Efficient Estimation and Collision-Group-Based Anticollision Algorithms for Dynamic Frame-Slotted ALOHA in RFID Networks

Chun-Fu Lin and Frank Yeong-Sung Lin

***Abstract*—There are two challenges for the frame-slotted ALOHA algorithms in radio-frequency identification (RFID). The first challenge is estimating unknown tag-set size accurately; the second challenge is improving the efficiency of the arbitration process so that it uses less time slots to read all tags. This study proposes estimation algorithm based on the Poisson distribution theory and identifies the overestimation phenomenon in full collision. Our novel anticollision algorithm alternates two distinct reading cycles for dividing and solving tags in collision groups. This makes it more efficient for a reader to identify all tags within a small number of time slots.**

***Note to Practitioners*—The contribution of this study is to propose solutions to the estimation and arbitration problem on the RFID MAC layer protocol for the frame-slotted ALOHA. We present efficient estimation and anticollision algorithms to estimating and arbitrating an unknown quantity of tags. The full collision phenomenon and the impact to the estimation accuracy are studied. When full collision occurs, our proposed anticollision algorithm can be effectively solved and use less time slots to read all tags and it has the maximum throughput close to the theoretical value. The proposed algorithms are useful for engineers to implement the RFID systems by setting proper frame length to enhance the current RFID standards such as ISO 18000-6 or EPC class 1 generation 2.**

***Index Terms*—Anticollision, radio-frequency identification (RFID), tag-set size estimation.**

I. Introduction

THE RADIO-FREQUENCY IDENTIFICATION (RFID) technology uses radio waves to identify objects. The RFID tags can operate in adverse condition such as heat or damp and can be read by an RFID reader without line-of-sight. These advantages make the RFID to become an alternative approach as a replacement of the barcodes. The RFID tags can be classified into three categories: active, semi-passive, and passive tags. The active and semi-passive tags are equipped with batteries as partial or complete power sources. The passive tags do not have their own batteries; instead, they derive power from the radio wave, which when sent from the RFID readers, encounter their antennas forming a magnetic field. The effective communication distances between reader and tag vary from few centimeters to tens of meters apart. In general, the passive tags have shorter distances than the active tags and semi-passive tags.

Since the RFID can detect and track a variety of items in a quick and flexible way, it is used in many applications such as object tracking, inventory management, and supply-chain management. It will be widely spread in consumer products as the manufacture cost of RFID tags reduces. To secure and facilitate the end-to-end supply chains of global trade, the World Customs Organization [22] encourages applying high technologies such as RFID to build smart containers. Taiwan Customs has already adopted the passive RFID E-seal system in Kaohsiung harbor [13] to improve the transit containers' security and efficiency of the operation, as well as cutting costs by reducing manned escorts.

There are two types of RFID collision problems, a reader collision problem and a tag identification problem [14]. The reader collision problem [12] is caused by two adjacent readers intersecting their interrogation zones so that neither reader is able to communication with any tags located in this intersection. The reader collision problem is described as N-coloring problem which is NP-complete. It can be solved in distributed or centralized planning methods such as Colorwave [10] and simulated annealing [11].

The tag identification problem is caused by collisions in situations where more than one tag responds to a single reader's query. There are four different types of the anticollision procedures: space-division multiple access (SDMA), frequency-division multiple access (FDMA), code-division multiple access (CDMA), and time-division multiple access (TDMA). On account of the simplicity and limited functions of tag, TDMA procedures are the largest group in RFID systems [14].

TDMA procedures can be further classified into tag-driven procedures and reader-driven procedures. Tag-driven procedures are asynchronous operations and referred to as tag-talk-first (TTF). Reader-driven procedures are synchronous operations and referred to as reader-talk-first (RTF). Currently, the major trend of TDMA anticollision procedures is reader-driven. The reader-driven procedures have two different modes: deterministic and stochastic. The deterministic procedures are tree-based algorithms, such as binary-tree, query tree, and their variants [3], [15], [16]. By using the prefix bits, tree-based procedures mute subsets of tags to reduce the number of contenders for the channel and decrease the probability of collision.

Manuscript received December 02, 2009; accepted January 31, 2010. Date of publication March 11, 2010; date of current version October 06, 2010. This paper was recommended for publication by Associate Editor J. Li and Editor M. Zhou upon evaluation of the reviewers' comments.

The authors are with the Department of Information Management, National Taiwan University, Taipei 106, Taiwan (e-mail: d92002@im.ntu.edu.tw; yslin@im.ntu.edu.tw).

