LETTER

A Reliable Tag Anti-Collision Algorithm for Mobile Tags

Xiaodong DENG[†a)], **Mengtian RONG**[†b)], ***Nonmembers*, *and* Tao LIU**[†c)], ***Member***

SUMMARY As RFID technology is being more widely adopted, it is fairly common to read mobile tags using RFID systems, such as packages on conveyer belt and unit loads on pallet jack or forklift truck. In RFID systems, multiple tags use a shared medium for communicating with a reader. It is quite possible that tags will exit the reading area without being read, which results in tag leaking. In this letter, a reliable tag anti-collision algorithm for mobile tags is proposed. It reliably estimates the expectation of the number of tags arriving during a time slot when new tags continually enter the reader's reading area and no tag leaves without being read. In addition, it gives priority to tags that arrived early among read cycles and applies the expectation of the number of tags arriving during a time slot to the determination of the number of slots in the initial inventory round of the next read cycle. Simulation results show that the reliability of the proposed algorithm is close to that of DFSA algorithm when the expectation of the number of tags entering the reading area during a time slot is a given, and is better than that of DFSA algorithm when the number of time slots in the initial inventory round of next read cycle is set to 1 assuming that the number of tags arriving during a time slot follows Poisson distribution.

key words: *RFID, mobile tags, tag anti-collision algorithm, dynamic framed-slotted ALOHA, Poisson distribution*

1. Introduction

Radio Frequency Identification (RFID) is an identification system capable of automated tracking to provide real time inventory visibility, operational status and movement of assets through the supply chain. When more than one tag responds simultaneously to a reader's request in RFID systems, tag collision occurs. Many RFID tag anti-collision algorithms have been presented to resolve the tag collision problem. Among tag anti-collision algorithms, Dynamic Framed-Slotted ALOHA (DFSA) is one of the most representative algorithms. In DFSA, purposeful works have been carried out on tag estimation. The lower bound of tag number can be conservatively determined as double the number of collided slots. Schoute's tag estimation method in [1] is largely based on the number of collisions in an inventory round. The number of tags is estimated by multiplying the number of collision slots in an inventory round by the expectation of collided tag numbers in slots, which is the same for all inventory rounds regardless of various numbers of tags and the number of time slots. Tag estimation algorithms proposed by Vogt [2], Chen [3] and Cha [4] are deduced from the success, collision and idle probability in an inventory round, but require complex computations.

Manuscript received September 2, 2011.
Manuscript revised January 18, 2012.

[†]The authors are with Electronic Engineering, Shanghai Jiao Tong University, China.
a) E-mail: dxd@sjtu.edu.cn
b) E-mail: rongmt@sjtu.edu.cn
c) E-mail: ttlyz@sjtu.edu.cn

DOI: 10.1587/transinf.E95.D.1527

It is fairly common to read tags when they are moving, such as packages on conveyer belt and unit loads on pallet jack or forklift truck. In this letter, a reliable tag anti-collision algorithm for mobile tags is proposed. It reliably estimates the expectation of the number of tags arriving during a time slot when new tags continually enter reader's reading area and no tag leaves without being read. In addition, it gives priority to tags arrived early among read cycles and applies the expectation of the number of tags arriving during a time slot to the determination of the number of slots in the initial inventory round of the next read cycle. Simulation results show that the reliability of the proposed algorithm is close to that of DFSA algorithm when the expectation of the number of tags entering the reading area during a time slot is a given, and is better than that of DFSA algorithm when the number of time slots in the initial inventory round of next read cycle is set to 1 assuming that the number of tags arriving during a time slot follows Poisson distribution. The letter concludes with some further improvements of the proposed algorithm.

2. The Reliable Tag Anti-collision Algorithm

There is a reader's reading area which has an overall length of D meters shown in Fig. 1. Generally, the outline of a reader's reading area is irregular, but is illustratively expressed by a rectangle for the sake of simplicity. It is assumed that tags enter reader's reading area from the lower side in chronological order, and get across the reading area at a constant speed of V in meters per second. Finally, they leave the reading area at the upper side. Therefore, the dwell time of a tag, specifically T_D in seconds, is expressed by D/V. In addition, all tags are attached to a carrier which is L meters in length. There will be new tags entering reader's

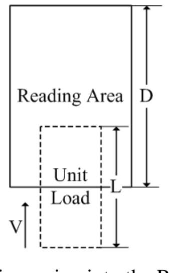

Fig. 1 An Unit Load is coming into the Reading Area of a reader.

reading area within a time $T_L = L/V$ in seconds. Suppose that the number of new tags coming into reader's reading area in every time slices complies with same probability distribution and the probability of there being k new tags during a time slice can be expressed by $P\{X = k\}$, the expectation of the number of new tags during a time slice is given by $E(X) = \sum_k kP\{X = k\}$.

