Submitted version

High-speed readout method of ID information on a large amount of electronic tags

Wataru Nagate [a] and Masahiro Sasabe [b] and Hirotaka Nakano [b]

[a]Graduate School of Information Science and Technology, Osaka University
1-32 Machikaneyama-cho, Toyonaka-shi Osaka 560-0043, Japan;
[b]Cybermedia Center, Osaka University
1-32 Machikaneyama-cho, Toyonaka-shi Osaka 560-0043, Japan

ABSTRACT

An electronic tag such as RFID is expected to create new services that cannot be achieved by the traditional bar code. Specifically, in a distribution system, simultaneous readout method of a large amount of electronic tags embedded in products is required to reduce costs and time. In this paper, we propose novel methods, called Response Probability Control (RPC), to accomplish this requirement. In RPC, a reader firstly sends an ID request to electronic tags in its access area. It successes reading information on a tag only if other tags do not respond. To improve the readout efficiency, the reader appropriately controls the response probability in accordance with the number of tags. However, this approach cannot entirely avoid a collision of multiple responses. When a collision occurs, ID information is lost. To reduce the amount of lost data, we divide the ID registration process into two steps. The reader first gathers the former part of the original ID, called temporal ID, according to the above method. After obtaining the temporal ID, it sequentially collects the latter part of ID, called remaining ID, based on the temporal ID. Note that we determine the number of bits of a temporal ID in accordance with the number of tags in the access area so that each tag can be distinguishable. Through simulation experiments, we evaluate RPC in terms of the readout efficiency. Simulation results show that RPC can accomplish the readout efficiency 1.17 times higher than the traditional method where there are a thousand of electronic tags whose IDs are 128 bits.

Keywords: RFID, access area, temporal ID

1. INTRODUCTION

Bar code has been the most popular technique to distinguish multiple objects. However, it cannot accomplish to read multiple objects simultaneously. Radio Frequency Identification (RFID) system [1] is expected to create new services that cannot be achieved by the bar code system. The RFID system consists of a reader and multiple electronic tags each of which is embedded in an object. One important feature of RFID system is the power supply to the electronic tag. Passive tags do not their own power supply, and therefore all power required for the operation of a passive tag must be drawn for the electrical/magnetic field of the reader. Conversely, active tags incorporate a battery that supplies all or part of the power for the operation.

Compared with the bar code system, the RFID system has two kinds of crucial advantages[2][3]. First, an electronic tag has a memory which enables to store much data than a bar code. Next, a reader can collect information from an electronic tag via a wireless communication even if there are obstacles between them. Despite of the attractive advantages, simultaneous readout method of numerous electronic tags has been an open issue.

Multi-access to numerous electronic tags can achieve new services in various industrial fields: stock and physical distribution controls in manufacturing fields, anti-counterfeit of bills or securities in financial fields, etc. However, a reader cannot simultaneously read information from multiple tags at once because a collision occurs among responses from them.

Further author information: (Send correspondence to Wataru Nagate)
Wataru Nagate: E-mail: w-nagate@ist.osaka-u.ac.jp, Telephone: +81-6-6850-6863

To prevent such collisions, an anti-collision mechanism is required. In the above mentioned services, numerous electronic tags will be needed. Since the cost of a passive tag is relatively smaller than that of an active tag, the passive tags are employed taking into account the introduction cost. Because the processing power and the antenna performance of the passive tag are limited, it is impractical for passive tags to avoid the collisions by sensing career each other. Consequently, the reader must accomplish an anti-collision mechanism such that it can read information from multiple tags one by one.

