A Method to Improve Identification Performance of UHF RFID Systems

Ya-ning Yan[1] and Jian Xiong[2]

1. ZTE Telecommunications College, Xi'an Peihua University, Xi'an 710065, China
2. School of Control Science and Engineering, Dalian Univeristy of Technology, Dalian 116024, China
394811297@qq.com, sunli256@126.com

***Abstract* — Dynamic frame slotted ALOHA (DFSA) has been widely adopted to reduce tag collisions in Ultra High Frequency (UHF) radio frequency identification (RFID) system. In the existing DFSA algorithms, the reader needs to accurately estimate tag backlog and set a new frame length which is equal to the backlog to close to the theoretical maximum throughput 36.8% of the framed ALOHA. To overcome throughput limitation and improve the identification efficiency, we propose an efficient slot-hopped algorithm based on ALOHA protocol. The slot-hopped mechanism was used in each frame to skip collision and idle slots. Accordingly, the scheme significantly reduces collision and idle slots. The performance analysis and simulation results show that the proposed algorithm outperforms other ALOHA-based protocols**
.

Keywords - RFID, anti-collision, ALOHA, slot-hopped.

I. INTRODUCTION

Recently, there has been an increasing demand in the development of communications systems for the automatic identification of objects [1]. Radio frequency identification technology makes it possible and has attracted extensive attention. One of the major challenges of RFID system is the tag collision problem resulted from sharing the common wireless channel by all devices in the system [2]. Tag collision occurs when multiple tags simultaneously respond to the reader with their signals. This collision degrades the RFID system's identification performance. Hence, the primary goal of anti-collision protocols is to minimize the total identification time, i.e., the time required to identify all tags, or equivalently to maximize the system throughput.

Tag anti-collision protocols have been proposed to improve the identification efficiency in many applications mainly can be divided into two categories ALOHA-based [3-5] and tree-based protocols [6-8]. All these tree-based protocols based on the collision bit identification and tracking techniques. In the High or Low Frequency (HF/LF) RFID systems, the asynchronization is not present; the bit-tracking technique is easy to achieve. However, it is very difficult to achieve in Ultra High Frequency (UHF) RFID system, including EPCglobal C1 Gen2 [9] and ISO 18000-6B systems. The reason is that as defined in EPCglobal C1 Gen2 standard, modulation signals feature a different symbol rate, which deviates up to ±22% between the single tags in a UHF RFID system [10]. And, the arrival times of tag responses also vary in a range as large as 24 microseconds (μs). That means that, we could not detect which bits are collided with others in the UHF RFID systems. Compared with the tree-based algorithms, the ALOHA-based anti-collision algorithms are more suitable for UHF RFID systems, because it does not identify a specific collision position. It is easy to implement in a real RFID system.

The DFSA is a popular version of ALOHA-based protocols widely applied in EPCglobal C1 Gen2 standard. The performance of DFSA depends on both the accuracy of tag backlog estimation and the frame length setting [11]. To increase the accuracy of tag backlog estimation, most of the previous anti-collision protocols [3-5] require large computational load or large amount of memory. For ease of algorithm implementation, the author in [12] presents an anti-collision protocol which can be easily applied into a computation-limited reader and can achieve throughput close to the theoretical maximum (0.368).

To overcome the throughput limitation of DFSA, we propose an efficient tag identification protocol based on slot-hopped mechanism. Considering the disparity between slot durations, the slot-optimal algorithm may not be effective in terms of identification time. Thus, the time efficiency, identification speed, and other metrics have been taken into account in our scheme. Compared with conventional DFSA, our proposed scheme can achieve much higher time efficiency and faster identification speed.

The rest of this paper is organized as follows. The DFSA algorithm used in EPCglobal C1 Gen2 standard is briefly reviewed in Section 2. The proposed scheme is presented in Section 3. Simulation results and performance comparisons are given in Section 4. Finally, we draw conclusion in Section 5.

