An Improved EPC Gen-2 Slot random Anti-collision Algorithm Based on Active Strategy

Jinghe Tian[1,2] Yushun Fan[3] Yunlong Zhu[1] Ruizi Zhang[1,2]
1. Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China
2. Graduate School of the Chinese Academy of Sciences, Beijing, 100039, China
3. Dept. of Automation, Tsinghua University, Beijing, 100084, China
E-mail: tianjinghe@sia.ac.cn

***Abstract*— The ability of simultaneously identifying many tags is crucial for many RFID applications. EPC Gen-2 protocol specified Slotted Random (SR) algorithm to solve tag collision problem. In this paper, an Active Slotted Random (ASR) algorithm which is an improved SR algorithm was proposed. Two slot random-redistribute commands are introduced to make the collision tags redistribute in the adjacent and increase the response slot of other tags, which will lead to relative average slot distribution of tags. Simulation results shows that the proposed ASR algorithm observably improve the tag throughout by 30 percent compared with SR algorithm. It is without question that ASR will contribute to improve the performance of EPC Gen-2 RFID system because the reader can identify more tags with shorter time.**

***Keywords*—RFID, EPC Protocol, Anti-collision, Slot**

I. INTRODUCTION

RADIO frequency identification (RFID) has been an important and ubiquitous infrastructure technology in supply chain management, toll-payment, libraries, e-passports, shopping and many other areas [1]. RFID system generally consists of a reader and many tags. A reader in RFID system broadcasts the request message to the tags. Upon receiving the message, all tags send the response back to the reader. If only one tag responds, the reader receives just one response. But if there is more than one tag response, their responses will collide on the RF communication channel, and thus cannot be received by the reader. This problem is referred to as the "Tag-collision" [1]. An effective system must avoid this collision by using anti-collision algorithm because the ability to identify many tags simultaneously is crucial for many applications [1-4].

Researchers have been addressing tag-collision problem in various ways; some methods seem to increase data transmission speed by extending the frequency bandwidth to increase tag identification efficiency via minimizing tag collisions. This is not a very satisfactory solution as the frequency band will always be limited. The most widely used techniques are the framed slotted ALOHA algorithm and binary search algorithm [5]. Since it is simple implementation, the framed slotted ALOHA algorithm [6-9] is the most frequently used.

EPC Gen-2 protocol adopts Slotted Random (SR) algorithm which is dynamic framed slotted ALOHA algorithm (DFSA) in the nature. In SR algorithm, as DFSA, Reader can adjust the number of slots in a frame, and tags can randomly choose one slot in a frame to response. Adjusting slot number at any time according to tag distribution, so SR can adjust frame size more flexibly, and its performance is better than other ALOHA algorithms. However, SR algorithm adopts ignoring strategy when tag collision occurs, i.e. ignoring collision tags and handling the tags located in the next slot. Consequently, in this paper, to improve the performance of anti-collision algorithms specified in EPC Gen-2 protocol [10], we proposed an Active SR algorithm (ASR) for EPC Gen-2 tags.

The remainder of this paper is organized as follows. Section 2 describes the SR algorithm specified in EPC Gen-2 protocol. The Active SR algorithm is detailed in section 3. Simulation settings and results are given in section 4. Conclusions and our future work are drawn in section 5.

II. DESCRIPTION OF SR ALGORITHM

A. Basic Commands

Every EPC Gen-2 tag has a random number generator and a 15-bit slot counter. In SR algorithm, when received slot information, tag will random choose a slot from the slot range defined by reader into slot counter. If there exist tag collision or no tag respond, reader will send some command to adjust the response slot of tag. There are mainly three commands: Query, QueryAdjust and QueryRep. Query command provides tag with initial Q value by which tag can randomly choose a slot from $[0, 2^Q - 1]$ as the response slot. QueryAdjust command runs Q+d operations, where the value span of d is {+1, -1, 0}. After received QueryAdjust command, all unread tags will choose a new response slot and begin a new frame. The function of QueryRep command is reducing the value of tag slot counter by 1.

B. Flow of SR Algorithm

The flow chart of SR algorithm is showed in Figure 1. Reader firstly broadcast Query command that includes a parameter Q (an integer in [0, 15]) to all tags. Tags will generate a random integer range from 0 and $2^Q - 1$, and load the random integer into slot counter. Only the tag whose slot counter is 0 will respond to reader's reading or writing operation by sending a RN16 (16-bit random data). There may be three cases:

Case1: only one tag respond to reader and send a RN16 to reader, then reader successfully identify the tag and send QueryRep command to identify residual tags.