Leaking of tags hardly happens and the reading rate is close to 100% if the tag anti-collision algorithm always timely read all new tags of the preceding time slices, in which case tag reading may be delayed, but average delay time is still considerably less than T_D. However, as time proceeds, new tags quickly accumulate until no tag comes into the reading area any more, specifically when time comes to T_L. Still worse, the foremost tag which arrived earliest is about to leave the reading area at T_D, then succeeding tags are bound to leave as a result of a finite dwell time of the tag, as time goes by. Leaking of tag occurs if the tag anti-collision algorithm fails to timely allocate a time slot to the tag about to leave for communicating with the reader. Therefore, intuitively speaking, a reliable tag anti-collision algorithm for reading mobile tags should put top priority to tags arrived earlier.

A read cycle is traditionally defined as the procedure for completely reading a certain number of tags in DFSA. A read cycle is composed of at least one inventory round. All available tags simultaneously join the time slot competition at the beginning of the initial inventory round in a read cycle. The outcome of competing for time slots can be grouped into three categories in terms of how many tags replied during a time slot: empty slot, successful slot and collided slot. In successful slots, the reader succeed in reading a tag due to a single tag reply. Tags will keep silent and not join any inventory rounds after they are read. Empty and collided slots end up with no inventoried tags because there is no tag reply and more than one tag reply respectively. A read cycle concludes if there is no collided slot at the end of an inventory round. Otherwise, another inventory round will be arranged to read collided tags until the number of collided slots is zero. It should be noted that tag estimation algorithm is carried out at the beginning of subsequent inventory rounds to determine the number of slots in the next inventory round in terms of optimal throughput rate which can be achieved when the number of slots is identical to the number of tags [1]. In DFSA, for example, the number of collided tags can be conservatively estimated as $2N_c$, where N_c is the number of collided slots. It is also roughly equals to $2.39N_c$ if the number of slots is set to the number of tags [1]. It can also be given by rN_c, where r requires complex calculation [5]. However, the number of slots in the initial inventory round can not be determined by tag estimation algorithm in DFSA. Generally, it is set to 1, which has proved a significant handicap to the performance of tag anti-collision algorithms [6].

No tags are available to the reader at the beginning because tags enter reader's reading area in chronological order. Which tags are available depends on time. It is assumed that the duration of a time slot in the tag anti-collision algorithm equals to the duration of a time slice for simplicity and can be given by T_s, and the number of new tags can be given by $N_{t,1}$ during $(0, 1]$ slot. They are all available to the reader. Consequently, the tag anti-collision algorithm carries out its duty to read all new tags. To enhance reliability, the inventory of these $N_{t,1}$ tags is a top priority. Therefore, a read cycle is arranged. As a result, it takes $N_{s,1}$ slots to fully read $N_{t,1}$ tags. The reader only gets a chance to check the availability of new tags after $N_{s,1}$ slots because no tag can be introduced during the execution of a read cycle. Suppose the number of tags came into the reading area among $(1, N_{s,1}]$ slots is given by $N_{t,2}$, a new read cycle is arranged to read these $N_{t,2}$ tags. $(i + 1)_{th}$ read cycle can be arranged in the same manner. To sum up, it takes $N_{s,i}$ slots to fully inventory $N_{t,i}$ tags. If $E(X)$ is a given, $E(N_{t,2}) = E(X)N_{s,1}$ If the initial value of N_s, specifically $N_{s,0}$, is set to 1, the expectation of the number of new tags in $(i + 1)_{th}$ read cycle can be express by

$$E(N_{t,i+1}) = E(X)N_{s,i} \tag{1}$$

Evidently, $E(N_{t,i+1})$ is a better choice as the number of slots of the initial inventory round of the $(i + 1)_{th}$ read cycle, or $N_{s,i+1,1}$, than 1 as long as new tags keep coming into the reading area. Of course, it becomes invalid when there is no new tag coming any more, specifically $\sum_{i=0}^{I} N_{s,i} > T_L$. Then,

$$E(N_{t,I+1}) = E(X)\left(T_L - \sum_{i=0}^{I-1} N_{t,i}\right) \tag{2}$$

is a reasonable alternative as $N_{s,I+1,1}$. It should be noted that the $(I + 1)_{th}$ will the last read cycle because no new tags are available. Therefore, all available unread tags should be handled in the $(I + 1)_{th}$ read cycle.