II. DFSA DESCRIPTION IN EPCGLOBAL C1 GEN2

The DFSA was applied in EPCglobal C1 Gen2 standard for solving the anti-collision problems. One of the primary features of DFSA is the dynamic adjustment of frame length. The performance of DFSA depends on whether frame length is accurately adjusted. To implement the DFSA, the

EPCglobal C1 Gen2 standard provides interrogator with series of commands, including Select, Query, QueryAdjust, QueryRep, and Ack. Before tags are identified, an interrogator first uses the Select command to select a particular tag group for further inventory and access. Second, the interrogator begins an inventory round by transmitting a Query command with parameter Q which is in the range of 0 to 15 and represents the value of frame length of 2Q. As the Query command is received, each tag generates a 16-bit random value RN16 and extracts a Q-bit subset from the RN16 as the tag's slot counter. This counter is decreased by command QueryRep. Until the counter reaches zero, the tag responds the interrogator with its RN16. The Query command has three possible outcomes: single tag reply, collided reply, and no reply, as shown in Fig. 1 [9].

If only one tag replies, the interrogator acknowledges the tag by sending an ACK command. If multiple tags reply, a collision of RN16s will be detected and no tags can be identified. These collision tags will contend again in the next rounds. If no tags reply, the interrogator will end the slot after a short period of waiting time. Until all 2Q time slots have been examined, the current inventory round finishes. The interrogator needs to start a new inventory round using the Query command to identify any collision tags. An example presented in Fig. 2 illustrates the anti-collision process with the DFSA scheme.

The reader starts first inventory round by using a Query command with the initial frame length of 4. Tag1 and tag 3 respond with their RN16s in time slot2. The collision of RN16s will be detected, and no tags can be identified. Tags 2 and 4 transmit their RN16s in slots 4 and 3, respectively, and thus can be successfully identified after the reader sends an ACK command. Since a collision occurs in time slot2 during the first inventory round, another round is required. Hence, the reader determines a new frame length 2, and broadcasts it by Query command. Finally, all tags were identified by the reader.

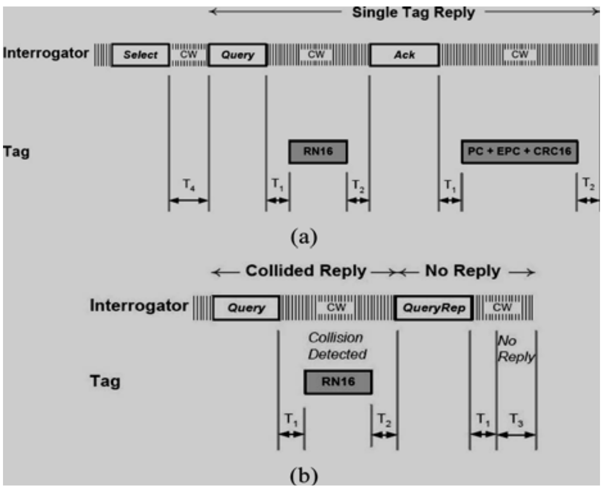

Fig. 1 Three outcomes for a given slot

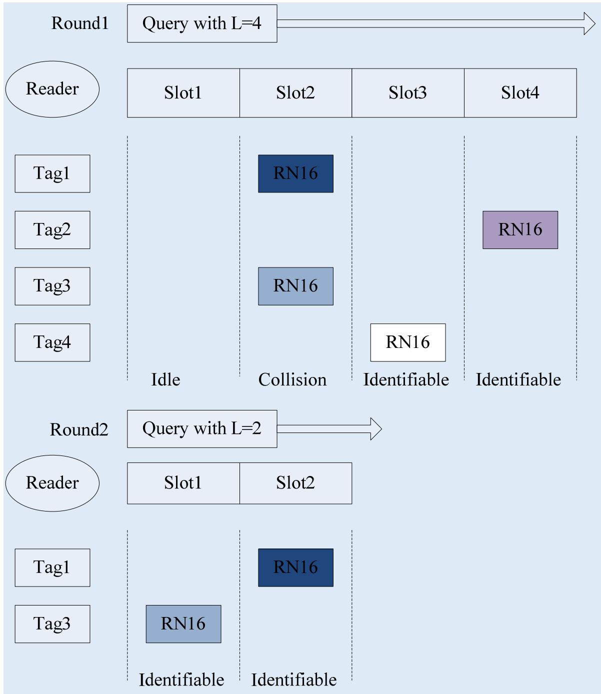

Fig. 2 an example with DFSA scheme

III. THE PROPOSED SCHEME

In this section, we present a novel protocol based on slot-hopped. We define the data format that tags respond to the Query command of a reader.