Case 2: no tag sends RN16 to reader, then reader can broadcast QueryRep command to reduce slot counter of all tags by 1, or reader broadcast QueryAdjust command to make all tags choose a new value of their slot counter.

Case 3: more than one tag send RN16 to reader, then tag collision occurs. Reader can broadcast QueryRep command to reduce slot counter of all tags by 1. Reader can also increase Q value and broadcast QueryAdjust command with the new Q to make all tags randomly choose response slot in a bigger range, which would reduce the probabilities of tag collision. By this way, Reader adjust response slot of every tag until all tags in interrogation zone would have been identified.

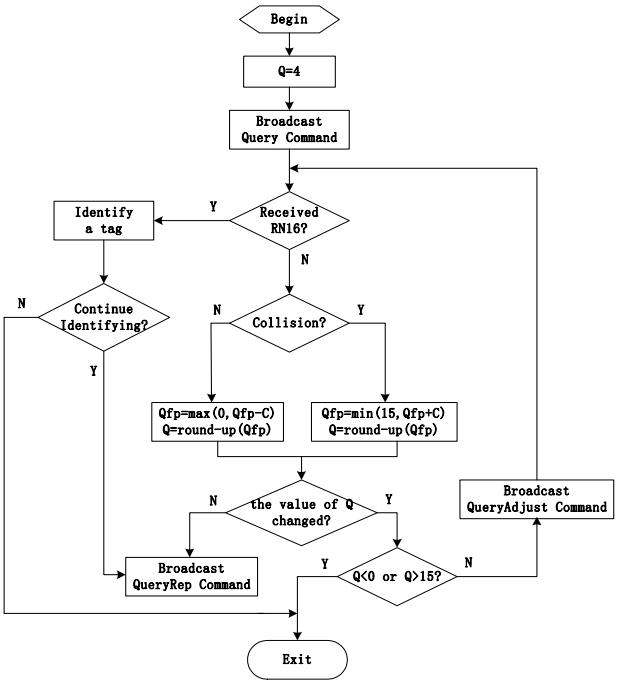

Figure 1. Flow chart of SR algorithm

C. Adjusting Parameter Q

For dynamic framed slotted ALOHA algorithm (DFSA), best throughout of system can be got when the slot number of frame is close to the number of tags [11]. Consequently, as concerning SR algorithm, we can get the following results: bigger Q will result in bigger range in which tag randomly choose response slot, and the probability of collision is lower. Otherwise, smaller Q will result in higher probability of collision. So reader should continuously change Q to ensure that the slot number be close to the tag number, which will bring high efficiency of identification. EPC Gen-2 protocol presents a Q-evolution algorithm as shown in Figure 2.

Where Qfp is the floating-point literal of Q, reader set Q of Query command by the round-up number of Qfp. The span of C is (0.1, 0.5). EPC Gen-2 protocol advises that C be set relative smaller value when Q value is bigger.

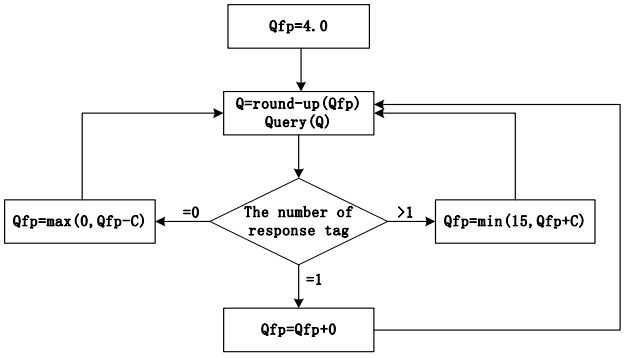

Figure 2. Evolution of parameter Q

III. ACTIVE SR ALGORITHM

A. Slot Random-Redistribute Command

When tag collision occurs, SR algorithm will skip over collided tags and continue handling tags in the latter slots, the collided tags can't have chance of being identified in the current frame. Consequently, the strategy that SR copes with tag collision is passive and inefficient. So we adopt active strategy to improve SR algorithm. The improved algorithm is referred to as ASR. And ASR introduces two slot random-redistribute commands: slot0 and slot1. The function of such two commands is as follows.