Generally, the expectation of the number of slots in $(i+1)_{th}$ read cycle can be given as a function of $N_{t,i+1}$.

$$E(N_{s,i+1}) = f(N_{t,i+1}) \tag{3}$$

if no tag leaves the reading area during the execution of a read cycle. $E(N_{s,i+1})$ rises monotonically as $N_{t,i+1}$ increases. Specifically, if optimal throughput rate is achieved among inventory rounds in the $(i + 1)_{th}$ read cycle, $E(N_{s,i+1})$ for multiple tags can be expressed by

$$E(N_{s,i+1}) \approx eN_{t,i+1} \tag{4}$$

where e is the base of natural logarithm [5]. If expectation operation is performed on both sides of Eq. (4) and $E(N_{t,i+1})$ is substituted by Eq. (1), $E(N_{s,i+1})$ can be expressed as

$$E(N_{s,i+1}) \approx eE(X)N_{s,i} \tag{5}$$

Equation (5) only works on condition that the number of tags in every inventory rounds are accurately estimated. Otherwise, $eE(X)N_{s,i}$ could be the lower bound of $E(N_{s,i+1})$.

It should be noted that priority is only given to tags in the fore part among read cycles, but tags still fairly compete with each other for time slots and tags arrived early is taken no priority over those arrived late inside a read cycle. The fundamental flaw in giving priorities to tags is hidden when tags are still rich in dwell time. However, leaking of tags occurs if tags are about to run out of dwell time. Since the tag anti-collision algorithm fails to intervene in the slot competition inside a read cycle, it is possible that every tag can be inventoried when the read cycle is about to conclude. Leaking of tags is likely to occur when the foremost tag among unread tags can not stay longer than $E(N_{s,i+1})$.

$$T_D - T_{in} < E(N_{s,i+1}) \tag{6}$$

where T_{in} describes how long the foremost tags among unread tags stayed in the reading area. All unread tags entered the reading area during the i_{th} read cycle when $(i+1)_{th}$ read cycle is going to start. Therefore, the unread tags stay in the reading area $[1, N_{s,i})$ slots,

$$T_{in} \leq N_{s,i} \tag{7}$$

Substituting T_{in} in Eq. (6) by Eq. (7), we obtain

$$T_D < E(N_{s,i+1}) + N_{s,i} \tag{8}$$

Substituting $E(N_{s,i+1})$ by Eq. (5) if $\sum_{i=0}^{I} N_{s,i} < T_L$, we obtain

$$\begin{aligned} &T_D < E(N_{s,I+1}) + N_{s,I} \approx (1 + eE(X))N_{s,I} \\ &N_{s,I} > \frac{T_D}{1 + eE(X)} \end{aligned} \tag{9}$$

Substituting $E(N_{s,I+1})$ by Eq. (4) and Eq. (2) if $\sum_{i=0}^{I} N_{s,i} > T_L$, we obtain

$$\begin{aligned} &T_D < E(N_{s,I+1}) + N_{s,I} \approx eE(X)\left(T_L - \sum_{i=0}^{I-1} N_{t,i}\right) + N_{s,I} \\ &N_{s,I} > T_D - eE(X)\left(T_L - \sum_{i=0}^{I-1} N_{t,i}\right) \end{aligned} \tag{10}$$

The leaking of tags is likely to occur in the $(I+1)_{th}$ read cycle if either Eq. (9) or Eq. (10) is satisfied.

As mentioned above, it is assumed that $E(X)$ is known during the explanation. Actually, $E(X)$ could be reliably estimated by $\bar{x}$. $\bar{x}$, the average of x_l $(0 < l < n)$ can be calculated after m read cycles, even through the value of x_l in each slot is unknown.

$$\begin{aligned} &\bar{x} = \frac{\sum_{i=0}^{m} N_{t,i}}{n} \\ &n = \sum_{i=0}^{m-1} N_{s,i} \end{aligned} \tag{11}$$

$\bar{x}$ can be considered as a unbiased estimation of $E(X)$ if n is approximately close to infinite. Undoubtedly, $\bar{x}$ must be calculated when $n \leq T_L$. In addition, the accuracy of estimation is totally independent of whether read cycles are optimal or not, but it depends heavily on that there is no leaking of tags. Therefore, $\bar{x}$ should be calculated before Eq. (9) is satisfied.

3. Evaluation

In the evaluation, we assume that tags enter the reading area according to Poisson distribution with λ.

$$P\{X = k\} = \frac{\lambda^k}{k!} e^{-\lambda} \tag{12}$$

where k can be any nonnegative integer. The expectation of the number of tags entering the reading area during a time slot can be given by $E(X) = \lambda$. In addition, the simulation results are obtained in *Matlab*.