$$T_{resp}(i) = \left[RN11 \mid sc = i\right] + \left[RN5 \mid sc = i+1\right] \qquad (1)$$

where $T_{resp}(i)$ is denoted as the tags' response data in i-th time slot within a frame, and sc is the current slot counter of tag. From equation (1), we know the data format can be divided into two parts. The first part is tag's RN11 in current time slot I; the other part is the 5-length random sequence generated by tags whose sc equals i+1 after RN11 time delay. The RN5 part is used to predict the next slot state of a frame. When the reader receives the response data from tags in current slot i, there are four cases of RN5 as follows.

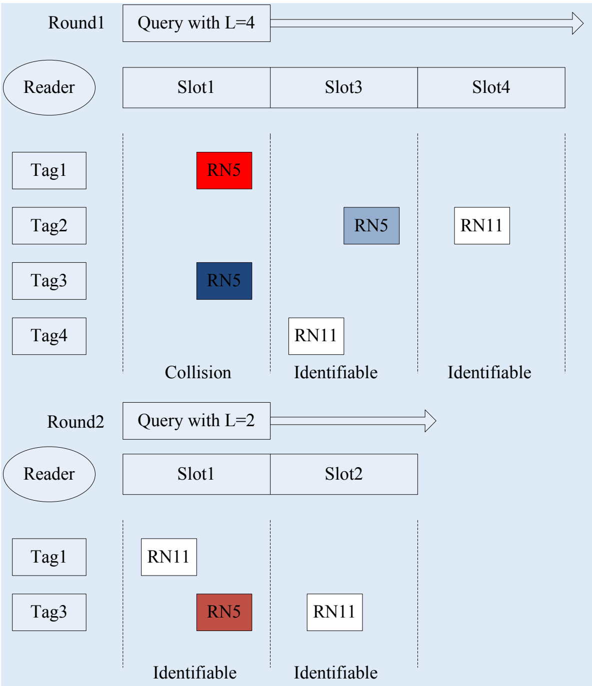

Fig. 3 An example with our proposed protocol

Case1: No tags generate the RN5. Accordingly, it demonstrates that the next slot i+1 is idle which can be hopped by the reader. In the next slot, the reader will allow tags whose sc equal i+2 to respond its request command.

Case2: The RN5 is generated by one tag; hence the reader can identify a tag successfully in the next slot i+1.

Case3: The RN5s are generated by more than one tag. That is to say, the next slot i+1 is the collision slot, similar to case1, the next slot i+1 can be hopped by the reader. In the next slot, the reader will allow tags whose sc equal i+2 to respond its request command.

Case4: Multiple tags generate the same 'RN5'. This will lead to 'RN5' collision. In other words, the next slot is failed to be hopped. The collision will occur because multiple tags will transmit their RN16s to the reader simultaneously.

We apply the proposed algorithm to the same previous example, where four tags exist within the reader's range. The identification procedure is illustrated in Fig. 3. We can compare the required time for four tags identification between Fig. 2 and Fig. 3. The required time for the example with DFSA scheme described in the Fig. 2 can be expressed as

$$T^1 = T_{idle} + T_{col} + 4 \times T_{succ} \tag{2}$$

where T_{idle}, T_{succ}, and T_{col} are the time durations for idle, successful, and collision slot can be written as:

$$T_{idle} = T_{QueryRep} + T_1 + T_3 \tag{3}$$

$$\begin{aligned} T_{succ} &= T_{Query} + 2(T_1 + T_2) + T_{RN16} \\ &+ T_{ACK} + T_{PC+EPC+CRC} \end{aligned} \tag{4}$$

$$T_{col} = T_{QueryRep} + T_1 + T_{RN16} + T_2 \tag{5}$$

According to the Fig. 3, the required time for all tags identification can be written as

$$T^2 = T_{col} + 4 \times T_{succ} \tag{6}$$

As time parameters specified in the EPCglobal C1 Gen2, T1, and T2 can be computed as 9100µs and 8800µs, respectively. Through this example, it may be seen that our proposed algorithm can identify all tags more rapidly than the traditional ALOHA-based algorithms.

