Enhanced Frame Break Mechanism for ALOHA-Based RFID Anti-Collision Algorithm

Xiangjian Zeng, Jiyun Qiu, Xijun Wang, and Xiang Chen(✉)

School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China
153600351@qq.com, 764636216@qq.com, wangxijun@mail.sysu.edu.cn, chenxiang@mail.sysu.edu.cn

Abstract. Radio frequency identification (RFID) technology is a kind of non-contact identification technology, which has been widely applied in logistics management, access control system and other fields. However, when multiple tags response the RFID reader simultaneously, the superposition of signals will deduce reader's incorrect reception, which is called collision. Dynamic frame slotted ALOHA (DFSA) algorithm is a effective anti-collision algorithm in RFID system, but its' high system efficiency relies on the reliability of the tags' number estimation. Some researches propose a frame break policy for ALOHA-based RFID system, which breaks the frame early when judging that the current frame length is not appropriate through slot-by-slot estimation of number of tags and is called SbS-DFSA. Here, we find some improvement space of SbS-DFSA, and make it more efficient by modifying some details. The simulation shows that our modification is efficient and improves the performance practically.

Keywords: Radio frequency identification technology · Anti-collision algorithm · Dynamic frame slotted ALOHA · Frame break

1 Introduction

Radio frequency identification (RFID) is a technology that uses radio frequency signal to automatically identify target objects and obtain relevant information.

This work was supported in part by the Guangdong Provincial Special Fund For Modern Agriculture Industry Technology Innovation Teams under Grant 2020KJ122, in part by the Collaborative Education Project of Industry University Cooperation (No.201802016050), in part by the State's Key Project of Research and Development Plan under Grants 2019YFE0196400, in part by the NSFC under Grant 61771495, and in part by the Guangdong R&D Project in Key Areas under Grant 2019B010158001.

B. Li et al. (Eds.): IoTaaS 2020, LNICST 346, pp. 222–233, 2021.
https://doi.org/10.1007/978-3-030-67514-1_18

As a key technology to build the Internet of things (IoT), RFID has been widely used in transportation, manufacturing, security and other fields. In recent years, More and more industries have used this technology to improve logistics efficiency and service level.

The implementation of RFID system consists of two hardware entities (reader and tag) and corresponding software. Reader is a active device, responsible for reading and writing the data stored in tag, and powering the passive tag. Passive tag needn't battery and gets energy through the wireless signal transferred by reader. With the advantage of small size and low cost, passive tag is used more widely than active tag, which need independent battery.

In a typical environment with one reader and multiple passive tags, the reader send wireless signal, multiple tags are energized and backscatter their modulation signal to the reader simultaneously, and then collision occurs, as the Fig. 1 shows. This means we have to design specific algorithm and mechanism to identify the each tags respectively.

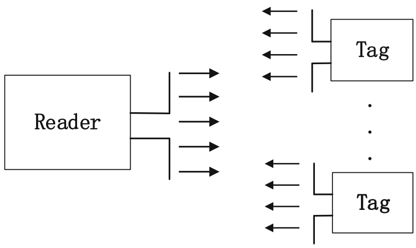

Fig. 1. Typical RFID system with passive tag

An-collision problem is not unique to RFID system and widely exists in the system which need to support access of multi-target through a common channel. In the field of RFID, time division scheme is relatively more appropriate because of its low-complexity in hardware, which is important for tag, especially passive tag.

The anti-collision algorithm in RFID system can be divided into two main branches: tree algorithm and ALOHA-based algorithm.

Tree algorithm essentially create a binary-tree structure to divide tags into different time slots for transmission, related algorithms include Binary Tree Splitting [1], Adaptive Binary Tree Spliting [2,3], Query Tree [4], Binary Search [5] and Bitwise Arbitration [7–9].

ALOHA-based algorithm is based on a scheme that when the collision occurs, tags that participate in the transmission will set a delay time, within which they can't transmission again. Based on this mechanism, Slotted ALOHA [10] and Static Frame Slotted Frame ALOHA [11] are proposed. Slotted ALOHA limits tags' transmission in a time slot with boundary and avoid the partially collision. Frame Slotted ALOHA specifies that a frame consists of multiple time slots and each tag can only transmission once in a frame. It can be proved mathematically that in Frame Slotted ALOHA, the maximum throughput is reached when the number of tag is equal to the frame length (the number of time slots in a frame) [12,13]. Static Frame Slotted ALOHA set a fixed frame length, while Dynamic Slotted ALOHA [5] estimates the number of tags in the end time slots of each frame, and set the next frame length to the estimated number of unidentified tags to achieve better performance.

