as in Figure 2; the rest of the code should be self-explaining.

B. Experimental Results

We have implemented the tag reading scheme presented here and executed the procedure `identifyStatic` for tag sets of up to 60 tags. The tags were arranged around the antenna of the reader device in a way that we hoped would optimize field coverage. During the test, the tags were not moved. For each tag set, we carried out 100 runs of the identification procedure (without changing the arrangement in between).

The intended level of a 0.99 probability of identifying all tags (meaning that, out of 100 runs only one run should miss any tags) could not be reached in all tests—the worst level being with a set of 34 tags, where only 92 of the 100 runs yielded the full tag set (see Fig. 4). Not surprisingly, accuracy suffers with increasing tag numbers, due to the fact that it becomes increasingly difficult to arrange a larger number of tags around the antenna while maintaining good coverage. The drop in accuracy at around 30 tags could be attributed to the increasing error of our estimation function. Another source of inaccuracy are objects located around the testbed (walls, chairs, etc.), which could have a negative influence on the tests. However, the procedure consumed only a bit more time than would be required if the number of tags were known in advance (see Fig. 5 for a comparison of actual to expected running time). There are running time peaks in ranges where the error of the estimate becomes larger, which causes good estimates often to be made only after some time has already been spent on a suboptimal frame size. Then, the frame size is adapted, and all cycles are re-performed, resulting in additional time overhead.

Another measure describing the accuracy of our procedure is the fraction of tags recognized over a large number of runs. In our tests, this figure (not shown graphically) never dropped below 0.99. This means practically that, although we might miss some tags in a few runs, these misses add up to only a small percentage of missed tags in the long perspective.

One has to bear in mind that these figures are very sensitive to environmental conditions and the

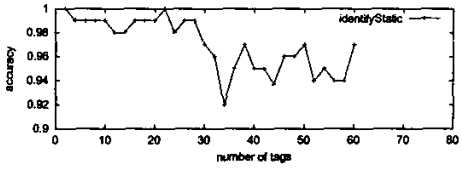

Fig. 4. Accuracy of tag identification in practice. The graph shows the percentage of runs that yielded the full tag set

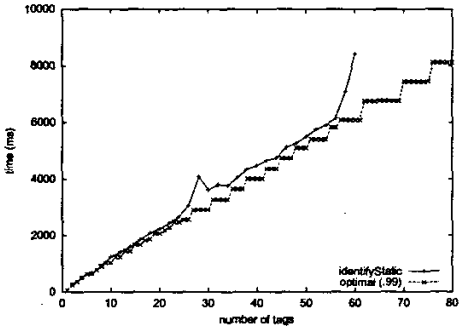

Fig. 5. Actual running time for tag identification compared to the optimal case (optimal setting for N multiplied by t_N)

relative positions of the tags to the antenna. We performed our tests in an office environment where we could carefully arrange the tags and were able to somewhat optimize the conditions under which the tests took place. We expect the accuracy that can be reached under real-world conditions to be much lower.

V. Related Work

Apart from mundane applications such as inventory or cattle tracking, tagging is often used for attaching information to real objects and for building a bridge between them and their virtual counterparts [1], [13]. Such approaches are based on the assumption that it is often more appropriate to integrate the world of information into our physical environment. This can help facilitate access to information by means of familiar objects acting as interfaces.

Aloha is a classical communication protocol that is described and analysed in many introductory textbooks. Framed Aloha was introduced in [8], [9]. The underlying model in that work differs from ours in the assumption that nodes are able to detect if a message could be sent successfully and will only resend it if this was not the case. A procedure

for frame size adaptation is given that depends on the outcome of a read cycle and tries to maximize throughput. The performance of framed Aloha systems is extensively studied in [14]. The analysis takes also into consideration the capture effect, by which a message is received despite of a collision.

Similar to framed Aloha is slotted Aloha with subchannels (S/V-Aloha), which is introduced by and studied in [10]. This work uses a combinatorial system model similar to ours.