The performance of tag estimation algorithm will be considered when it is assumed that no tag leaves during the execution of the $(i+1)_{th}$ read cycle. The mean value of $N_{s,i+1}$ is described as a function of $N_{s,i}$ shown in Fig. 2 for $\lambda = 0.5$, 1 and 2. Random numbers from the Poisson distribution with mean parameter λ are generated by *poissrnd* function in *Matlab*. *poissrnd*$(\lambda, 1, N_{s,i})$. The $(i+1)_{th}$ read cycle is arranged to read all tags came during the i_{th} read cycle. $N_{s,i+1,j}$ represents the number of slots in the j_{th} inventory round of the $(i+1)_{th}$ read cycle. $N_{s,i+1,1}$ is estimated as $\lambda N_{s,i}$ and $N_{s,i+1,j}$ $(j \geq 2)$ should be determined by tag estimation algorithm. The tag estimation algorithm proposed by Schoute assumes that

$$N_{s,i+1,j+1} = 2.39 N_{c,i+1,j} \tag{13}$$

where $N_{c,i+1,j}$ represents the number of collided slots in the j_{th} inventory round of the $(i+1)_{th}$ read cycle. In addition, the Lower-bound tag estimation algorithm assumes that

$$N_{s,i+1,j+1} = 2N_{c,i+1,j} \tag{14}$$

$N_{t,i+1,j}$ represents the actual number of available tags in

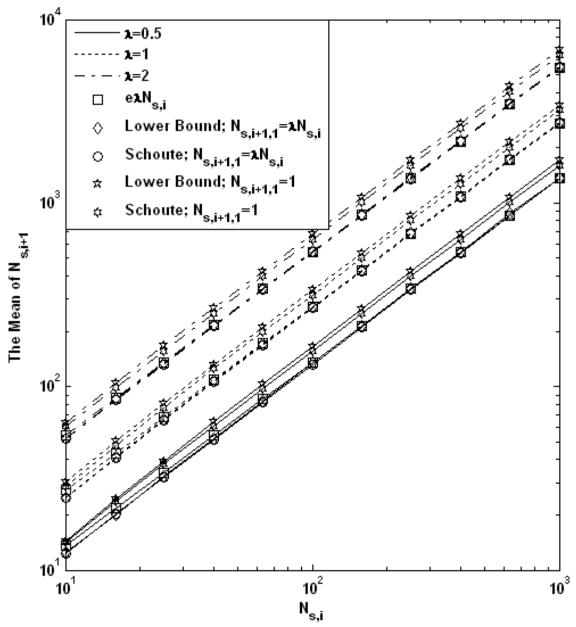

Fig. 2 The mean value of $N_{s,i+1}$ if no tag leaves.

the j_{th} inventory round of the $(i+1)_{th}$ read cycle. They are supposed to generate random numbers in the range of $[1, N_{s,i+1,j+1}]$ to compete for a time slot for communicating with the reader. Uniformly distributed pseudo-random integers on the interval $[1, N_{s,i+1,j}]$ are generated by *randi* function in *Matlab*. *randi*$(N_{s,i+1,j}, 1, N_{t,i+1,j})$. The outcome of competing for time slots can be obtained by checking the uniqueness of each number. The read cycle concludes when $N_{c,i+1,j} = 0$ and $N_{s,i+1} = \sum_j N_{s,i+1,j}$. In the evaluation, 1000 attempts are made for each $N_{s,i}$ to calculate the mean value of $N_{s,i+1}$. 1 is an acceptable choice if we have no idea of how many tags available at the beginning of each read cycle. By comparison with setting $N_{s,i+1,1}$ to 1, $\overline{N_{s,i+1}}$ is considerably reduced if $N_{s,i+1,1}$ is set to $\lambda N_{s,i}$ and is close to $e\lambda N_{s,i}$ regardless of Lower-bound or Schoute's tag estimation algorithm. Figure 2 shows that it takes less slots to read multiple tags if λ is a given and $\lambda N_{s,i}$ is used to determine the number of unread tags in the next read cycle in DFSA. The reduction of $\overline{N_{s,i+1}}$ hardly depends on the value of λ, because $\overline{N_{s,i+1}}$ rises almost linearly as the number of tags increases. But the reduction of $\overline{N_{s,i+1}}$ greatly depends on which tag estimation algorithm is used. Specifically, $\overline{N_{s,i+1}}$ is reduced by about 20 percent when Lower-bound tag estimation algorithm is used and by about 16 percent when Schoute's tag estimation algorithm is used.