Dynamic Framed Slotted ALOHA (DFSA)[4,5] is one of the anti-collision mechanisms. DFSA is based on Slotted ALOHA.[6,7] In DFSA, the reader constructs a frame which consists of multiple slots. In a frame, each tag sends its own information to the reader at a slot randomly selected. If only one tag responds at one slot, the reader can read the information. Otherwise, a collision or timeout occurs. To reduce the latter case, the size of the frame is the most important in DFSA. To tackle this problem, the reader determines an appropriate size of a frame by estimating the number of tags in its access area. However, there are some problems in DFSA. In DFSA, all slots must have the same length. If the data size of each electronic tag is different, the length of each slot becomes large so that the reader can read information from an electronic tag whose data size is maximum in the access area. Furthermore, the reader cannot skip a slot to maintain the synchronization among tags even if none of the tags respond at the slot. Finally, all information is lost when a collision occurs. This deteriorates the system performance when the slot size becomes large.

In this paper, we propose RPC (Response Probability Control) method that enables to effectively gather information from multiple electronic tags independent of the number of electronic tags. In RPC, a reader firstly sends an ID request to electronic tags in its access area. It successes reading information on a tag if one of them only responds to the request. To avoid that multiple tags respond to the same ID request, the reader appropriately controls a probability that an electronic tag responds to an ID request (i.e., response probability) in accordance with the number of tags. In addition, to reduce the amount of ID lost by a collision, we divide the ID registration process into two steps. The reader first collects the former part of an original ID, called temporal ID, based on the approach using the response probability. Then, it gathers the latter part of the original ID, called remaining ID, from the node designated by the obtained temporal ID. We first make quantitative evaluations of RPC by mathematical analyses. Then, we compare RPC with DFSA through several simulation experiments. Note that we exclude coupling effect and capture effect for simplicity as in DFSA.

Section 2 presents the details of RPC. Next, we conduct simulation experiments to evaluate RPC in section 3. Finally, we describe conclusion and future works in section 4.

2. PROPOSED METHOD

2.1. Overview

We first explain the overview of our anti-collision protocol, RPC. In a RFID system, an electronic tag maintains its ID and data about the product in which the tag is embedded. A reader first collects the IDs from all tags in its access area, then gathers data from them. Since it does not know which ID exists in the area in advance, it should collects the IDs of them one by one with a probabilistic way. Although DFSA employs Slotted ALOHA, RPC controls a probability that an electronic tag responds to an ID request, i.e., response probability. The reader sends an ID request including a response probability to all tags in the area. Each tag responds to the ID request with the response probability. As a result, three kinds of cases occur: success, timeout, and collision. Success is the case that only one tag responds to the ID request. If no tag responds to the ID request, the reader detects timeout. The reader detects a collision when more than one tags respond to the same ID request. We assume that the reader can detect the collision using error detecting code. To suppress that timeout or collision case occurs, the response probability must be appropriately determined taking into account the number of tags in the access area. We describe how to estimate the number of tags in the area in subsection 2.3 and how to determine and control the response probability in subsection 2.4.

We further divide the registration process of an ID into two steps. The reader first collects a temporal ID from a tag with the above mentioned probabilistic way. The temporal ID is generated by the original ID so that the reader can distinguish each tag in its access area. The number of bits required to distinguish tags in the access area is often smaller than that of the original ID. For example, we can distinguish a thousand tags with only 10 bits if the temporal IDs are well distributed. If the reader can obtain a temporal ID, then it directly requests a remaining ID to the tag designated by the temporal ID. This approach has two advantages. First, we can reduce the number of bits lost due to a collision. Next, a tag can know whether its temporal ID is registered to the reader by receiving a request to its remaining ID. The reader can implicitly abandon that a tag already registered will respond to succeeding requests of temporal IDs. The details of the division of the ID registration process are given in subsection 2.5.

Finally, we derive the completion time for the ID registration in subsection 2.6.

2.2. Commands exchanged between a reader and electronic tags

Before explaining the details of RPC, we first introduce commands exchanged between a reader and electronic tags as follows.