Slot0 command: After received broadcast command, the tags whose slot is 0 random select a slot number from {0, 1}, the selected slot number then would be loaded in slot counter. As for the tags whose slot is nonzero, they add their slot counter by 1.

Slot1 command: After received broadcast command, the tags whose slot is 1 random select a slot number from {0, 1}, the selected slot number then would be loaded in slot counter. As for the tags whose values of slot number are not 1, they add their slot counter by 1.

B. Flow of ASR Algorithm

The flow chart of ASR algorithm is showed in Figure 3. Firstly, the reader identifies one tag by using SR algorithm. During identifying the first tag, SR algorithm continually change the value of Q to adjust the slot number of frame to make it be possibly close to the number of tags. If tag collision occurs, reader will increase parameter Q and the slot number of frame; if there are no tag responses, reader will reduce parameter Q. In SR algorithm, the initial Q is 4. After identify the first tag using SR algorithm, a rational value of Q will be attained, i.e. the value of 2^Q is close to the number of tags.

After identifying the first tag by SR algorithm, the ASR algorithm will identify residual tags as follows:

(1) Reader broadcasts QueryRep command to the tags to be identified.

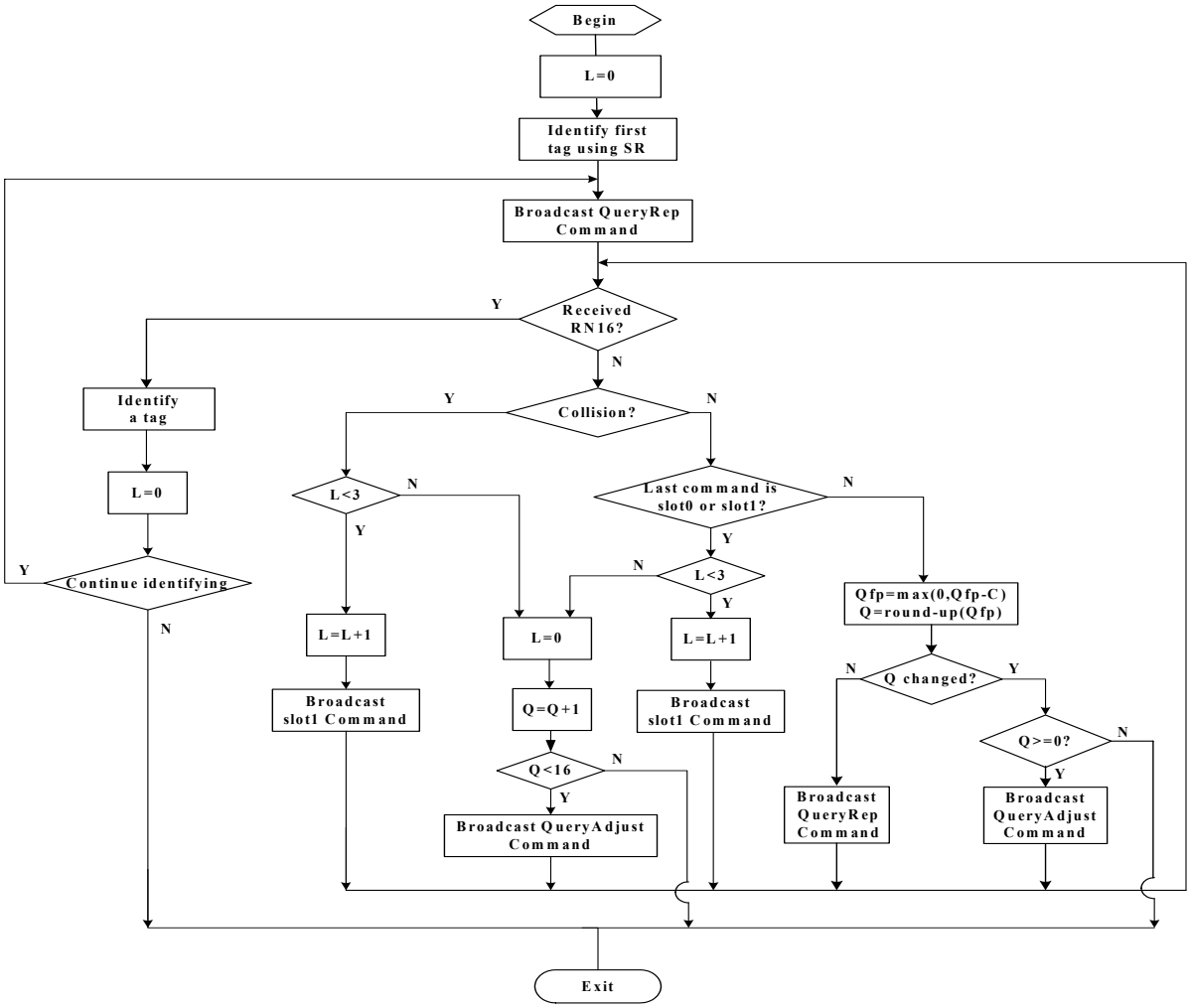

Figure 3. Flow chart of ASR algorithm

(2) Reader waits for responses from tags and cope with different cases. There should be three possible cases:

Case 1: Only one tag respond to reader, then reader has identified the tag, and reader goes to step (5).

Case 2: No tag responds to reader, then reader reduce Q by C ($0.1<C<0.5$) and round-up the new Q. Reader broadcasts QueryAdjust or QueryRep command based on judging whether the round-up of new Q is equal to Q. If $Q \geq 0$, then reader repeats step (2); Otherwise, it is indicated that no tag exists in the interrogation zone of the reader, then ASR algorithm exits identifying tags.