Then, let us consider the case demonstrated in Fig. 1. The probability of leaking of tags, defined as the ratio of the number of unread tags to the number of all available tags, is expected after a number of tags came through a reader's reading area. In the evaluation, 1000 attempts are made to calculate the mean value of the probability of leaking of tags when T_L and T_D are given. Similarly, random numbers from the Poisson distribution with mean parameter λ are generated by *poissrnd* function in *Matlab*. *poissrnd*$(\lambda, 1, T_L)$. The random number represents how many new tags are coming into the reading area during each time slot in the range of $[1, T_L]$. In addition, $N_{s,i+1,1}$ is determined by Eq. (1) or Eq. (2) according to the relationship between T_L and $\sum_{i=0}^{I} N_{s,i}$. Schoute's tag estimation algorithm in [1] is used to determine $N_{s,i,j}$ $(j \geq 2)$. Equation (9) or Eq. (10) is used to determine whether leaking of tags occurs or not. After the I_{th} read cycle ends, Eq. (11) is used to estimate λ if $\sum_{i=0}^{I-1} N_{s,i} \leq T_L$ and $N_{s,I} \leq \frac{T_D}{1+e\lambda}$. The graphs of the probability of leaking of tags as a function of λ are shown in Fig. 3 for $T_D = T_L/2$, T_L and $2T_L$ when $T_L = 1000T_s$. This figure illustrates that the improvement of reliability depends on λ. If $T_D = 2T_L$, the probability of leaking of tags can be reduced from 3.9×10^{-2} to 6.25×10^{-4} when $\lambda = 0.9$. However, the probability of leaking of tags can only be reduced from 0.5 to 0.35 when $\lambda = 1.6$. If $T_D = T_L/2$ or T_L, the probability of leaking of tags varies according to λ in the manner of $T_D = 2T_L$. The longer T_D is, the lower the probability of leaking tags is when λ holds constant. Therefore, by increasing the ratio of T_D to T_L, one can increase λ required to achieve a given probability of leaking of tags. In addition, the reliability of the proposed algorithm is more or less the same with that of DFSA when λ is a given due to a reliable estimation of λ, and is much better than that of DFSA when λ is unknown and $N_{s,i+1,1}$ is set to 1.

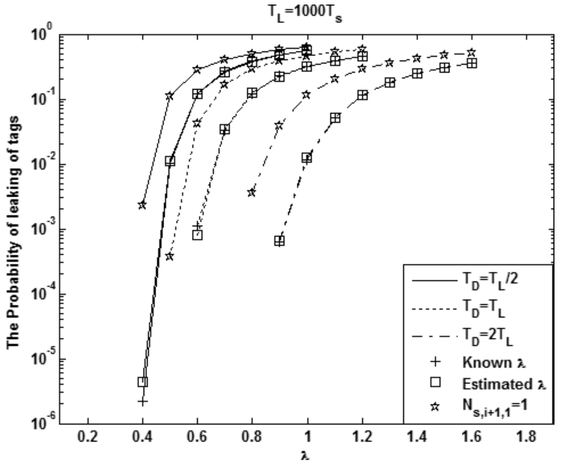

Fig. 3 Reliability comparison.

4. Conclusion

The proposed algorithm reliably estimates the expectation of the number of tags arriving during a time slot when new tags continually enter reader's reading area and no tag leaves without being read. It also applies the expectation of the number of tags arriving during a time slot to the determination of the number of time slots in the initial inventory round of the next read cycle. However, the reliability of the proposed algorithm could be further improved if top priority could be given to the tags arrived early again instead of dealing with all unread tags in a single read cycle regardless of their priorities, when the leaking of tags is likely to occur.

References

[1] F. Schoute, "Dynamic frame length aloha," IEEE Trans. Commun., vol.31, no.4, pp.565–568, April 1983.

[2] H. Vogt, "Efficient object identification with passive RFID tags," Pervasive Computing, pp.98–113, 2002.

[3] W.T. Chen and G.H. Lin, "An efficient anti-collision method for tag identification in a RFID system," IEICE Trans. Commun., vol.E89-B, no.12, pp.3386–3392, Dec. 2006.

[4] J.-R. Cha and J.-H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system," Proc. 11th International Conference, Parallel and Distributed Systems, 2005. vol.2, pp.63–67, July 2005.

[5] J.-B. Eom and T.-J. Lee, "Accurate tag estimation for dynamic framed-slotted aloha in RFID systems," IEEE Commun. Lett., vol.14, no.1, pp.60–62, Jan. 2010.

[6] W.-T. Chen, "An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length aloha," IEEE Trans. Automation Science and Engineering, vol.6, no.1, pp.9–15, Jan. 2009.