Digital Object Identifier 10.1109/TASE.2010.2042806

The stochastic procedures are based on ALOHA protocols. Shoute [21] shows that dynamic frame-slotted ALOHA could achieve better throughput than frame-slotted. The dynamic frame-slotted ALOHA algorithms [8], [9] restrict tags to randomly transmitting their IDs in bounded *frame size* $\boldsymbol{f}$ and adjust $\boldsymbol{f}$ for the next reading cycle if necessary. The reader broadcasts the request and each tag in its interrogation zone selects a random time slot in $\boldsymbol{f}$ then replies at that slot. The reader does not know the exact number of *tag-set size* $\boldsymbol{n}$ in its interrogation zone. However, it can obtain the values of $\boldsymbol{E}, \boldsymbol{S}$, and $\boldsymbol{C}$, which correspond to the *number of empty*, *occupied*, and *collision slots*, by detecting the reading results of $\boldsymbol{f}$ time slots. The values of $\boldsymbol{E}, \boldsymbol{S}, \boldsymbol{C}$ and $\boldsymbol{f}$ can be used to derive the *estimated tag-set size* $\tilde{\boldsymbol{n}}$.

Several anticollision algorithms for dynamic frame-slotted ALOHA [1], [2], [6], [19] calculate $\tilde{\boldsymbol{n}}$ in order to adjust $\boldsymbol{f}$ properly. The improper setting of $\boldsymbol{f}$ may causes more collisions or create more idle slots that decrease the throughput. Our research shows that *full collision* $(\boldsymbol{C} = \boldsymbol{f})$ affects the accuracy of the estimation algorithms. In this study, we focus on the frame-slotted ALOHA protocols and propose novel tag-set size estimations and anticollision algorithms to solve these problems.

II. Overview of the Frame-Slotted ALOHA Algorithms

There are two challenges for the frame-slotted ALOHA algorithms. First, we need to know how to estimate unknown tag-set size accurately. Second, we need to know how to improve the efficiency of the arbitration process so that it uses less time slots to read all tags.

In our study, most of the estimation algorithms are based on the binomial distribution theory. Given $\boldsymbol{f}$ and $\boldsymbol{n}$, the probabilities of zero, one and more than two tags transmitting in a slot are p_0, p_1 and p_c. They are defined as [1], [2], [6], [17]

$$p_0 = (1 - 1/f)^n \tag{1}$$

$$p_1 = (n/f)(1 - 1/f)^{n-1} \tag{2}$$

$$p_c = 1 - p_0 - p_1. \tag{3}$$

By multiplying $\boldsymbol{f}$ with (1) and (2), their expected values $a_0^{f,n}, a_1^{f,n}, a_c^{f,n}$ are

$$a_0^{f,n} = f \cdot p_0 \tag{4}$$

$$a_1^{f,n} = f \cdot p_1 \tag{5}$$

$$a_c^{f,n} = f - a_0^{f,n} - a_1^{f,n}. \tag{6}$$

Vogt [6] proposed two estimation algorithms. One is the lower bound estimation where $\tilde{\boldsymbol{n}}$ is calculated by $2 * \boldsymbol{C}$; the other is the minimum distance vector (MDV) based on Chebyshev's inequality theory. The tag-set size is estimated by varying $\boldsymbol{n}$ to find the minimum distance between $\langle \boldsymbol{E}, \boldsymbol{S}, \boldsymbol{C} \rangle$ and their expected values according to (4)–(6). The author modeled the arbitration process as a Markov decision process and proposed a lookup table to provide the optimal frame sizes according to estimated tag-set size $\tilde{\boldsymbol{n}}$. The optimal frame sizes are used to ensure that the arbitration process can guarantee the assurance level in minimum steps (cycles).

Lee *et al.* [4] claimed Vogt's arbitration process has poor performance when $\boldsymbol{n}$ is large. They proposed an anticollision algorithm called Enhance Dynamic Frame Slotted ALOHA (EDFSA) and use MDV for the estimation as well. EDFSA limits the number of contending tags to improve the arbitration efficiency. If there are collisions, EDFSA calculates the estimated tag-set size $\tilde{\boldsymbol{n}}$ and uses it to look for the corresponding $\boldsymbol{f}$ and *modulo value* $\boldsymbol{M}$ from the lookup table. Both values are broadcasted to all tags. On receiving the request, the tag generates its random number and divides this number by $\boldsymbol{M}$. Only those tags deriving zero remainder can join the contention of next reading cycle.

Cha *et al.* [1] proposed two estimation algorithms. Collision Ratio Estimation (CRE) algorithm defined collision ratio $\boldsymbol{C}_{\mathbf{ratio}} (= \mathrm{C}/\boldsymbol{f})$ and made it equals to (3). Estimated tag-set size $\tilde{\boldsymbol{n}}$ can be found by varying $\boldsymbol{n}$ to satisfy the equation as possible. The second algorithm called 2.39C which is based on the optimal collision rate theory and $\boldsymbol{n}$ can be estimated by $\boldsymbol{S} + 2.3922 \cdot \boldsymbol{C}$.