General collision-resolution algorithms are discussed in [6]. Deterministic, tree-based anti-collision schemes for RFID systems are studied in [2], [5]. Trees are constructed "on the fly" as collisions occur. Each branch corresponds to a partition of the tag set, leafs represent singled-out tags that are identified without a collision. The approaches differ in how the branch of a tag is determined. In [5], ID prefixes determine the partitioning, while in [2], random coin flipping is used. The latter work also considers trees with arity ≥ 2. Both papers investigate how much effort is required for full, reliable tag identification. In contrast to stochastic schemes such as examined in our work, deterministic ones allow for a recognition rate of 100%, but note that this rate is only achievable under optimal conditions; if tags are allowed to enter or leave while the protocol is in progress, accuracy may suffer.

VI. Conclusions and Future Work

We have demonstrated how to efficiently identify a fixed set of RFID tags if the number of tags is not known in advance. We have shown how to determine the parameters for tag reading in order to achieve optimal running time under a given assurance level (i.e., full identification probability). Due to physical limitations and environmental influences, it is hard to achieve that level in practice, however.

This work should be expanded to the case of *continuous* tag reading, where tags enter and leave the field continuously at high frequencies. The challenge here is to maintain a high identification rate.

References

[1] L. E. Holmquist, J. Redström, and P. Ljungstrand. Token-Based Access to Digital Information. In Hans-W. Gellersen, editor, *Handheld and Ubiquitous Computing*, volume 1707 of *LNCS*, pages 234–245. Springer-Verlag, 1999.

[2] Don R. Hush and Cliff Wood. Analysis of Tree Algorithms for RFID Arbitration. In *IEEE International Symposium on Information Theory*, pages 107–. IEEE, 1998.

[3] Normal Lloyd Johnson and Samuel Kotz. *Urn Models and Their Applications*. Wiley, 1977.

[4] Valentin F. Kolchin, Boris A. Svast'yanov, and Vladimir P. Christyakov. *Random Allocations*. V. H. Winston & Sons, 1978.

[5] Ching Law, Kayi Lee, and Kai-Yeung Siu. Efficient Memoryless Protocol for Tag Identification. In *Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 75–84. ACM, August 2000.

[6] James L. Massey. Collision-Resolution Algorithms and Random-Access Communications. In G. Longo, editor, *Multi-User Communication Systems*, number 265 in CISM Courses and Lectures, pages 73–137. Springer-Verlag, 1981.

[7] http://www.semiconductors.philips.com/markets/-identification/products/icode/.

[8] Frits C. Schoute. Control of ALOHA Signalling in a Mobile Radio Trunking System. In *International Conference on Radio Spectrum Conservation Techniques*, pages 38–42. IEE, 1980.

[9] Frits C. Schoute. Dynamic Frame Length ALOHA. *IEEE Transactions on Communications*, COM-31(4):565–568, April 1983.

[10] Wojciech Szpankowski. Packet Switching in Multiple Radio Channels: Analysis and Stability of a Random Access System. *Computer Networks: The International Journal of Distributed Informatique*, 7(1):17–26, February 1983.

[11] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, 1996.

[12] Harald Vogt. Efficient Object Identification with Passive RFID Tags. In *International Conference on Pervasive Computing*, LNCS. Springer-Verlag, 2002.

[13] Roy Want, Kenneth P. Fishkin, Anuj Gujar, and Beverly L. Harrison. Bridging Physical and Virtual Worlds with Electronic Tags. In *Proceeding of the CHI 99 Conference on Human Factors in Computing Systems: the CHI is the Limit*, pages 370–377. ACM Press, 1999.

[14] Jeffrey E. Wieselthier, Anthony Ephremides, and Larry A. Michaels. An Exact Analysis and Performance Evaluation of Framed ALOHA with Capture. *IEEE Transactions on Communications*, COM-37, 2:125–137, 1989.