Case3: More than one tag respond to reader, i.e. tag collision occurs. Then Reader broadcast Slot0 command, then reader goes to step (3). After received Slot0 command, tags whose slot is 0 would randomly choose a new slot from {0, 1}, and other tags add their slot by 1. Such operation of tags will reduce the probability that slots of multiple tags simultaneously are 1, and the possibility of tag collision would be reduced.

(3) Reader waits for tag responses after broadcasting Slot0 or Slot1 command, and there should be three cases:

Case ⅰ: No tag responds to reader. If slot adjusting times reach cycle upper limit L (L is set 3), then reader goes to step (4); Otherwise, reader broadcasts Slot1 command and repeats step (3).

Case ⅱ: More than one tag responds to reader, here tag collision occurs. If slot adjusting times reach cycle upper limit L (L is set 3), then reader goes to step (4); otherwise, reader broadcasts Slot0 command and repeat step (3);

Case ⅲ: Only one tag respond and send a random number RN16 to the reader, which indicates that a tag has been identified successfully. Then reader goes to step (5).

(4) No tag were identified when slot adjusting times reach cycle upper limit L, which indicates that the value of Q is so small that excessive tags choose the same slots. So in this step, reader would add Q by 1 to increase slots of a frame and broadcast QueryAdjust command to tags, and then go to step (2). Tag will randomly choose a new slot number and load it to slot counter after received QueryAdjust command.

(5) If reader continues identifying next tag, then go to step (1); otherwise exit identifying process tags.

IV. SIMULATION RESULTS

In this paper, we do simulations for SR algorithm and its improved algorithm by MATLAB, the tag number is 10, 20, … , 200. Figure4 shows the curve that the average communication times vary with different number of tags. T is the communication times when all tags have been identified. In the Figure 4, it is indicated that the improved algorithm has less communication times than SR algorithm for the same number of tags.