- Commands sent from a reader to electronic tags
  - Temporal ID request
    - This command is used to obtain a temporal ID from an electronic tag. It includes a response probability with which each tag responds to this command.
  - Remaining ID request
    - This command is used to collect the remaining ID from the electronic tag that responded the last temporal ID request.
- Commands sent from an electronic tag to a reader
  - Temporal ID response
    - This command is used to send a temporal ID to a reader. It includes an error detecting code so that the reader can detect a collision.
  - Remaining ID response
    - This command is used to send a remaining ID to a reader.

2.3. Estimation of the number of tags

In general, a reader cannot know the number of tags in the access area, which are not registered to it, when it initially starts to gather information from them. As mentioned before, the reader should appropriately control the response probability to improve the success ratio of ID registration. In this subsection, we propose a method to estimate the number of unregistered tags in the access area based on the previous result of the ID registration process.

The result of the ID registration process is categorized in three cases: success, timeout, and collision. Success indicates that the reader can appropriately estimate the number of unregistered tag in the area. If timeout occurs, we suspect that the estimation is lower than the actual number of unregistered tags. On the contrary, a collision indicates that the estimation exceeds the actual number of unregistered tags. The details of the estimation algorithm are following.

1. When a reader initially starts the information gathering, it determines the estimated number of unregistered tags, m, by using one of the following methods.
   - The reader sets a random value to m.
   - If a camera monitoring the access area is available, the reader determines m based on the number of tags that obtained from the camera.

   By using the following adaptive mechanism, the initial value of m is not so critical to the system performance.

2. When a reader sent a temporal ID request, it adjusts m based on the result of the last ID registration process as follows.
    - Success case in which only one unregistered tag responded to the last temporal ID request.
        - The reader expects that m was approximately equal to the actual number of tags, n, and reduces m by one.
    - Timeout case in which no unregistered tag responded to the last temporal ID request.
        - The reader suspects that m is underestimated compared with n, then it sets $m \leftarrow m * C_d$. C_d is a control parameter that ranges (0,1). C_d represents the trade-off between accuracy of the estimation and adaptability to changes of n.
    - Collision case in which two or more unregistered tags simultaneously responded the last temporal ID request.
        - The reader imagines that m is overestimated than n, then it adjusts as follows: $m \leftarrow m * C_i$. C_i is a control parameter that is larger than one. As in C_d, C_i indicates the trade-off between the accuracy and the adaptability.

    The reader continues Step 2 until it finishes collecting IDs from all tags in the access area.

2.4. Decision of response probability

In this subsection, we describe how the reader determines a response probability P_{rsp} based on m to improve the efficiency of ID registration. We first derive a probability that each case, i.e., success, timeout, or collision, occurs when the reader sends a temporal ID request. We should here note that the probability is a function of not m but rather than n because the actual number of unregistered tags is n. The probability $P_0(n)$ that no tag does not respond to the temporal ID request, i.e, timeout, is

$$P_0(n) = (1 - P_{rsp})^n. \tag{1}$$

The probability $P_1(n)$ that only one tag responds to the temporal ID request, i.e., success, becomes

$$P_1(n) =_n C_1 P_{rsp}(1 - P_{rsp})^{n-1}. \tag{2}$$

Finally, the probability $P_{2+}(n)$ that two ore more tags simultaneously respond to the temporal ID request is as follows.

$$P_{2+}(n) = 1 - (1 - P_{rsp})^n -_n C_1 P_{rsp}(1 - P_{rsp})^{n-1} \tag{3}$$

Based on these equations, we calculate an optimal value of P_{rsp} in the next subsection. Then, we analyze the lower bound of $P_1(n)$ with the optimal value of P_{rsp} in subsection 2.4.2.

