A Dynamic Detection Point Frame Length Adjustment Method for RFID Anti-collision

Xiaoning Feng[1](✉), Zhuo Wang[2](✉), Bijun Yan[1], Guangjie Han[3],
Feng Wang[4], and Xue Song[1]

[1] College of Computer Science and Technology, Harbin Engineering University,
Harbin 150001, China
fengxiaoning@hrbeu.edu.cn
[2] College of Shipbuilding Engineering, Harbin Engineering University,
Harbin 150001, China
wangzhuo@hrbeu.edu.cn
[3] Department of Information and Communication Systems, Hohai University,
Changzhou, China
hanguangjie@gmail.com
[4] School of Information Science and Engineering, Changzhou University,
Changzhou 213164, China
wfeng@cczu.edu.cn

Abstract. The tag collision problem is an important issue that affects the efficiency of RFID system. A method of frame length adjustment based on dynamic detection points is proposed in this paper. The method calculates the size of the detection slot area in the frame according to the calculation of the sample size. And then it determines the location of the detection point in the frame. So, the detection point is dynamically adjusted with the length of the frame. Compared with the fixed detection point method, simulation results showed that the dynamic detection point method can adjust the frame length more accurately and improve the system throughput and speed the tag identification.

Keywords: RFID · Anti-collision algorithm · Frame length adjustment
Detection point

1 Introduction

With the development of the identification methods, Radio Frequency Identification (RFID) technology has been widely used in various fields because of its advantages such as non-contact, rapid identification, small size and so on. Combined with Internet of Things, the ‘EPC C1 Gen2’ (EPC-C-G below) standard becomes the most widely used standard of RFID technology in recent years [1, 2]. In general, the RFID system consists of three parts: tags, readers and processing system. The tag collision occurs when a large number of tags simultaneously respond to one reader in RFID system. The problem of tag collision increases tag identification time, reduces tag identification rate and system throughput. Many anti-collision methods have been proposed to solve

Y.-B. Lin et al. (Eds.): IoTaaS 2017, LNICST 246, pp. 308–315, 2018.
https://doi.org/10.1007/978-3-030-00410-1_36

the problem. Most of them can be divided into three categories: tree-based, ALOHA based and hybrid algorithms [3].

Tree-based anti-collision algorithms consist of multiple periodic for reader's query commands and tag's responses. The set of tags was divides into two branches according to each collision's tag bit. The process will loop until each branch has only one tag. And all the tags can be identified. The typical tree-based algorithms include dynamic binary tree search algorithm, backward binary tree search algorithm and query tree algorithm.

Hybrid algorithms have the advantages of ALOHA-based algorithms and tree-based algorithms. But tree-based and hybrid anti-collision algorithms cannot be implemented by the EPC-C-G standard because the reader based on this standard cannot effectively identify collision tags based on the collision bits. Therefore, we have focused on the ALOHA-based anti-collision algorithm in this paper.

The main idea of ALOHA-based anti-collision algorithms is to make the tags to select different time slot when responding to the reader in order to reduce the probability of tag collision. And the algorithms mainly are pure ALOHA algorithm, slotted ALOHA (SA) algorithm, framed slotted ALOHA (FSA) algorithm and dynamic framed slotted ALOHA (DFSA) algorithm [4]. DFSA algorithm is the most efficient and widely adopted by RFID standards. The improvement of DFSA anti-collision algorithm for EPC-C-G standard is time slot random anti-collision algorithm called Q-algorithm. The performance improvement for the Q-algorithm can be accomplished in two aspects. One is how to adjust the frame length, and the other one is how to estimate the number of remaining tags.

The problem of how to adjust the frame length can be divided into two parts. One is how to match the optimal frame length for the remaining tags. The other one is when to adjust the frame length. On the basis of when to adjust the frame length the existing algorithms can be divided into three categories. According to the frame length needs to be adjusted slot by slot, partial slots or all slots in the frame. The [5] proposed a novel algorithm, access probability adjustment based fine-grained Q-algorithm (APAFQ) algorithm. The APAFQ scheme is driven by updating Q value with two different weights, slot by slot. However, the frame length is adjusted slowly in the algorithm because the frame length is adjusted slot by slot. The [6] proposed an early adjustment of frame length (EAFL) algorithm. This algorithm adjusts the frame length by partial slots or all slots in the frame. However, if the selection of partial slots is not appropriate, the early adjustment of frame length is lost. The [7] introduced an improvised dynamic frame slotted ALOHA (ID-FSA) algorithm. The algorithm adjusts the frame length based on the inventory results of all the slots in previous cycles. But the reader takes too many collision slots and empty slots.