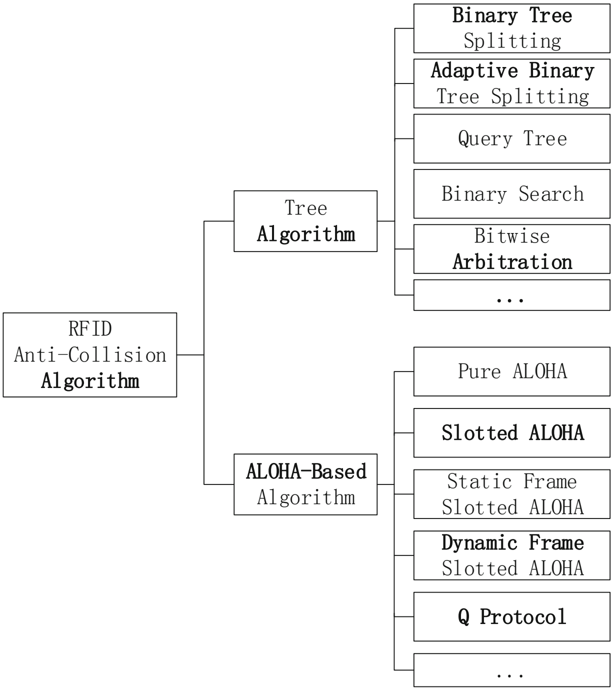

Fig. 2. RFID anti-collision algorithms

Further, some researchers propose slot-by-slot estimations of number of tags, which break the limit that estimation is only made in the end of a frame. This make it possible for reader to break the frame early when the frame length is far from the result of slot-by-slot estimation.

Based on these slot-by-slot estimation methods, a frame break policy for ALOHA-based algorithm [6] was proposed, which is called SbS-DFSA. However, In this paper, we find some irrationalities of SbS-DFSA. To fix the problem, we propose the correction and improvement scheme for SbS-DFSA and call it Enhanced-SbS-DFSA. Simulation results show that Enhanced-SbS-DFSA surpasses the performance of SbS-DFSA shown in [6].

This paper is organized as follows. In section two, we give a brief introduction of DFSA and SbS-DFSA. In section three we point out the defects of SbS-DFSA and propose our Enhanced-SbS-DFSA. In section four we show the simulation results and section five is conclusion part.

2 Related and Basic Work

2.1 DFSA Algorithm

In Frame Slotted ALOHA, the reader can divide the time slots into three types—empty time slot, successful time slot and collision time slot, with none, one and more than one tags response to the reader respectively. Here we use L to stand for the frame length, and use E, S, C to stand for the number of empty time slots, successful time slots and collision time slots in a frame respectively. Universally, the estimation algorithms in DFSA use L, C, S, E to estimate number of tags.

Vogt [12,13] is a high-accuracy estimation method in DFSA, which is based on the minimum mean square error criterion. The formula for Volt estimation is show in (1), where $a_t^{L,k}$ is the expected number of time slots that t tags response to reader when frame length is L and number of tags is k. Note that here we set the search range to 1 to 10L, which avoid the unreasonal estimation result in the case $C = L$ [14]. However, one disadvantage of Volt algorithm is that it needs to use some iterative or traversing method to solve this optimization problem, so that its calculation cost is very high, which induces high requirements to the hardware and software of reader.

$$Volt\_Estimation(L, E, S, C) = \arg \min_{k \in (1,10L)} \left\| \begin{pmatrix} a_0^{L,k} \\ a_1^{L,k} \\ a_{\geq 2}^{L,k} \end{pmatrix} - \begin{pmatrix} E \\ S \\ C \end{pmatrix} \right\|^2$$

$$a_t^{L,k} = L \binom{k}{t} \left(\frac{1}{L}\right)^t \left(1 - \frac{1}{L}\right)^{k-t}, a_{\geq 2}^{L,k} = L - a_0^{L,k} - a_1^{L,k} \tag{1}$$