Wong *et al.* [7] proposed the Grouping Based Bit-Slot ALOHA (GBBSA) anticollision algorithm. Each tag has a reservation sequence (RS) which is 128 bits long. On receiving the reader's request, a tag will set a random bit of RS to 1 and the rest of the bits to 0. This bit represents the reservation time slot. GBBSA assume tags respond with their RSs at the same time and the reader can derive a sequence, including colliding bit-slots and zero bit-slots, from these RSs. Colliding bit-slot means that more than one tag has selected this time slot and zero bit-slot indicates there is no selection of this time slot. In consideration of efficiency, tags have to select a random value in $[0, 2^{\mathrm{Q}} - 1]$ that Q is an integer variable maintained by the reader. Only those tags that select zero value can reply their RSs immediately. The reader dynamically adjusts Q when the number of colliding bit-slots of RSs is less than 11 or larger than 20. The reader generates a request list according to the sequence and requests tags by scrolling tag IDs.

Chen [2] derived the estimation $\tilde{\boldsymbol{n}}$ by modeling the estimation problem as a multinomial distribution problem with $\boldsymbol{f}$ independent trials and exactly $\boldsymbol{E}, \boldsymbol{S}$, and $\boldsymbol{C}$ outcomes. The author used the multinomial estimation (ME) to estimate $\boldsymbol{n}$. The Dynamic Frame-slotted ALOHA algorithm (DFA) was presented in a way that set the frame size $\boldsymbol{f}$ to $\tilde{\boldsymbol{n}} - \boldsymbol{S}$ for the next reading cycle. The experimental results showed that ME has lower average estimate errors for $\boldsymbol{n} <= 250$ and DFSA outperformed the algorithms such as [6] that used fewer slots to read all tags.

Bonuccelli *et al.* [19] proposed a tree-based anticollision algorithm called Tree Slotted Aloha (TSA). The author proposed bounded MDV (BMDV) that enhance MDV by limiting the lower and upper bounds at $[\boldsymbol{S} + 2\boldsymbol{C}, 2(\boldsymbol{S} + 2\boldsymbol{C})]$ for calculating the distance values. The arbitration process follows a tree structure. After a reading cycle, if the reader detects collisions, it inserts new child nodes into the tree according to the number of the collisions, estimates the tag-set size, and decides the frame size of the child nodes. The reader goes down to the child nodes by following the depth-first order of tree and repeats the arbitration process. All the tags keep their random selection number and a tree level counter to ensure only those tags colliding in that child node can join the contention at the next reading cycle. TSA repeats the process recursively until all tags are read.

Maselli *et al.* [20] indicated the estimation of BMDV to be inaccurate when the collision rate is high. The authors claimed the calculation of the minimum vector distance can be stopped when the distance vector becomes extremely small. The Dynamic Minimum Distance Vector (DMDV) and Dynamic Tree Slotted ALOHA (DyTSA) were proposed as an enhancement of TSA. Unlike BMDV, DMDV has a loose upper bound where $\boldsymbol{n}$ is varied until the vector distance becomes less than one. DyTSA follows the depth-first order of TSA for the arbitration process. Instead of setting the estimated frame size for a collision node, DyTSA uses the data of the previous completely solved sibling nodes. It then sets the frame size of the child node to the average number of the previous completely solved sibling nodes.

III. The Tag-Set Size Estimation

The Poisson distribution [5] is a discrete probability distribution based on the number of events occurring in a fixed time period T. Herein, T is divided into $\boldsymbol{f}$ short slots of length $\boldsymbol{\Delta}\mathbf{t}$. The occurrence probability of an event in $\boldsymbol{\Delta}\mathbf{t}$ is $\boldsymbol{p}$. If $\boldsymbol{f}$ goes to infinity, the probability of exactly $\boldsymbol{k}$ occurrences of such event can be written as $\boldsymbol{P} = \boldsymbol{\lambda}^{\boldsymbol{k}} \boldsymbol{e}^{-\boldsymbol{\lambda}}/\boldsymbol{k}!$, where $\boldsymbol{\lambda}(= \boldsymbol{p} \cdot \boldsymbol{f})$ is the expected number of occurrences in T.

When $\boldsymbol{n}$ and $\boldsymbol{f}$ are large, $(1 - 1/\boldsymbol{f})^{\boldsymbol{n}}$ can be approximate to $e^{-\boldsymbol{n}/\boldsymbol{f}}$. Kodialam *et al.* [18] rewrote (4)–(6) and defined three estimators: Zero Estimator (ZE), Singleton Estimator (SE), and CE (Collision Estimator). The ZE and SE are used to develop the estimation algorithms for $\boldsymbol{n}$.