In this paper, we presented a new algorithm based on dynamic frame slotted ALOHA. To obtain the number of slots that can represent the distribution of tags in the whole frame, the algorithm uses the sample size to calculate the size of the detection slots area in the frame so that the position of the detection point is adjusted with the frame length dynamically.

2 Algorithm Description

2.1 Dynamic Detection Point Frame Length Adjustment Method

Combining the idea of the frame length needs to be adjusted by partial slots in the frame, a method of frame length adjustment based on dynamic detection points is proposed in this paper. We want to calculate the number of slots that can accurately represent the distribution of the tags in the whole slots of frame. Thus, we can determine the location of the detection points in the frame. The degree of dispersion of the tags distribution in the frame is an important factor that affects the size of the detection region and can be expressed in terms of variance of the number of tags in the frame.

In the RFID system, tags can obtain the frame length information for the current inventory cycle after receiving the Query(Q) command sent by reader. And then they randomly select any slot in the frame to respond their own identity information. To obtain the variance of the number of tags in slots we need to know how many slots in each case and how many slots have tags when there are *n* tags. The probability of selecting any slot for each tag is the inverse of the frame length $1/\mathrm{L}$, under the condition of the inventory cycle with frame length L. Suppose $\mathrm{n}-1$ tags have chosen $\mathrm{A}_{\mathrm{n}-1}$ slots. There are two cases If adding a tag to choose those slots in the current frame. One case is that the tag selects those slots with tag success or collision. The probability is $\mathrm{A}_{\mathrm{n}-1}/\mathrm{L}$. The other is that the tag choose idle slot. The probability is $1-\mathrm{A}_{\mathrm{n}-1}/\mathrm{L}$. Then the value of A_{n} is shown in (1), (2) and (3).

$$A_n = \frac{A_{n-1}}{L} \times A_{n-1} + \left(1 - \frac{A_{n-1}}{L}\right) \times (A_{n-1} + 1). \tag{1}$$

$$A_n = 1 + \lambda \times A_{n-1}. \tag{2}$$

$$A_n = \frac{1-\lambda^n}{1-\lambda}. \tag{3}$$

where n is the number of tags, L is the length of frame, and $\lambda = (L-1)/L$.

The average slot number with x_i tags is shown in (4) and (5).

$$l_n^{x_i} = \left(1 - \frac{A_{n-1}}{L}\right) l_{n-1}^{x_i} + \frac{l_n^{x_i-1}}{L} \times \left(l_{n-1}^{x_i} + 1\right) + \frac{A_{n-1} - l_{n-1}^{x_i-1} - l_{n-1}^{x_i}}{L} l_{n-1}^{x_i} + \frac{l_{n-1}^{x_i}}{L} \times \left(l_{n-1}^{x_i} - 1\right). \tag{4}$$

$$l_n^{x_i} = \frac{\lambda^n}{1-\lambda} \binom{n}{x_i} \left(\frac{1-\lambda}{\lambda}\right)^{x_i}. \tag{5}$$

where x_i represents the number of tags which select the same slot, $i = 0, 1, 2, 3, \ldots, n$. When i is 0, x_0 stands for slot without tag selected, $l_n^{x_0}$ represents the number of idle slots. When i is 1, x_1 stands for slots with only one tag selected, $l_n^{x_1}$ represents the number of successful slots. When the i is greater than or equal to 2, x_i stands for a slot with i tags selected, $l_n^{x_i}$ represents the number of collision slots containing x_i tags.