Estimation method based on maximum likelihood criterion [15] is shown in (2), where $N(k, S)$ is number of ways to distribute k tags to S slots and $N(k - S, C)$ is number of ways to distribute k-S tags to C slots. Similarly, the calculation cost is very high too. To reduce the calculation cost at the price of decreasing estimation accuracy, it is simplified to formula (3), which is called improved linearized combinatorial model (ILCM) [16].

$$\begin{aligned} &Maximum\_Likelihood\_Estimation(L, E, S, C) \\ &= \arg \max_k P(E, S, C \mid k, L) \\ &= \arg \max_k \frac{N(k - S, C)N(k, S)}{L^k} \frac{L!}{E!S!C!} \end{aligned} \tag{2}$$

$$k = max\{\frac{C}{\left(\frac{L}{-2.282-0.273L}\right)C + (4.344L - 16.28)} + 0.2407\ln(42.56 + L), 0\}$$
$$l = (1.2592 + 1.513L)\tan(1.234L^{-0.9907}C)$$
$$ILCM\_Estimation(L, E, S, C) = kS + l \; when \; c \neq 0, \; S \; when \; c = 0 \tag{3}$$

It should be emphasized that frame length should be limited to integer power of 2. Therefore if the estimation number of identified tags is x, we should set the next frame length to f(x) and function f is defined in (4). Compared to the scheme without limit to frame length, this reduces hardware requirements for tag, which is important for passive tag. This limit has been adopted by EPC RFID G2 UHF protocol [18].

$$f(x) = 2^{round(log_2(max(1,x)))} \tag{4}$$

2.2 SbS-DFSA Algorithm

We define system efficiency as the number of tags divided by the number of time slots costed during the identified process, as showed in (5), which is the main index used to measure the performance of RFID anti-collision algorithm.

$$System\ efficiency = \frac{Number\ of\ tags}{Number\ of\ time\ slots\ costed} \tag{5}$$

The maximum system efficiency of DFSA is almost 0.368 [17], but it is hard to reach because the estimation error of number of tags. Especially, when the estimation result is far from the real value, the large estimation error can't be revised until the end of this frame, which seriously affect the system efficiency. Fortunately, several slot-by-slot estimation methods were proposed, such as SbS-Volt [19] whose formula is shown in (6) and SbS-ILCM [6] whose formula is shown in (7). Here, i is index of time slot, whose range is 1 to L. Definition of e,s,c are similarly to E,S,C, but they are only counted within time slot 1 to time slot i.

$$a_{t,i}^{L,k} = i\binom{k}{t}\left(\frac{1}{L}\right)^{t}\left(1-\frac{1}{L}\right)^{k-t}, a_{\geq 2,i}^{L,k} = L - a_{0,i}^{L,k} - a_{1,i}^{L,k}$$
$$SbS\_Volt\_Estimation(L, i, E, S, C) = \arg\min_{k\in(1,10L)} \left\| \begin{pmatrix} a_{0,i}^{L,k} \\ a_{1,i}^{L,k} \\ a_{\geq 2,i}^{L,k} \end{pmatrix} - \begin{pmatrix} E \\ S \\ C \end{pmatrix} \right\|^2 \tag{6}$$

$$k'(i) = max\{\frac{c}{\left(\frac{i}{-2.282-0.273i}\right)c + (4.344i - 16.28)} + 0.2407\ln(42.56 + i), 0\}$$
$$l'(i) = (1.2592 + 1.513i)\tan(1.234i^{-0.9907}c)$$
$$SbS\_ILCM\_Estimation(L, i, e, s, c) = k'(i)s + l'(i) \; when \; c \neq 0, \frac{SL}{i} when \; c = 0 \tag{7}$$

Based on these slot-by-slot estimation methods, a mechanism that the frame is stopped early when slot-by-slot estimation value is far from the frame length is proposed [6], which is called SbS-DFSA. The complete pseudocode of SbS-DFSA using SbS-ILCM estimation is shown in algorithm 1. SbS-DFSA can also be combined with other slot-by-slot estimation methods by modifying row 2 of algorithm 1. For example, if we change algorithm 1 row 2 to '$R(i) = max(1, SbS\_Volt\_Estimation(L, i, e, s, c))$', this algorithm will become SbS-DFSA using SbS-Volt estimation. Note that in SbS-DFSA using SbS-Volt estimation, $SbS\_Volt\_Estimation(L, i, e, s, c) = \frac{2.39cL}{i}$ *when* $c = i$, which we doesn't adopt in Enhanced-SbS-DFSA using SbS-Volt estimation.