The Poisson distribution can be used as the approximation of a binomial distribution as the number of trials goes to infinity and the expected number of successes remains fixed. When the reader performs a reading cycle, an outcome that is composed of $\boldsymbol{E}, \boldsymbol{S}$, and $\boldsymbol{C}$ is observed. Note that in real world the values of $\boldsymbol{E}, \boldsymbol{S}$, and $\boldsymbol{C}$ are subject to $\boldsymbol{f}$, i.e., $\boldsymbol{f} = \boldsymbol{E} + \boldsymbol{S} + \boldsymbol{C}$. In our approximation model, we assume $\boldsymbol{f}$ goes to infinity and define (1)–(3) as the occurrence probabilities of three events $\mathrm{H}_1, \mathrm{H}_2$, and H_3. Let $\boldsymbol{\lambda_E} = \boldsymbol{p}_0 \cdot \boldsymbol{f}, \boldsymbol{\lambda_S} = \boldsymbol{p_1} \cdot \boldsymbol{f}$, and $\boldsymbol{\lambda_C} = \boldsymbol{p_c} \cdot \boldsymbol{f}$. The probabilities that exactly $\boldsymbol{E}$ occurrences of H_1, $\boldsymbol{S}$ occurrences of H_2, and $\boldsymbol{C}$ occurrences of H_3 are $\boldsymbol{P_E}, \boldsymbol{P_S}$, and $\boldsymbol{P_C}$ and they can be written as

$$P_E(E; \lambda_E) = \frac{\lambda_E^E e^{-\lambda_E}}{E!} \tag{7}$$

$$P_S(S; \lambda_S) = \frac{\lambda_S^S e^{-\lambda_S}}{S!} \tag{8}$$

$$P_C(C; \lambda_C) = \frac{\lambda_C^C e^{-\lambda_C}}{C!}. \tag{9}$$

The outcome is analogous to the one of the combinations of three events. Since no estimation is required if $\boldsymbol{C}$ is equal to 0, the possible four combinations are $\{\boldsymbol{P_C}, \boldsymbol{P_E P_C}, \boldsymbol{P_S P_C}, \boldsymbol{P_E P_S P_C}\}$. Each probability is no greater than one. Therefore, the estimation probability $\boldsymbol{P}_{\mathbf{est}}$ is defined as

$$P_{\text{est}} = \begin{cases} P_E P_S P_C, & \text{if } E, S, C \neq 0 \\ P_E P_C, & \text{if } S = 0 \text{ and } E, C \neq 0 \\ P_S P_C, & \text{if } E = 0 \text{ and } S, C \neq 0 \\ P_C, & \text{if } E, S = 0 \text{ and } C \neq 0 \end{cases} \tag{10}$$

The estimation value $\tilde{\boldsymbol{n}}$ of tag-set size is calculated as

$$\tilde{n} = \operatorname*{Arg\,Min}_{n} \{1 - P_{\text{est}}\}. \tag{11}$$

IV. The Simulation Results of Estimation Algorithms

The proposed algorithm is compared with ME [2], MDV [6], 2.39 C [1], CRE [1], BMDV [19], and DMDV [20] through Monte Carlo simulation. The frame size $\boldsymbol{f}$ was fixed to 128 for the comparison purpose. Tag-set size $\boldsymbol{n}$ was varied from 32 to 1024. The number of the full collision is recorded and divided by the number of the simulations to obtain the *full collision ratio* ***fc-ratio***. The *estimation error* $\boldsymbol{\varepsilon}\, (= |\tilde{\boldsymbol{n}} - \boldsymbol{n}|/\boldsymbol{n})$ defined in [2] was used to measure the estimation quality. We ran the simulation 10 000 times to average ε's.

During the implementation, we found the factorial computation at large values (>150) caused serious program overflows which impacted the estimation. Therefore we replaced and rewrote (7)–(9) by Stirling's approximation as follows:

$$P_E(E; \lambda_E) \cong \frac{\lambda_E^E e^{-\lambda_E}}{\sqrt{2\pi E} E^E e^{-E}} \tag{12}$$

$$P_S(S; \lambda_S) \cong \frac{\lambda_S^S e^{-\lambda_S}}{\sqrt{2\pi S} S^S e^{-S}} \tag{13}$$

$$P_C(C; \lambda_C) \cong \frac{\lambda_C^C e^{-\lambda_C}}{\sqrt{2\pi C} C^C e^{-C}}. \tag{14}$$