In this paper, the relation between the optimal frame length and the number of tags is obtained by the theoretical maximum of the system throughput. Then the average number of tags allocated in each slot is 0.5. Then, the variance of the slot distribution in the frame is shown in (6).

$$\sigma^2 = \frac{(x_1 - \bar{x})^2 l_n^{x_1} + (x_2 - \bar{x})^2 l_n^{x_2} + \cdots + (x_n - \bar{x})^2 l_n^{x_n}}{n}. \tag{6}$$

where $\bar{x}$ represents the average number of tags in slots.

The selection of partial slots from the current frame representing all slots is analogous to a sampling survey of the population in order to infer some property or feature of the population. The frame is population, and the partial slots is sample, called the sample slots. To determine the sample slots size, we need to consider the degree of dispersion of the slot distribution in the frame, as well as the estimation error and the confidence level of infer result. Therefore, the formula for calculating the sample size of the sample slots in the frame is as follows in (7).

$$s = \frac{L\left(z_{\alpha/2}\right)^2 \sigma^2}{(L-1)d^2 + \left(z_{\alpha/2}\right)^2 \sigma^2}. \tag{7}$$

where the confidence level is $1-\alpha$, $z_{\alpha/2}$ is the standard error confidence level, d is the maximum allowable error, L is the frame length, which is population.

EPC-C-G standard specifies that frame length can only be the power of 2. The sample slots size in the frame is shown in Table 1.

Table 1. Time slots sample size in frame.

Q	3	4	5	6	7	8	9	10
Frame length L	8	16	32	64	128	256	512	1024
Sample slots size	6	10	12	16	24	34	56	64
Detection point	3/4	5/8	3/8	1/4	3/16	17/128	7/64	1/16

Reader determines whether the current frame length is optimal at the location of the detection point.

2.2 The Proposed Algorithm

In the process of identifying tags, reader first sends the Select command to select a set of specific tags that have been activated within its radio frequency range. After that, reader initializes the frame length by sending the Query(Q) command to those tags, where the frame length is 2Q.

Tags receive this command and get the frame length information. And then tags select a slot through its internal random number generator and send its Random Number 16 (RN16) when reader inventories this slot. Reader inventories slot, determines and records the type of slots. If there is a collision or empty slot, reader records the slot and continues inventorying. As for successful slot, reader not only needs to record the slot type but also needs to exchange data with the tag to get its information. Reader sends QueryRep command after each slot in order to reduce the random number of the tag. Thus, the tag is kept synchronous with the reader. In order to get the detection point i, reader should estimate the number of tags according to $n = (S + 2.17 \cdot C)L/i$ and obtains the Q value of the optimal frame length based on $Q_{opt} = round(log2(1.89 \cdot n))$. If the number of collision slots C equals 0, the tag identification is completed. Otherwise, there are tags which have not been identified and the reader continues to inventory.

The algorithm presented in this paper is described in pseudo code as followed.

```
Initialize Frame;
while (unidentified tags≠0)
Start inventory cycle and initialize different types of slot record values;
 while (slots without inventory≠0)
    Interrogate the tags in slot and record slot type;
    if Inventory the detection point
         then Estimate tags number and obtain the optimal frame length;
     if the current frame length optimal
         then Continue inventory;
     else Estimate remaining tags number, update Q and start a new inventory
cycle;
     end if
    else Continue inventory
    end if
 end while
end while
```

3 Simulation Results

We evaluated the number of collision slots and system throughput of the proposed algorithm and compared its performance with existing methods including DFSA, APAFQ and EAFL algorithm.

The simulation conditions in this paper are based on the EPC-C-G standard and referred to [7]. The main parameters used in the simulations are shown in Table 2. It also defines a basic duration for interrogator-to-tag signaling called Tari. The basic duration for interrogator-to-tag signaling to be 12.5 μs, this corresponds to a transmission speed of 80 kb/s. For the interrogator-to-tag link, the durations of data-0 and data-1 are one Tari and two Taris, respectively. On the other hand, the durations of data-0 and data-1 for tag-to-interrogator link are 6.25 μs, which corresponds to a link frequency of 160 kHz.

The number of collision slots is the quantities of all collision slots that are recorded by reader during the identification process for tags. Figure 1 displays the number of collision slots of four algorithms when the initial frame length is 128, which means the

Table 2. Simulation parameter values.