Algorithm 1: SbS-DFSA with SbS-ILCM

1: **while** Reader is working **do**
2: $\quad R(i) = max(1, SbS\_ILCM\_Estimation(L, i, e, s, c))$
3: $\quad R_Q(i) = f(R(i))$
4: $\quad$ **if** $(i > 1)$ *and* $(|R(i) - R(i-1)| \leq 1)$ **then**
5: $\qquad P_1 = \frac{R(i)}{L}(1 - \frac{1}{L})^{R(i)-1}$
6: $\qquad P_2 = \frac{R(i)}{R_Q(i)}(1 - \frac{1}{R_Q(i)})^{R(i)-1}$
7: $\qquad$ **if** $(LP_1 - s) < (R_Q(i)P_2)$ **then**
8: $\qquad\quad R_Q(i) = f(R(i))$
9: $\qquad\quad i = 1$ *and start a new frame with* $L = RQ(i)$
10: $\qquad$ **else**
11: $\qquad\quad i = i + 1$
12: $\qquad$ **end if**
13: $\quad$ **else**
14: $\qquad i = i + 1$
15: $\quad$ **end if**
16: $\quad$ **if** $i == (L + 1)$ **then**
17: $\qquad i = 1$ *and start a new frame with old* L
18: $\quad$ **end if**
19: $\quad$ *Send message to tags*
20: $\quad$ *Update* e, s, c
21: **end while**

3 Enhanced-SbS-DFSA Algorithm

3.1 Defects of SbS-DFSA

Although paper [6] claimed that SbS-DFSA shows better performance than traditional DFSA accordingly to simulation results, it is easy for us to find following unreasonable places in SbS-DFSA.

First, the judging rule of convergence is unreasonable, which we can see in algorithm 1 row 12. Specifically, the judging rule of convergence in SbS-DFSA is that the difference between estimation value of this time slot and that of last time slot is less than or equal to one. Typical convergence cases of slot-by-slot estimation are shown in Fig. 3 and obviously the judging rule of convergence in SbS-DFSA can't recognize them easily.

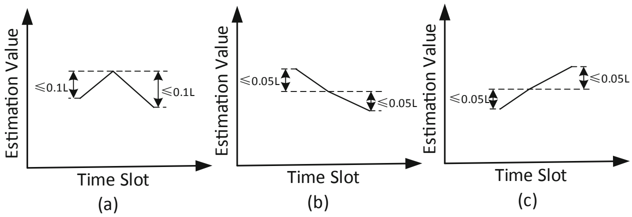

Fig. 3. Typical convergence cases of slot-by-slot estimation

Second, the judging rule of frame break at convergence is unreasonable, which we can see in algorithm 1 row 15, where P_1 is the probability that a time slot is a successful time slot in this frame and P_2 is the probability that a time slot is a successful time slot in next frame whose length is denoted by slot-by-slot estimation, under the assumption that the estimation value is equal to the real value. Therefore, $LP_1 - s$ is the expected number of tags identified successfully in remaining time slots of this frame and $R_Q(i)P_2$ is the expected number of tags identified successfully in next frame. Obviously, the judging rule of frame break that $LP_1 - s$ is lower than $R_Q(i)P_2$ is not intuitively proper.

Third, when number of time slots is small, information reader gets is not enough to give a accurate estimation, which may induce a unreasonal frame break in SbS-DFSA.

Forth, when reader decide to break the frame, if new frame length is equal to the old one, this frame break is meaningless.

Fifth, when the estimation value doesn't converge during the whole frame, SbS-DFSA set next frame length equal to the prior one, which may maintain a improper frame length for many frames.

3.2 Enhanced-SbS DFSA Algorithm

To overcome the deficiencies of SbS-DFSA we mentioned before, we propose several improvements for it and call the new method Enhanced-SbS-DFSA. The complete pseudocode of Enhanced-SbS-DFSA using SbS-ILCM estimation is shown in algorithm 2. Enhanced-SbS-DFSA can also be combined with other slot-by-slot estimation methods by modifying row 3, row 3 and row 28 of algorithm 1. For example, if change row 3, row 7 and row 28 of algorithm 1 to '$R(i) = max(1, SbS\_Volt\_Estimation(L, i, e, s, c) - s)$', this algorithm will become Enhanced-SbS-DFSA using SbS-Volt estimation.