Tag1 and tag3 select the same slot2, and generate RN5, (10010) and (11011), respectively. Subsequently, the 'RN5' collision occurs in the slot1; the reader can predict the slot2 is a collision slot. According to case3, the slot2 can be hopped by the reader.

Tag4 and tag2 send their RN16 in slots 3 and 4, respectively, and thus can be successfully identified by the reader because the two slots are singly occupied. Since slot2 is collision slot, this implies there are at least two tags, which need to be identified. Hence, another inventory round is required. In the round2, the reader completes the identification of all tags because no collision occurs in this round. Compared with Fig. 2, we know the lower number of slots is required to identify all tags by using our proposed scheme. Consider n tags that need to be identified using frame length L.

Assumed that the length of random sequence 5, which is used to predict a slot state. Let Pr denote the probability that r tags select the same slot, which can be written as

$$P_r = \binom{n}{r}\left(\frac{1}{L}\right)^r\left(1-\frac{1}{L}\right)^{n-r} \tag{7}$$

Accordingly, we obtain the probabilities of successful and collision for the slot as

$$P_s = \binom{n}{1}\left(\frac{1}{L}\right)\left(1-\frac{1}{L}\right)^{n-1} \tag{8}$$

$$P_c = 1-\left(1-\frac{1}{L}\right)^n - \binom{n}{1}\left(\frac{1}{L}\right)^1\left(1-\frac{1}{L}\right)^{n-1} \tag{9}$$

$$\begin{aligned} P_{fail-hop} &= \sum_{r=2}^{n} P_r \binom{2^5}{1}\left(\frac{1}{2^5}\right)^r \\ &= \sum_{r=2}^{n}\binom{n}{r}\left(\frac{1}{L}\right)^r\left(1-\frac{1}{L}\right)^{n-r}\binom{2^5}{1}\left(\frac{1}{2^5}\right)^r \end{aligned} \tag{10}$$

Furthermore, we denote $P_{fail\text{-}hop}$ as the probability that the collision slot is failed to be hopped. $P_{fail\text{-}hop}$ corresponds

DOI 10.5013/IJSSST.a.17.38.32 ISSN: 1473-804x online, 1473-8031 print

to the probability that r tags generate the same ‘RN5’ in the same slot. Since the collision slot will not be hopped when two or more tags generate the same ‘RN5’. We can have equation (10). In order to derive the maximum channel throughput, we assume that both collision slot and idle slot are stuck in the middle of successful slots which can be depicted in the Fig. 4.

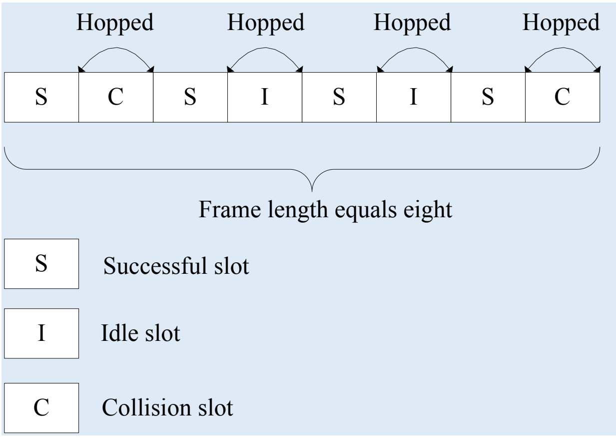

Fig. 4 An example of the proposed scheme

Let $Ps_{succ\text{-}hop}$ denote the probability that the collision slot is succeeded to be hopped. $P_{succ\text{-}hop}$ can be derived as

$$P_{succ-hop} = P_c - P_{fail-hop} = P_c - \sum_{r=2}^{n} P_r \binom{2^5}{1} \left(\frac{1}{2^5}\right)^r \tag{11}$$