2.4.1. Optimal value of P_{rsp}

We derive the optimal value of P_{rsp} which maximize $P_1(n)$. By differentiating Eq. (2) with respect to P_{rsp}, we obtain the following equation.

$$\frac{dP_1(n)}{dP_{rsp}} = n(1 - P_{rsp})^{n-2}(1 - nP_{rsp}) \tag{4}$$

Since P_{rsp} ranges $[0, 1]$, $(1 - P_{rsp})$ is always positive. When $0 \leq P_{rsp} < \frac{1}{N_{rs}}$, Eq. (4) becomes positive. When $\frac{1}{N_{rs}} < P_{rsp} \leq 1$, Eq. (4) becomes negative. Thus, the optimal value of P_{rsp} is following.

$$P_{rsp}^{opt} = \frac{1}{n} \tag{5}$$

Since a reader cannot know n in an actual situation, it sets P_{rsp} to $\frac{1}{m}$. We expect that the estimation method in subsection 2.3 can reduce the difference between n and m.

2.4.2. Analysis of lower bound of $P_1(n)$

We analyze the lower bound of $P_1(n)$ by assuming that n reaches to infinity. Since a reader sets P_{rsp} to $\frac{1}{m}$, Eq. (2) becomes as follows.

$$P_1(n) = \frac{n}{m}\left(1 - \frac{1}{m}\right)^{n-1} \tag{6}$$

Equation 6 is maximized when $n = m$, that is, the reader can correctly estimate the number of unregistered tags in the access area. In this case, $P_1(n)$ becomes as follows.

$$P_1^{opt}(n) = \left(1 - \frac{1}{n}\right)^{n-1} \tag{7}$$

Suppose that $n \rightarrow \infty$ and $x = -\frac{1}{n}$, we obtain the lower bound of $P_1(n)$.

$$\begin{aligned}\lim_{n\rightarrow\infty} P_1^{opt}(n) &= \lim_{n\rightarrow\infty}\left(1 - \frac{1}{n}\right)^{n-1} \\ &= \lim_{x\rightarrow 0}\{(1+x)^{\frac{1}{x}}\}^{-1}(1+x)^{-1} \\ &= \frac{1}{e} \end{aligned} \tag{8}$$

Equation 8 indicates that RPC can collect temporal IDs from tags with a constant probability of $\frac{1}{e}$ even when the number of tags in the access area becomes enormously large. We can also obtain $P_0(n)$ and $P_{2+}(n)$ when $n \rightarrow \infty$ as follows.

$$\lim_{n\rightarrow\infty} P_0(n) = \frac{1}{e} \tag{9}$$

$$\lim_{n\rightarrow\infty} P_2(n) = 1 - \frac{2}{e} \tag{10}$$

Figure 1 illustrates transitions of $P_1(n)$ when $m = 10, 100, 1000$. We find that the maximum value of $P_1(n)$, that is $P_1^{opt}(n)$, is almost the same independently of m.

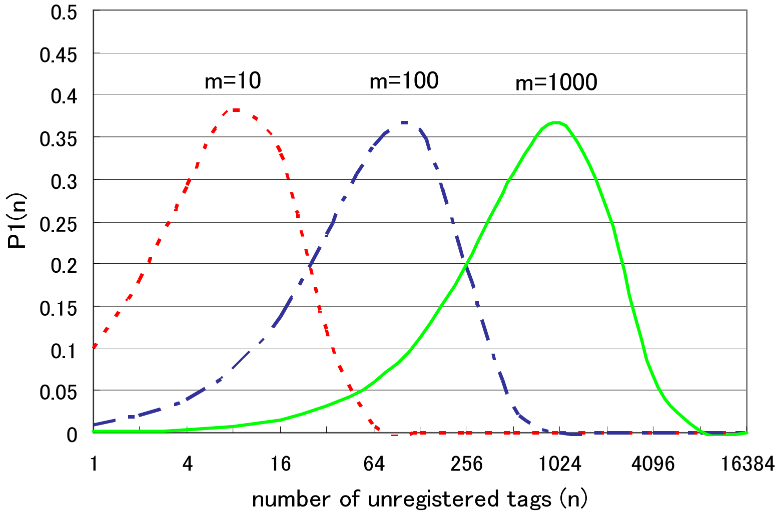

Figure 1. $P_1(n)$ vs. n

2.5. Division of ID registration process

We divide the registration process of an original ID into two steps: registrations of a temporal ID and the corresponding remaining ID. At first, a reader sends a temporal ID request which includes P_{rsp} to all tags in the access area. If the reader receives a temporal ID response from only one tag, it directly sends a remaining ID request to the tag designated by the temporal ID. By reducing the size of a temporal ID as possible, we can alleviate the time wasted by timeout and collision.