First, change the judging rule of convergence. $R(i-2)$, $R(i-1)$ and $R(i)$ represent the slot-by-slot estimation value of time slot i-2, i-1 and i respectively, while $R_Q(i)$ stands for $f(R(i))$. To deal with the situation like cases shown in Fig. 3, we design a judging rule of convergence shown in algorithm 2 row 9 to row 15. Compared to SbS-DFSA, it uses information of three time slots (SbS-DFSA only uses information of two time slots), sets the convergence threshold dynamically according to the frame length, and propose a scheme to recognize the slight fluctuation of slot-by-slot estimation value shown in (a) of Fig. 3.

Second, change judging rule of frame break at convergence. For Enhanced-SbS-DFSA combined with SbS-ILCM, when estimation value converges, the reader stops the frame early if LP_1 is lower than $R_Q(i)P_2$, which is an empirical conclusion and can be seen in algorithm 2 row 19.

Third, set a start point for slot-by-slot estimation. The start point can be set by setting the value of M in algorithm 2 row 6. For Enhanced-SbS-DFSA using SbS-ILCM estimation M takes 3, while for Enhanced-SbS-DFSA using SbS-Volt estimation M takes 8.

Forth, Enhanced-SbS-DFSA won't break a frame with a new frame length that is equal to the old one, while SbS-DFSA will. This mechanism is integrated into the judging rule of convergence of Enhanced-SbS-DFSA.

Fifth, when the estimation value doesn't converge during the whole frame, Enhanced-SbS-DFSA will set next frame length to the last estimation value. Compared to SbS-DFSA which doesn't change frame length in such a situation, Enhanced-SbS-DFSA ensures that improper frame length won't be maintained for many frames when estimation value doesn't converge. This mechanism can be seen in algorithm 2 row 3 to row 5.

Algorithm 2: Enhanced-SbS-DFSA with SbS-ILCM

```
while Reader is working do
  if i == L then
    R(i) = max(1, SbS_ILCM_Estimation(L, i, e, s, c))
    R_Q(i) = f(R(i))
    i = 1 and start a new frame (L = R_Q(i))
  else if i > M then
    R(i) = SbS_ILCM_Estimation(L, i, e, s, c)
    R_Q(i) = f(R(i))
    if (R_Q(i) ≠ L) and (|R(i) − R(i − 1)| ≤
    max(1, 0.1L)) and (|R(i − 1) − R(i − 2)| ≤
    max(1, 0.1L)) and (((R(i) ≥ R(i − 1)) and (R(i − 1) ≤
    R(i − 2))) or ((R(i) ≤ R(i − 1)) and (R(i − 1) ≥ R(i − 2)))) then
      FLAG = 1
    else if (R_Q(i) ≠ L) and (|R(i) − R(i − 1)| ≤
    max(1, 0.05L)) and (|R(i) − R(i − 1)| ≤ max(1, 0.05L)) then
      FLAG = 2
    else
      FLAG = 0;
    end if
    if (FLAG == 1) or (FLAG == 2) then
      P_1 = (R(i)/L)(1 − 1/L)^(R(i)−1)
      P_2 = (R(i)/R_Q(i))(1 − 1/R_Q(i))^(R(i)−1)
      if LP_1 < RQ(i)P_2 then
        i = 1 and start a new frame L = R_Q(i − 1)
      else
        i = i + 1
      end if
    else
      i = i + 1
    end if
  else
    R(i) = SbS_ILCM_Estimation(L, i, e, s, c)
    i = i + 1
  end if
  Send message to tags
  Update e, s, c
end while
```

4 Simulation Results

The relationship curve of system efficiency versus number of tags is influenced by initial frame length. Specifically, the peak of system efficiency will appear at the place where the number of tags is equal to the initial frame length. We set initial frame length to 2^4 and get the simulation results of DFSA with ILCM estimation, DFSA with Volt estimation, SbS-DFSA with SbS-ILCM estimation, SbS-DFSA with SbS-Volt estimation, Enhanced-SbS-DFSA with SbS-ILCM estimation, Enhanced-SbS-DFSA with SbS-Volt estimation and DFSA with ideal estimation. Here, we use function 'fminbnd' of MATLAB R2014a to solve the optimization problems in Volt and SbS-Volt estimation. DFSA with ideal estimation assumes that estimation value is equal to real value, which is impossible in realistic. Note that owing to the limit to frame length to integer power of 2, the system efficiency can not reach 0.368 even if we assume the estimation is deal, as Fig. 4 shows.