Based on the analysis above, the ideal channel efficiency can be given by

$$U_{idea} = \frac{LP_s}{L - L \cdot P_{succ-hop} - L\left(1 - P_s - P_c\right)} \tag{12}$$

In the real RFID system, the situation depicted in Fig. 4 does not always occurs, the actual channel efficiency can be given by:

$$U = \frac{E(S)}{L - E(c) - E(e)} \tag{13}$$

where E(S) is the number of tags successfully identified by the reader during an inventory cycle with frame length L. E(c), and E(e) are the number of collision slots hopped and the number of idle slots hopped during an identification process, respectively. They are measured by the reader during the identification process. Considering the disparity between slot duration, the guarantee of achieving the optimal time efficiency must be taken into consideration in the design of the anti-collision algorithm. We defined time efficiency $T_{efficiency}$ as follow.

$$T_{efficiency} = \frac{ST_{succ}}{ST_{succ} + ET_{idle} + CT_{coll}} \tag{14}$$

Here, S, E, and C are the successful slots, idle slots and collision slots provided by the reader. T_{succ}, T_{idle} and T_{coll} are the time duration of S, E and C, respectively. For feasible and efficient to implement an anti-collision algorithm, a simple method is required to reduce the computation complexity for estimating the tag backlog.

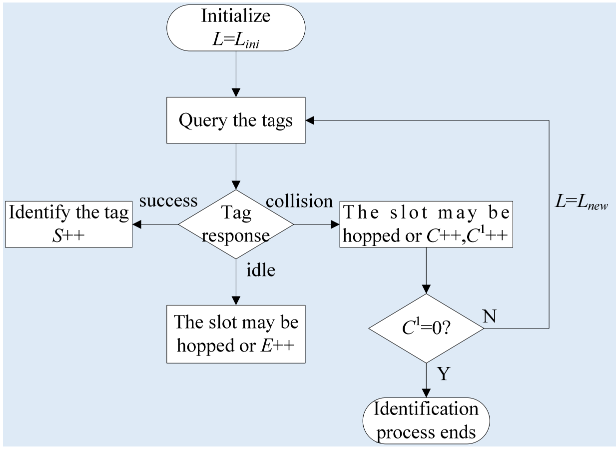

Fig. 5 The pseudo code of the slot-hopped ALOHA protocol

We adopt Schoute’s estimate method in our proposed algorithm. Thus, the tag backlog can be written as [4]

$$n_{est} = S^1 + 2.39C^1 \tag{15}$$

where S^1 is the number of the single occupied slots and C^1 is the collision slots (including the collision slots successfully hopped by the reader).

As analyzed above, our slot-hopped ALOHA protocol is summarized in Fig. 5.

IV. NUMERICAL AND ANALYTICAL RESULTS

In this section, the identification performance of the proposed algorithm and the reference methods was examined by our extensive simulations based on the Monte Carlo technique. We first compare the time efficiency of various anti-collision protocols including GDFSA [5], Q-algorithm [9], and FEIA [12]. To evaluate the specific time efficiency of our proposed algorithm, we need to calculate the time intervals of every step and command used in the anti-collision process. The primary time parameters used in the simulations are listed in Tab. 1. The parameters are set according to the EPCglobal C1 Gen2 standard.

The initial frame size is set to 64 when the number of tags increasing from 5 to 1500. Fig. 6 shows the simulation results for normalized time efficiency according to Eq. (14). The results show that our proposed scheme can achieve higher time efficiency. It can be found that as the number of tags increase, the performance of our scheme converges to 0.8, which is more than FEIA’s efficiency 0.7. Although FEIA’s time efficiency could keep stable by adopting the in-frame adjustment mechanism to fast and efficiently

DOI 10.5013/IJSSST.a.17.38.32 ISSN: 1473-804x online, 1473-8031 print

adjust frame length, it does not break through the limitation of DFSA.