Parameters	Time interval (μs)
Reader-to-tag preamble	112.5
Reader-to-tag frame sync	62.5
Reader-to-tag data-0	12.5
Reader-to-tag data-1	25
Tag-to-reader preamble	112.5
Tag-to-reader data-0/data-1	6.25
Query(Q) command	412.5
QueryAdjust(Q) command	168.75
QueryRepeat command	75
ACK command	337.5
RN16 (preamble included)	212.5
PC+EPC+CRC16 (preamble included)	912.5
T_1	62.5
T_2	62.5
T_3	5
T_4	112.5

value of Q is 7. From Fig. 1 we can see our algorithm has the least number of collision slots among the algorithms. Since the location of judge whether the current frame length is optimal or not is fixed, the number of collision slots in EAFL is more than that of the proposed algorithm. APAFQ inventories more collision slots due to its slot by slot adjustment strategy. Since there is no early adjustment strategy for DFSA, the number of collision slots to be recorded is the largest compared to the other three algorithms.

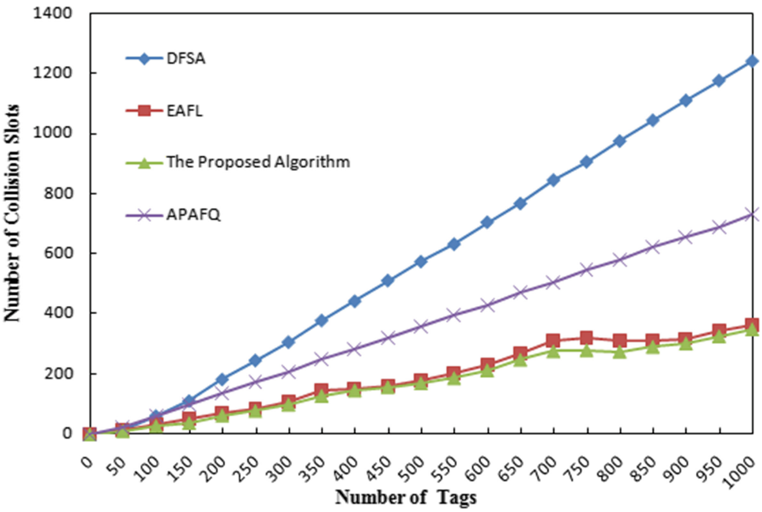

Fig. 1. Number of collision slots for various algorithms.

And we also compared the system throughput of the four algorithms with the same initial frame length 128 in Fig. 2. The system throughput is the ratio of the time spent by successful slots and the total time spent by reader to identify all tags. The total time spent in identifying tags includes the time taken by the successful time slot plus the time spent in the collision slot and the idle slot. The throughput formula is shown in (8).

$$T_{hr} = \frac{S \cdot T_S}{S \cdot T_S + C \cdot T_C + E \cdot T_E}. \tag{8}$$

where S is the number of successful slots, E is the number of empty slots, C is the number of collision slots, T_S, T_C and T_E are the durations of successful, collision, and empty slot, respectively.

As shown in Fig. 2, the system throughput of those four algorithms is rapidly reaching its maximum and then fluctuates near the maximum. Among them, the system throughput of the proposed algorithm is better than all other algorithms. It maintained a steady state near its maximum after the number of tags reaches 50. EAFL will take inventory of excess collision slots and free slots when the current frame length is not optimal, so its performance is lower than the proposed algorithm. APAFQ is slow to adjust the frame length due to the slot by slot strategy, so the system throughput is lower than that of EAFL and the proposed algorithm. The performance of DFSA fluctuates greatly when the number of tags is small, because the influence of initial frame length is larger.