The logarithm and exponentiation were taken to transform the problem into an easy one. For example, if $\boldsymbol{E}, \boldsymbol{S}$ and $\boldsymbol{C}$ are not equal to zero, $\mathrm{P_{est}}$ can be written as

$$\begin{aligned} &P_{\text{est}}\{E, S, C \neq 0\} \\ &\cong \exp\left(\log\left(\frac{\lambda_E^E \lambda_S^S \lambda_C^C e^{-(\lambda_E+\lambda_S+\lambda_C)}}{\sqrt{2\pi E}\sqrt{2\pi S}\sqrt{2\pi C} E^E S^S C^C e^{-f}}\right)\right) \\ &= \exp(E \log \lambda_E + S \log \lambda_S + C \log \lambda_C - (\lambda_E + \lambda_S + \lambda_C) \\ &- \frac{1}{2}(\log 2\pi E + \log 2\pi S + \log 2\pi C) \\ &- (E \log E + C \log C + S \log S) + f). \end{aligned}$$

Other implementations are listed as follows.

1) ME: The factorial terms were also rewritten by Stirling's approximation and taken logarithm and exponentiation as

$$\begin{aligned} &\frac{f!}{E!S!C!} \\ &\cong \exp\left(\log\left(\frac{\sqrt{2\pi f} \cdot f^f \cdot e^{-f}}{\sqrt{2\pi E}\sqrt{2\pi S}\sqrt{2\pi C} E^E S^S C^C e^{-E} e^{-S} e^{-C}}\right)\right) \\ &= \exp\left(\frac{1}{2}\left(\log(2\pi f) - \log(2\pi E) - \log((2\pi S) - \log(2\pi C)\right)\right. \\ &\quad \left. + (f \log f - E \log E - S \log S - C \log C)\right). \end{aligned}$$

2) MDV: The minimum distance vector $\boldsymbol{\xi_n}$ was calculated as:

$$\begin{aligned} &\xi_n(f, E, S, C) \\ &\qquad = \min_n \left(\left|a_0^{f,n} - E\right| + \left|a_1^{f,n} - S\right| + \left|a_C^{f,n} - C\right|\right). \end{aligned}$$

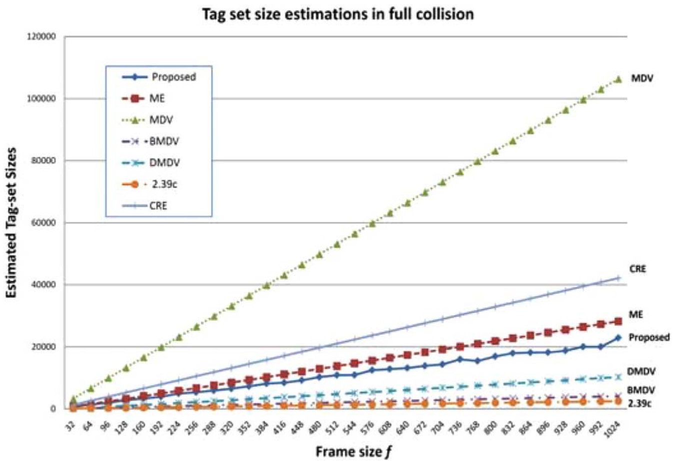


Fig. 1. The tag-set size estimations in full collisions ($E = 0, S = 0, C = f$) and f is varied from 32 to 1024.

3) CRE

$$\min_{n}\left|\frac{C}{f} - 1 + \left(1 - \frac{1}{f}\right)^{n}\left(1 + \frac{n}{f-1}\right)\right|.$$

4) BMDV and DMDV: The BMDV calculated ξ_n when $n \in [S+2C, 2(S+2C)]$. The DMDV has a loose upper bound when n is varied until $\xi_n < 1$.

The experimental results are listed in Table III. The proposed algorithm has a lower average ε for and outperforms others for $n \le 608$. However, for $n > 608$, DMDV apparently has a lower average ε than ours. In order to study this phenomenon, we conducted another experiment. We set $\langle E, S, C\rangle = \langle 0, 0, f\rangle$ to simulate the full collisions and varied f from 32 to 1024. As shown in Fig. 1 and Table II, the full collision estimations by 2.39 C, BMDV and DMDV are relative small compared to ME, CRE, MDV, and the proposed algorithm. Note $f = 128$, the full collision estimations are 983 by DMDV and 2759 by the proposed algorithm.

Fig. 2 shows the experimental results of the average estimate errors. Fig. 3 shows that the ***fc-ratio*** starts increasing at $n = 608$ and rapidly reaches 69.07% at $n = 1024$. We can see the average ε of DMDV inversely decreasing as n increases because DMDV has the full collision estimation of 983. Figs. 4 and 5 illustrates similar phenomenon which also appear on ME, CRE, MDV, and the proposed algorithm as the experiment was conducting to $n = 5000$ and the ***fc-ratio*** are 100%. Since 2.39 C and BMDV compute the full collision estimations at 307 and 512, respectively, as we know from Table I, the occurrence of full collision is zero at these tag-set sizes. Therefore, this phenomenon is not remarkable to 2.39 C and BMDV for $f = 128$.