TABLE. 1 PRIMARY TIME PARAMETER FOR EPCGLOBAL C1 GEN2

Parameters	Values	Parameters	Values
Data coding	Miller subcarrier modulation	Number of subcarrier cycles per symbol	4
R->T preamble	112.5μs	PC+EPC+CRC16	800μs
T->R preamble	112.5μs	T_1	62.5μs
Tari (reference time interval)	6.25μs	T_2	62.5μs
*RN*5	31.25μs	DR (divide ratio)	8
T_{succ}	2012.5μs	T_3	52.08μs
T_{idle}	300μs	T_4	15.625μs
T_{coll}	750μs	*RN11*	68.75μs

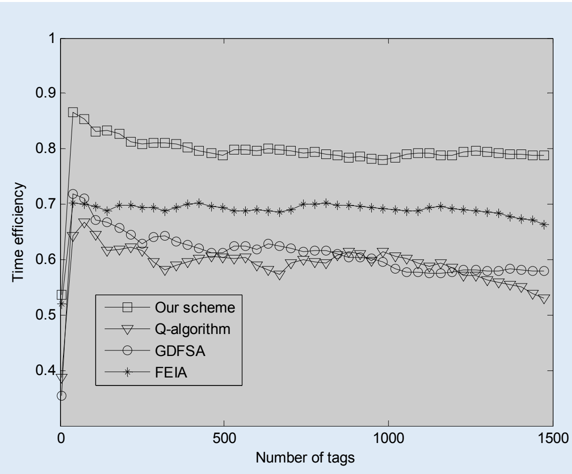

Fig. 6 Comparison of time efficiency for various algorithms

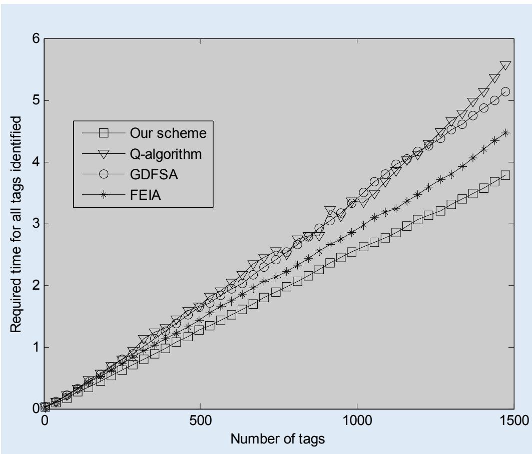

Fig. 7 Required time for all tag identified

Fig. 7 presents the simulation results for the total time required to identify all tags with initial frame size 64. To read 1300 tags, the proposed scheme spends about 3.30 seconds, i.e., the read speed is 400tags per second. Meanwhile, the read speed of Q-algorithm, GDFSA, and FEIA are 278tags/s, 286tags/s, and 345tags/s, respectively. In real RFID systems, a read speed is a key metric for multiple tags identification especially in logistics management. According to the simulation results, our algorithm could meet the speed requirement.

Moreover, in Fig. 8 we compare the complexity of communication of our anti-collision algorithm with the other algorithms. It gives average number of bits for identifying one tag when $5 \le n \le 1500$. Similar to the results for the time efficiency, our proposed algorithm has less complexity than the other algorithms. The communication overhead of our scheme is 408bits for identifying one tag. By contrast, FEIA, GDFSA, and Q-algorithm require 470bits, 535bits, and 550bits for one tag identification.

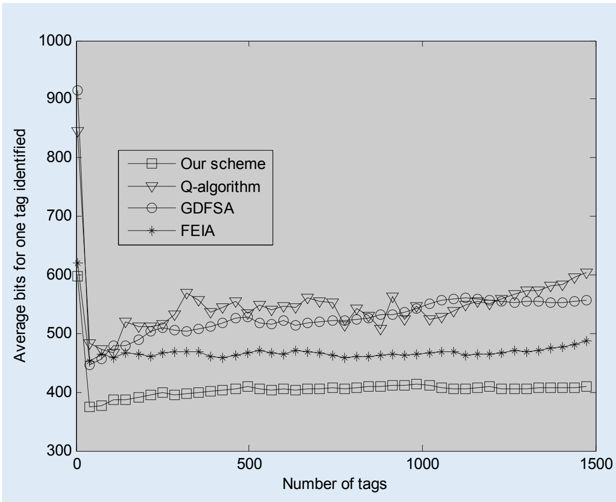

Fig. 8 Comparison of communication overhead for various algorithms

For better implementation of our proposed algorithm into actual UHF RFID systems, we also provide a comparison of complexity of tag's circuit, and computational complexity of algorithm. The results of the comparisons are shown in Tab. 2.