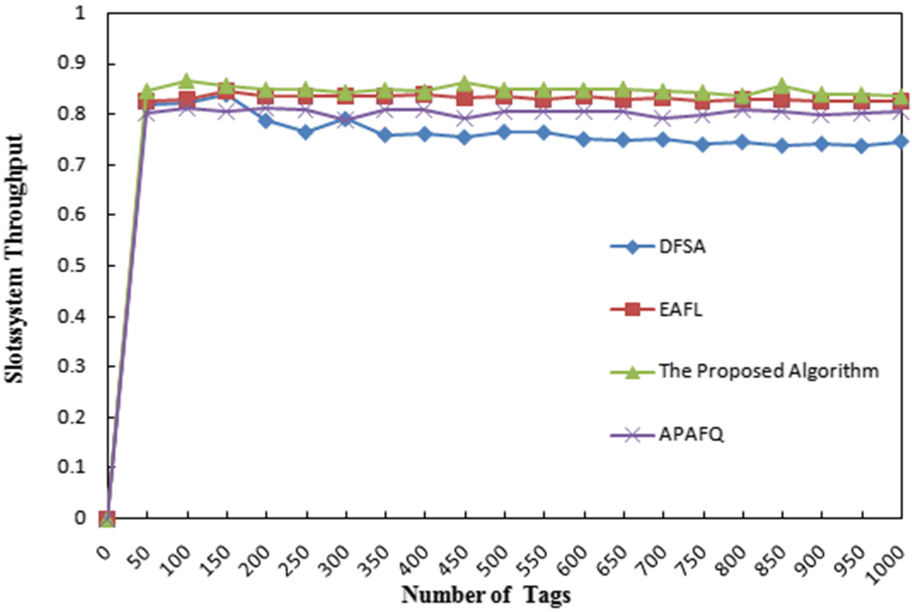

Fig. 2. System throughput for various algorithms

4 Conclusion

In order to adjust the detection point with the frame length dynamically, the paper presented an algorithm which uses the sample size to calculate the size of the detection slots area in the frame. Simulation results show the proposed algorithm has the advantages of less number of collision slots and higher system throughput.

Acknowledgements. This work was supported by The National Natural Science Foundation of China No. 61571150, Changzhou Sciences and Technology Program No. CJ20160016, General Financial grant from the China Postdoctoral Science Foundation No. 2016M601707.

References

1. EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz–960 MHz, version 1.2.0
2. Floerkemeier, C.: Transmission control scheme for fast RFID object identification. In: IEEE International Conference on Pervasive Computing and Communications Workshops, p. 457. IEEE Computer Society (2006)
3. Chen, Y., Su, J., Yi, W.: An efficient and easy-to-implement tag identification algorithm for UHF RFID systems. IEEE Commun. Lett. **PP**(99), 1 (2017)
4. Šolić, P., Radić, J., RožIć, N.: Energy efficient tag estimation method for ALOHA-based RFID systems. Sens. J. IEEE **14**(10), 3637–3647 (2014)
5. Su, J., Zhao, X., Hong, D.: Q-value fine-grained adjustment based RFID anti-collision algorithm. IEICE Trans. Commun. **E99. B**(7), 1593–1598 (2016)
6. Chen, W.T.: Optimal frame length analysis and an efficient anti-collision algorithm with early adjustment of frame length for RFID systems. IEEE Trans. Veh. Technol. **65**(5), 3342–3348 (2016)
7. Dhakal, S., Shin, S.: Parametric heuristic schemes for performance improvement in Frame Slotted Aloha based RFID systems. In: International Conference on ICT for Smart Society, pp. 1–5. IEEE (2013)
8. Klair, D.K., Chin, K.W., Raad, R.: A survey and tutorial of RFID anti-collision protocols. IEEE Commun. Surv. Tutor. **12**(3), 400–421 (2010)
9. Schoute, F.C.: Dynamic frame length ALOHA. Mob. Commun. **31**(4), 565–568 (1983)
10. Cha, J.R., Kim, J.H.: Novel anti-collision algorithms for fast object identification in RFID system. In: Proceedings of the International Conference on Parallel and Distributed Systems, pp. 63–67. IEEE (2005)
11. Han, G., Que, W., Jia, G., Shu, L.: An efficient virtual machine consolidation scheme for multimedia cloud computing. Sensors **16**(2), article no. 246 (2016)
12. Han, G., Chao, J., Zhang, C., Shu, L., Li, Q.: The impacts of mobility models on DV-hop based localization in mobile wireless sensor networks. J. Netw. Comput. Appl. **42**(6), 70–79 (2014)
13. Han, G., Jiang, J., Shu, L., Guizani, M., Nishio, S.: A two-step secure localization for wireless sensor networks. Comput. J. **56**(10), 1151–1153 (2013)