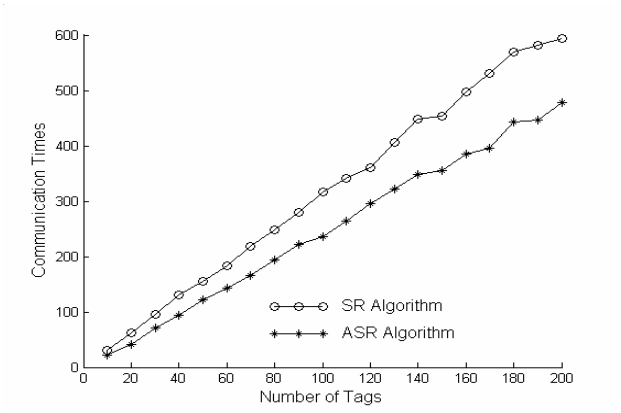

Figure 4. Communication times with different algorithms

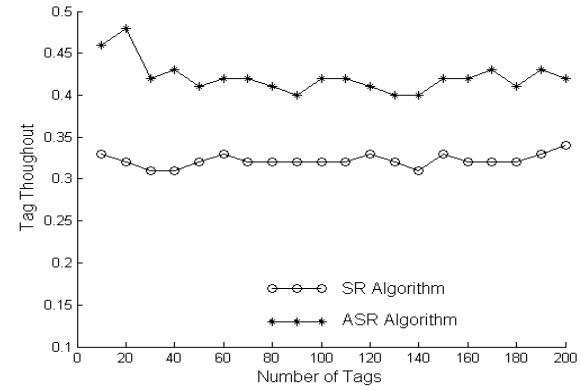

Figure 5 Throughout with different algorithms

Figure 5 shows the curve that the throughout of SR algorithm and its improved algorithm vary with different tag number. From the simulation results, two algorithms both have stable throughout. The average throughout of SR algorithm for different number of tags is 0.32. The corresponding average throughout of ASR algorithm is 0.42. Compared with SR algorithm, the proposed ASR algorithm observably improves the tag throughout by 30 percent.

As can be seen, the proposed ASR algorithm is better than SR algorithm at communication times and throughout. Two algorithms adopt different measures to tag collision. SR algorithm adopts passive strategy when tag collision occurs, i.e. ignoring collision tags in the current slot and handling the tags located in the next slot. The ASR algorithm adopts more active measure to handle tag collision. The proposed algorithm make the collision tags redistribute in the adjacent slots, and increase the response slot of other tags, which make tags averagely distribute in all slots of a frame. Then good performance of the ASR anti-collision algorithm will be got.

V. CONCLUSIONS

This paper has proposed an active SR anti-collision algorithm for EPC Gen-2 tags. The proposed algorithm is referred to as ASR. The design, implementation of ASR algorithm was discussed in detail. Simulation of ASR and its comparisons with SR algorithm were done by MATLAB. Simulation results show that, compared with SR algorithm, ASR improve the tag throughout by 30 percent and be more efficient than SR algorithm specified in EPC Gen-2 protocol.

If the proposed ASR algorithm is used in EPC Gen-2 RFID system, it will contribute to improve the performance of RFID system because the reader can identify more tags with shorter time.

ACKNOWLEDGMENT

This work was supported by the National High-Tech R&D Program of China, under Grant No.2006AA04A124.

REFERENCES

[1] K. Finkenzeller, RFID Handbook: Fundamentals and applications in Contact-less Smart Cards and Identification, 2nd Edition, John Wiley and Sons Ltd, pp. 195-219, 2003.

[2] S. Sarma, J. Waldrop, and D. Engels, Colorwave : An Anti-collision Algorithm for the Reader Collision Problem. IEEE International Conference on Communications, ICC ’03, vol.2, pp. 1206-1210. May, 2003.

[3] S. Sarma, D. Brock, and D.Engels, Radio frequency identification and electronic product code, IEEE MICRO, 2001.

[4] H. S. Choi, J. R. Cha and J. H. Kim, Fast Wireless Anti-collision Algorithm in Ubiquitous ID System. Proceedings of IEEE VTC 2004, L.A., USA. pp. 26-29, 2004.

[5] Tao Chen, Li Jin. Analysis and Simulation of RFID Anti-collision Algorithms. The 9th International Conference on Advanced Communication Technology, ICACT2007, pp. 697-701, 2007.

[6] Jae-Ryong Cha, Jae-Hyun Kim. Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID system. Proceedings of 2006 IEEE Consumer Communications and Networking Conference, volume 2, pp. 768-772, 2006.

[7] Peng, Qingsong; Zhang, Ming; Wu, Weimin;Variant Enhanced Dynamic Frame Slotted ALOHA Algorithm for Fast Object Identification in RFID system. 2007 IEEE International workshop on Anti-counterfeiting, Security, Identification, pp. 88-91, 2007.

[8] Leian Liu, Shengli Lai. ALOHA-Based Anti-Collision Algorithms Used in RFID System. Proceedings of 2006 International Conference on Wireless Communications, Networking and Mobile Computing., pp. 1-4. 2006.

[9] Su-Ryun Lee, Sung-Don Joo and Chae-Woo Lee. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification. Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services. pp. 166 – 172. 2005.

[10] EPCglobal.EPC™ Radio-Frequency Identity Protocols Class-1 Genaration-2 UHF RFID Protocol for Communications at 860 MHz—960MHz Version 1.1.0 Draft1. 2005.

[11] Wang, Jianwei; Wang, Dong; Zhao, Yuping;A novel anti-collision algorithm with dynamic tag number estimation for RFID systems. 2006 International Conference on Communication Technology, ICCT’06. pp. 1 – 4. Nov. 2006.