The number of bits consisting an original ID is typically 64 or 128 bits while that required to distinguish multiple tags in the access area seems to be much smaller. For example, we can distinguish a thousand tags with only 10 bits if the temporal IDs are well distributed. However, an original ID consists of a hierarchical bit pattern in many cases. This feature makes it difficult to use the former part of the original ID as a temporal ID because it may increase the size of a temporal ID. One possible way to solve this problem is to encrypt an original ID in advance because the encryption may reduce the regularity of the bit pattern of the original ID.

Next, we discuss what extent the effectiveness is improved by the division of the ID registration process. For simplicity, we ignore the overhead caused by the division in the following discussion. We define α as the ratio of the size of a temporal ID to that of an original ID. From Eq. 8, the expected ratio of the original ID received at the reader, E_{rd}, becomes as follows.

$$\begin{aligned} E_{rd} &= \frac{1}{e}\alpha + (1-\alpha) \\ &= \frac{1}{e}(\alpha + e(1-\alpha)) \end{aligned} \tag{11}$$

Consequently, E_{rd} is $\alpha + e(1-\alpha)$ times larger than that in the case without the division of the ID registration process. When $\alpha = 0.156$, namely the size of a temporal ID is 20 bits, we get $\alpha + e(1-\alpha) \simeq 2.43$.

2.6. Analysis of completion time for ID registration

In this subsection, we analyze the time t_f that the reader finishes collecting IDs from all tags in the access area. The number of unregistered tags at time t, $n(t)$, can be derived as follows:

$$n(t) \quad = \quad n(0) - \int_0^t \left(\frac{1}{\frac{T_1}{P_1^{opt}(n)} + T_2} \right) ds, \tag{12}$$

where T_1 and T_2 is average time required for registration of a temporal ID and a remaining ID, respectively. As shown in Fig. 1, we can approximate $P_1^{opt}(n)$ as $\frac{1}{e}$. Thus, Eq. 12 results in the following.

$$n(t) \quad \simeq \quad n(0) - \frac{t}{eT_1 + T_2} \tag{13}$$

t_f can be obtained by substituting 0 to $n(t)$ in Eq. 13.

$$t_f = n(0) * (eT_1 + T_2) \tag{14}$$

Note that we ignore the overhead caused by the division in the following discussion for simplicity.

We also should note here that the reader can recognize whether there is no more tags to be registered by sending a temporal ID request with $P_{rsp} = 1$. If the reader does not receive any responses, it finishes collecting IDs and starts to gather data from all tags in accordance with the list of IDs.

3. SIMULATION EXPERIMENTS

In this section, we conduct several simulation experiments to evaluate the performance of RPC by comparing with the traditional method DFSA. The system performance is evaluated by two kinds of criteria. One is readout time that the reader spends collecting IDs from all tags in its access area. Another is the sensitivity to the initial value in the estimation method proposed in section 2.3.