TABLE 2 COMPARISON OF OTHER METRICS

algorithms	Extra circuit requirement	Computational complexity
Our proposed scheme	(11+4) RNG	$O(1)$
Q-algorithm	(16+4) RNG	$O(1)$
GDFSA	(16+4) RNG	$O(N\text{-}C_1\text{-}2C_k\text{+}1)$
FEIA	(16+4) RNG	$O(1)$

where RNG is the random number generator, N is the maximum number of tags during the reader's coverage. C1 and Ck are numbers of successful slots and collision slots computed by the reader in a frame, respectively. The number of RNG required for our proposed scheme is significantly less than the size involved in the Q-algorithm, GDFSA, and FEIA algorithms. From comparisons above, we can know our algorithm, which requires low complexity, can be easily implemented on UHF RFID systems.

V. CONCLUSION

In this paper, a feasible and efficient anti-collision algorithm has been proposed for EPCglobal C1 Gen2 systems. Our proposed algorithm is based on the mechanism of slot-hopped and a simple but sufficiently accurate estimator for tag backlog. Analytical and simulation results show that the proposed algorithm outperforms existing ALOHA-based protocols.

REFERENCES

[1] K. Finkzeller, RFID Handbook, Fundamentals and Applications in Contacts smart Cards and Identification, Wiley, West Sussex, 2004.

[2] K. Dheeraj, K. W. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols", IEEE Commun. Surv. Tutor., Vol. 12, No. 3, pp. 400-421, 2010.

[3] H. Vogt, "Efficient object identification with passive RFID tags", in Int. Conf. Pervasive Comput., London, U. K., pp. 98-113, 2002.

[4] W.-T. Chen, "An accurate tag estimate method for improving the performance of an RFID anti-collision algorithm based on dynamic frame length ALOHA", IEEE Trans. Autom. Sci. Eng., Vol. 6, No. 1, pp. 9-15, 2009.

[5] Y. Pan, Q. Peng, Q. Zhou, et al., "Reducing tag collision in radio frequency identification systems by using a grouped dynamic frame slotted ALOHA algorithm", Acta Phys. Sin, Vol. 62, No. 14, pp. 148401-1-148401-8, 2013.

[6] Y. Wang, Y. Liu, H. Leung, et al., "A multi-bit identification protocol for RFID tag reading", IEEE Sensor Journal, Vol. 13, No. 10, pp. 3527-3536, 2013.

[7] Z. Liu, J. Ji, et al., "An anti-collision algorithm based on continuous collision bit detection", Acta Electronica Sinica, Vol. 41, No. 11, pp. 2156-2160, 2013.

[8] J. Song, Y. Guo, et al., "An adjustive hybrid tree anti-collision algorithm for RFID multi-tag identification", Acta Electronica Sinica, Vol. 42, No. 4, pp. 685-689, 2014.

[9] EPCglobal, EPC radio-frequency identify protocols class-1 generation-2 UHF RFID protocol for communications at 860 MHz-960 MHz ver. 1. 2. 0, Oct. 2008.

[10] C. Angerer, R. Langwieser, M. Rupp, "RFID reader receivers for physical layer collision revocery", IEEE Trans. Commun., Vol. 58, No. 12, pp. 3526-3537, 2010.

[11] W. Su, N. V. Alchazidis, and T. T. Ha, "Multiple RFID tags access algorithm", IEEE Trans. Mobile Computing, Vol. 3, No. 2, pp. 174-187, 2010.

[12] W.-T. Chen, "A feasible and easy-to-implement anticollision algorithm for the EPCglobal UHF Class-1 Generation-2 RFID protocol", IEEE Trans. Autom. Sci. Eng., Vol. 11, No. 2, pp. 485-491, 2014.