TABLE I
THE PROCEDURE OF CGA

Cycle i
Step 1. The reader sends $\boldsymbol{R_1}$:{f}.
Step 2. An active tag randomly selects k; it set to k and transmits its ID in slot k in the range [1, f].
Step 3. The reader counts <E_i, S_i, C_i> and updates Δ. If C_i = 0, the algorithm stops; otherwise the reader perform the estimation algorithm and set f_g to min(128, max($\lceil(\tilde{n} - S_i)/C_i\rceil$, 2)).
Cycle i+1
Step 5. The reader sends $\boldsymbol{R_2}$:{f_g, Δ}.
Step 6. Tag verifies Δ. If tag is success, it deactivates itself. If tag has collision in slot k, tag calculates its collision group id q, where q = $_j$, for all $j \le k$.
Step 7. Tag randomly selects x in the range [1, f_g] and transmits its ID at slot $w = f_g \cdot (q - 1) + x$.
Step 8. The reader counts <E_{i+1}, S_{i+1}, C_{i+1}>. If C_{i+1} = 0, the algorithm stops; otherwise the reader set the next frame size f to min(128, max(($\tilde{n} - S_i - S_{i+1}$), 2)).
Step 9. Repeats the above procedures until algorithm stops.

TABLE II
THE TAG-SET SIZE ESTIMATIONS IN FULL COLLISION

n	Proposed	ME	MDV	BMDV	DMDV	2.39C	CRE
32	719	757	3276	128	193	77	1298
64	1338	1572	6604	256	440	154	2615
96	1953	2403	9931	384	705	230	3933
128	2759	3247	13258	512	983	307	5250
160	3309	4098	16585	640	1270	383	6568
192	3898	4957	19912	768	1564	460	7885
224	4935	5820	23239	896	1864	536	9203
256	5333	6689	26566	1024	2169	613	10520
288	5898	7562	29893	1152	2478	689	11838
320	6497	8438	33221	1280	2792	766	13155
352	7261	9318	36548	1408	3109	843	14473
384	8129	10201	39875	1536	3429	919	15790
416	8418	11087	43202	1664	3752	996	17108
448	9181	11975	46529	1792	4077	1072	18425
480	10213	12865	49856	1920	4406	1149	19743
512	10840	13758	53183	2048	4736	1225	21060
544	10899	14653	56510	2176	5069	1302	22378
576	12447	15550	59837	2304	5404	1378	23695
608	12796	16448	63165	2432	5741	1455	25013
640	13136	17349	66492	2560	6079	1532	26330
672	13815	18251	69819	2688	6420	1608	27648
704	14311	19154	73146	2816	6762	1685	28965
736	15923	20060	76473	2944	7106	1761	30283
768	15401	20966	79800	3072	7451	1838	31600
800	16879	21873	83127	3200	7797	1914	32918
832	17921	22783	86454	3328	8146	1991	34235
864	18117	23694	89781	3456	8495	2067	35553
896	18168	24605	93108	3584	8846	2144	36870
928	18732	25518	96436	3712	9198	2220	38188
960	20032	26433	99763	3840	9551	2297	39505
992	20019	27347	103090	3968	9905	2374	40823
1024	22897	28264	106417	4096	10260	2450	42140

V. THE COLLISION GROUP ALGORITHM

Kodialam [18] defined the *load factor* $\rho = n/f$ and found the expected number of the occupied slots attain a maximum when $\rho = 1$, i.e., $n = f$. Lee [4] and Chen [2] also derived the same conclusions in regard to the *system utility* U

$$U = \frac{\text{The expected number of occupied slots}}{\text{The current frame size f}}.$$

By taking the first derivative of U with respect to f, the maximum efficiency happens when f equals n. Since n is unknown, therefore, f will be dynamically adjusted to the estimated tag-set size $\tilde{n}$.

The experimental results in the previous section show that our proposed estimation algorithm calculates high value in full collision. If the frame size is set to such value, it could be inefficient