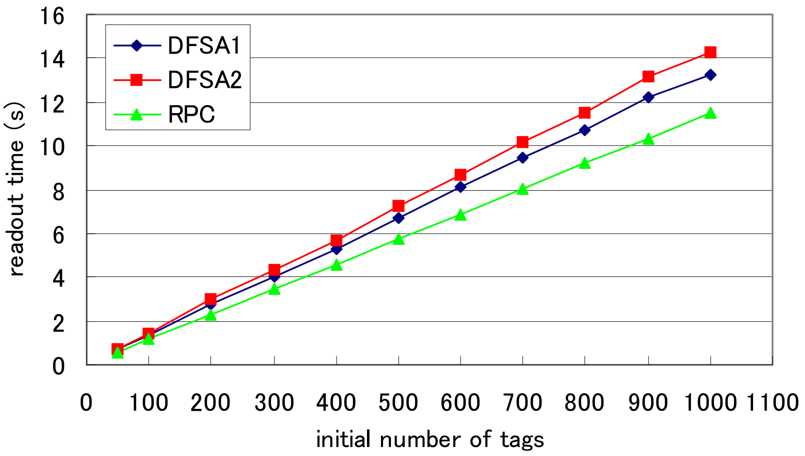

Figure 2. readout time vs. initial number of tags

3.1. Simulation settings

We evaluate in the following simulation environment taking into account the RFID system (ISO15693) in which data rate is 26 Kbps. We first explain the parameter settings of RPC. The transmission time of the commands: temporal ID request and remaining ID request, is set to 1 ms. The size of temporal ID and remaining ID is set to 20 and 180 bits, respectively. As a result, the transmission time of temporal ID and remaining ID become 1 and 5 ms. Note that it includes the overhead required for dividing the original ID. Timeout is set to 1 ms, which means that the reader waits for a temporal ID response for 1 ms after it sent a temporal ID request. We set C_d and C_i to 0.9462 and 1.08 in accordance with the results in our preliminary simulation experiments.

Next, we describe the parameter settings of DFSA. The transmission time of commands used to request an ID is set to 1 ms as in RPC. The slot size is set to 5 ms which is 1 ms shorter than the total transmission time of an ID in RPC. DFSA has an additional command to notify an acknowledgement (ACK) to a tag so that the tag will not respond the succeeding ID requests. By assuming the ideal and realistic situations, we set the transmission time of the additional command to 0 and 1 ms, respectively. We call the ideal case as DFSA1 and another as DFSA2 in the following results.

3.2. Evaluation of readout time

Figure 2 illustrates the transitions of readout time of RPC, DFSA1, and DFSA2 when the initial number of tags in the access area varies from 50 to 1000. In this scenario, we assume an ideal situation where the reader can precisely estimate the initial number of tags in the access area when the system starts.

As shown in Fig. 2, the readout time linearly increases regardless of the methods. However, RPC constantly overcomes both DFSA1 and DFSA2. Specifically, the readout time of RPC is 1.17 and 1.26 times faster than that of DFSA1 and DFSA2, respectively. In DFSA1 and DFSA2, the reader wastes 5 ms every time timeout occurs. On the other hand, RPC reduce the waste of time to 1 ms by introducing the dividing of the original ID. Furthermore, DFSA2 that is a realistic version of DFSA requires extra 1 ms to send ACK to the tag that responded to the last ID request. In RPC, the remaining ID request can play the role while gathering the remaining ID.

As not shown in this paper, we also expect that the readout time of RPC becomes shorter in an environment where IDs of different size, e.g., 64 and 128 bits, coexists in the access area. On the contrary, DFSA cannot improve the readout time because it must set the slot size so that an ID of the maximum size can be transmit in the slot.

Table 1. analyzed readout time vs. experimental readouttime in RPC

initial number of tags	100	200	300	400	500	600	700	800	900	1000
analyzed readout time (t_f)	1.144	2.287	3.431	4.574	5.718	6.862	8.005	9.149	10.292	11.436
experimental readout time	1.159	2.271	3.482	4.561	5.792	6.866	8.069	9.241	10.354	11.503
difference (%)	1.363	0.696	1.465	0.303	1.274	0.061	0.796	0.993	0.597	0.582