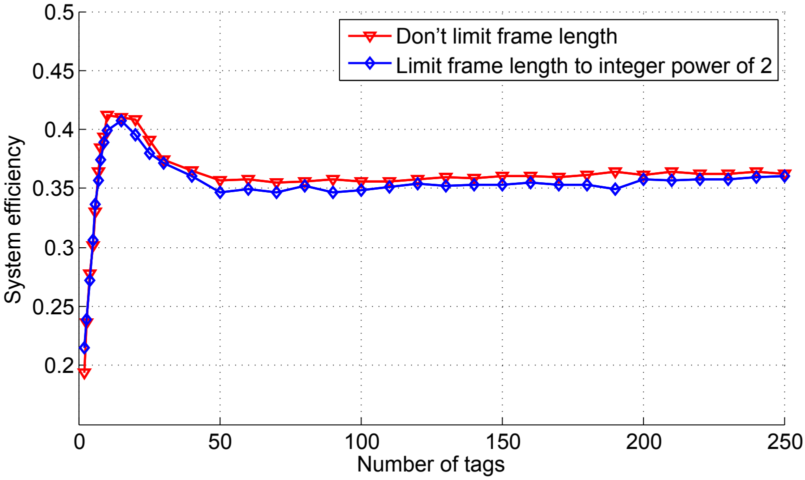

Fig. 4. Performance of DFSA with ideal estimation

The simulation results with the initial frame of 2^4 is shown in Fig. 5. In order to make the performances of different algorithms more differentiated, here we give relationship curve of number of time slots costed versus number of tags, which can be equivalent to relationship curve of system efficiency versus number of tags. From the simulation results we can know Enhanced-SbS-DFSA shows better performance in system efficiency than DFSA, and shows higher system efficiency than SbS-DFSA whether the estimation algorithm used is SbS-Volt or SbS-ILCM. Under the limit that frame length must be integer power of two, Enhanced-SbS-DFSA almost achieves the performance of DFSA with ideal estimation in system efficiency.

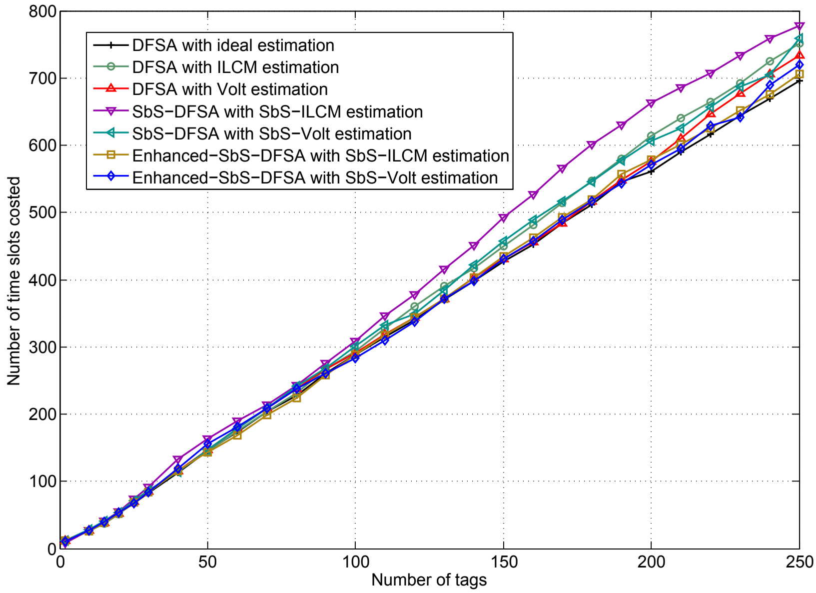

Fig. 5. Simulation results when initial frame length is 16

5 Conclusions

In this paper, focus on the field of RFID anti-collision algorithm, we find several defects of SbS-DFSA and propose some improvements in the aspects of judging rule of convergence, judging rule of frame break at convergence and so on. Simulation results show that our algorithm achieve higher system efficiency than DFSA and SbS-DFSA, no matter we use Volt, SbS-Volt, ILCM, SbS-ILCM as our estimation method. Under the limit to frame length adopted by EPC RFID G2 UHF protocol, in aspect of system efficiency our algorithm almost achieves the performance of DFSA with ideal estimation, which is highly meaningful and proves its application value.