TABLE III
THE AVERAGE ESTIMATE ERRORS OF DIFFERENT ESTIMATION ALGORITHMS

n	Proposed	ME	MDV	BMDV	DMDV	2.39C	CRE	fc-ratio
32	0.929375%	0.929375%	5.937188%	3.477188%	6.287188%	4.711875%	20.466563%	0.00%
64	1.903125%	1.903125%	5.384688%	4.394375%	5.198125%	4.753125%	9.076563%	0.00%
96	2.235000%	2.235000%	5.395521%	5.110417%	5.474687%	3.622708%	5.694479%	0.00%
128	2.589844%	2.589844%	4.006641%	3.966875%	4.042266%	2.326953%	4.592031%	0.00%
160	2.870625%	2.870625%	4.051625%	4.051125%	4.054562%	3.317688%	4.062812%	0.00%
192	3.143490%	3.143490%	3.936042%	3.936042%	3.929323%	6.827083%	3.965885%	0.00%
224	3.376920%	3.376920%	3.855089%	3.855089%	3.877455%	10.875670%	3.856339%	0.00%
256	3.619688%	3.619688%	4.041992%	4.041992%	4.026094%	15.093867%	4.034375%	0.00%
288	3.808160%	3.808160%	4.072049%	4.072049%	4.056319%	19.292639%	4.060833%	0.00%
320	4.196875%	4.196875%	4.427250%	4.427250%	4.454844%	23.370469%	4.415375%	0.00%
352	4.438381%	4.438381%	4.617472%	4.617472%	4.635682%	27.262443%	4.629716%	0.00%
384	4.704349%	4.707109%	4.858854%	4.857344%	4.929323%	30.947083%	4.833750%	0.00%
416	5.070168%	5.074014%	5.154904%	5.075457%	5.185000%	34.479976%	5.132380%	0.00%
448	5.303192%	5.329933%	5.406496%	4.689219%	5.375937%	37.757321%	5.373683%	0.00%
480	5.695188%	5.754521%	5.838021%	3.012792%	5.840167%	40.821354%	5.863771%	0.00%
512	6.022754%	6.135000%	6.223242%	5.840859%	6.382656%	43.702051%	6.243301%	0.00%
544	6.502169%	6.675441%	6.765129%	10.098162%	6.735809%	46.374007%	6.721673%	0.00%
576	6.963681%	7.157292%	7.412760%	14.289132%	7.284253%	48.857135%	7.222760%	0.00%
608	7.622270%	8.135197%	9.858109%	18.217763%	7.777730%	51.167599%	8.510000%	0.10%
640	8.765016%	9.907766%	16.341141%	21.853969%	8.481750%	53.308500%	11.197922%	0.39%
672	10.433616%	12.914777%	30.058661%	25.237530%	9.606354%	55.303601%	16.375372%	1.04%
704	13.048324%	18.180057%	55.167457%	28.372926%	11.143395%	57.157344%	25.639858%	2.80%
736	17.849878%	27.545965%	103.078519%	31.275462%	13.051182%	58.881753%	42.689633%	5.74%
768	23.497656%	40.174284%	171.575964%	33.975130%	13.596133%	60.487305%	66.462812%	9.74%
800	31.133200%	55.774050%	254.767225%	36.503525%	13.369200%	61.995037%	95.600825%	15.92%
832	38.638690%	73.774916%	354.154183%	38.842428%	13.109327%	63.389627%	129.885601%	23.71%
864	43.711319%	86.936227%	432.008333%	41.043843%	11.765752%	64.703299%	155.986435%	30.61%
896	53.453783%	108.123158%	551.258415%	43.088504%	9.695100%	65.921172%	196.793839%	39.77%
928	59.300539%	123.223308%	635.546886%	45.003491%	6.946692%	67.063244%	225.739041%	47.32%
960	64.063979%	136.829302%	714.044229%	46.804167%	4.027083%	68.136323%	252.331635%	55.13%
992	66.149143%	144.497631%	763.337056%	48.494536%	3.296714%	69.145222%	268.325786%	61.90%
1024	68.600342%	154.457324%	828.264170%	50.080547%	5.681611%	70.090244%	289.287891%	69.07%

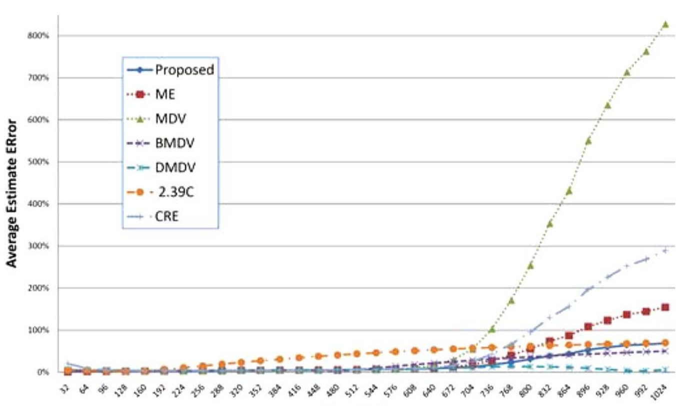


Fig. 2. The experimental results when tag-set sizes are varied from 32 to 1024 and f is fixed to 128.