Table 2. readout time vs. initial estimated number of tags

initial estimated number of tags	100	200	300	400	500	600	700	800	900	1000
readout time of DFSA1 (s)	14.55	14.57	14.21	14.46	13.63	14.12	13.81	13.53	13.36	13.05
readout time of RPC (s)	11.54	11.45	11.56	11.57	11.57	11.46	11.46	11.61	11.51	11.45

Next, we evaluate the validity of the analysis of t_f in subsection 2.6. T_1 and T_2 are set to 2 and 6 ms by taking into account the simulation settings in subsection 3.1. Table 1 presents the analyzed readout time, T_f, and experimental readout time when the initial number of tags varies from 100 to 1000 in increments of 100. We find that there are almost no differences between them regardless of the initial number of tags. Thus, Eq. 14 can precisely calculate the readout time.

3.3. Evaluation of sensitivity to parameter setting in estimation method

In an actual situation, the reader does not necessarily know the initial number of tags in the access area. In this subsection, we evaluate what extent the estimation error of the initial number of tags increases the readout time. We set the initial number of tags to 1000. Table 2 presents the readout time of RPC and DFSA1 when the initial estimated number of tags varies from 100 to 1000 in increments of 100.

As shown in Tab. 2, the readout time of DFSA1 increases 1.52 sec at the maximum while that of RPC increases only 0.16 sec in the worst case. Since DFSA is based on Slotted ALOHA, it estimates the number of tags at the beginning of each frame. The larger the initial number of tags is, the more the estimation error affects the readout time. On the other hand, RPC conducts the estimation for each temporal ID request, thus it can improve the accuracy of the estimation compared with DFSA. We also find that RPC is not sensitive to the degree of the initial estimation error because the readout time of RPC does not almost change. Actually, the variance of the readout time of RPC is 0.0032 that is much smaller than that of DFSA1, i.e., 0.257.

4. CONCLUSION

In this paper, we proposed RPC that is a high-speed readout method of ID information on a large amount of electronic tags. RPC is composed of three kinds of methods. First, we discussed how the reader appropriately estimates the number of tags in its access area. Then, we derived the optimal value of the response probability based on the estimation in an analytical way. Finally, we introduced the division of the ID registration process into two steps to shorten the readout time and reduce the amount of data lost by a collision. Through several simulation experiments, we evaluated the effectiveness of RPC by comparing with that of DFSA. Specifically, the readout time of RPC becomes 1.17 times faster than that of DFSA when the initial number of tags is 1000.

As future research works, we further evaluate RPC and DFSA in an environment where objects embedded electronic tags enter and leave the access area at a certain rate. A belt conveyor system used in a physical distribution system is one such example. In such a case, the reader should frequently estimate the number of tags in the access area to adapt the changes of system conditions. We expect that RPC is more suitable than DFSA in that case.

Submitted version

REFERENCES

1. K. Frinkenzeller, *RFID Handbook*, John wiley and Sons, 2003 (second edition).
2. R. Want, "An introduction to rfid technology," *IEEE Pervasive Computing* **5**, pp. 25–33, 2006.
3. A. Juels, "Rfid security and privacy: A research survey," *IEEE journal on selected areas in communications* **24**, 2006.
4. J.-R. Cha and J.-H. Kim, "Novel anti-collision algorithms for fast object identification in rfid system," in *ICPADS 2005*, **2**, pp. 63–67, 2005.
5. H.-S. Choi, J.-R. Cha, and J.-H. Kim, "Fast wireless anti-collision algorithm in ubiquitous id system," *IEEE Vehicular Technology Conference* **6**, pp. 4589–4592, 2004.
6. T. N. SAADAWI and A. EPHREMIDES, "Analysis, stability and optimization of slotted aloha with a fine number of buffered users," *IEEE Transactions on AUTOMATIC CONTROL* , pp. 680–689, 1981.
7. Y. Yang and T.-S. P. Yu, "Delay distributions of slotted aloha and csma," *IEEE Transactions on Communications* **51**, pp. 1846–1857, 2003.