References

1. Hush, D.R., Wood, C.: Analysis of tree algorithms for RFID arbitration. In: The IEEE International Symposium on Information Theory, Mexico City, USA, p. 107 (1998)
2. Myung, J., Lee, W.: Adaptive binary splitting: a RFID tag collision arbitration protocol for tag identification. In: IEEE BROADNETs, Boston, MA, USA, pp. 347–355 (2005)
3. Myung, J., Lee, W., Srivastava, J.: Adaptive binary splitting for efficient RFID tag anti-collision. IEEE Pers. Commun. **10**(3), 144–146 (2006)

4. Law, C., Lee, K., Siu, K.-Y.: Efficient memoryless protocol for tag identification (extended abstract). In: Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Toronto, CA, pp. 75–84 (2000)
5. Finkenzeller, K.: RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification, 2nd edn. John Wiley and Sons Ltd., Hoboken (2003)
6. Solic, P., Radic, J., Rozic, N.: Early frame break policy for ALOHA-based RFID systems. IEEE Trans. Autom. Sci. Eng. **13**(2), 1–6 (2015)
7. Bo, F., Jin-Tao, L., Jun-Bo, G., Zhen-Hua, D.: ID-binary tree stack anticollision algorithm for RFID. In: Proceedings of 11th IEEE Symposium on Computers and Communications, Sardinia, Italy, pp. 207–212 (2006)
8. Jacomet, M., Ehrsam, A., Gehrig, U.: Contact-less identification device with anti-collision algorithm. In: Conference on Circuits Systems, Computers and Communications, Athens, Greece (1999)
9. Kim, S.H., Shin, M.K., Park P.: A new tree-based tag anticollision protocol for RFID systems. In: International Conference on Communications in Computing, Nevada, pp. 83–86 (2006)
10. Schwartz, M.: Telecommunication Networks Protocols, Modeling and Analysis. Addison-Wesley, Boston (1988)
11. Zhen, B., Kobayashi, M., Shimizu, M.: Framed ALOHA for multiple RFID objects identification. IEICE-Trans. Commun. **E88–B**, 991–999 (2005)
12. Vogt, H.: Multiple object identification with passive RFID tags. In: The IEEE International Conference on Systems, Man and Cybernetics, Tunisia, pp. 6–13 (2002)
13. Vogt, H.: Efficient object identification with passive RFID tags. In: IEEE PerCom, TX, USA (2002)
14. Knerr, B., Holzer, M., Angerer, C., Rupp, M.: Slot-wise maximum likelihood estimation of the tag population size in FSA protocols. IEEE Trans. Commun. **58**(2), 578–585 (2010)
15. Šolić, P., Radić, J., Rožić, N.: Algorithm for deriving optimal frame size in passive RFID UHF class1-gen2 standard using combinatorial model boundaries. AUTOMATIKA **51**, 255–263 (2010)
16. Solic, P., et al.: Improved linearized combinatorial model (ILCM) for optimal frame size selection in ALOHA-based RFID systems radio. In: Proceedings of the IEEE International Conference on Communications Workshops (ICC), Budapest, Hungary, pp. 1092–1097 (2013)
17. Chen, W.-T.: An accurate tag estimate method for improving the performance of an RFID anticollision algorithm based on dynamic frame length ALOHA. IEEE Trans. Autom. Sci. Eng. **6**(1), 9–15 (2009)
18. EPCglobalInc.: EPC Radio-Frequency Identity Protocols Generation-2 UHF RFID. EPCglobal (2015)
19. Knerr, B., Holzer, M., Angerer, C., Rupp, M.: Slot-wise maximum likelihood estimation of the tag population size in FSA protocols. IEEE Trans. Commun. **58**(2), 578–585 (2010)