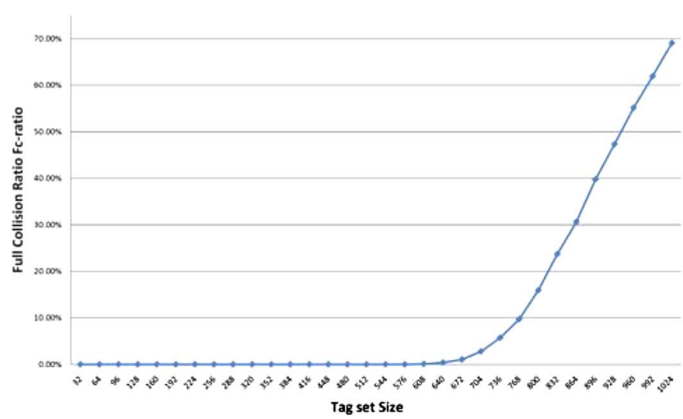


Fig. 3. The full collision ratio ***fc-ratio*** when tag-set sizes are varied from 32 to 1024 and f is fixed to 128.

in high tag-density environment. Our proposed anticollision algorithm called Collision Group Algorithm (CGA) basically follows the principle of $f = \tilde{n}$. However, we use the divide and conquer methodology to improve the arbitration efficiency. The operation scenario of CGA is described as follows.

1) A time slot has enough time either for a reader to send request, or for a tag to respond and receive acknowledgment (ACK) from a reader [20]. If collision occurs, the reader will not send ACK.
2) An RFID tag can deactivate itself when it receives ACK.
3) A reading cycle includes the time slots needed for the reader's request and relative tags response.
4) The reader maintains a *bit string* $\Delta = [\delta_1 \delta_2 \ldots]$ with maximum bit length of 128 long to record the collision status. If slot j collides, δ_j is set to 1.
5) The reader has two request commands R_1 and R_2 that have different message headers for tags to distinguish from. The reader sends R_1 and R_2 alternately. R_1 includes frame size f. R_2 includes *group frame size* f_g and collision status Δ.
6) Tag maintains a local variable γ to remember its selection. If tag answers at slot j, it set γ to j.

Each CGA loop has two reading cycles. In the first cycle, the reader sends R_1 including frame size f to all tags. On receiving R_1, the active tags answer at their randomly selected time slots within f. The reader observes the reading results and updates

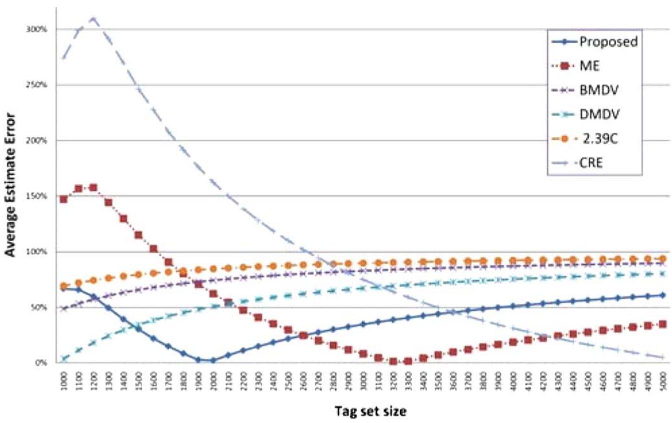


Fig. 4. The experimental results when tag-set sizes are varied from 1000 to 5000 and $\boldsymbol{f}$ is fixed to 128.

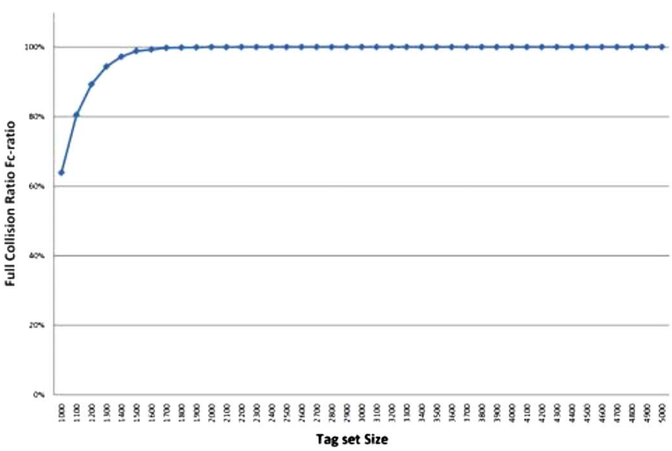


Fig. 5. The full collision ratio ***fc-ratio*** when tag-set sizes are varied from 1000 to 5000 and $\boldsymbol{f}$ is fixed to 128.

the collision status Δ. If the number of collision $\boldsymbol{C_i} \neq 0$, the reader performs the estimation algorithm to derive $\tilde{\boldsymbol{n}}$ and set group frame size $\boldsymbol{f_g}$ to $\min(128, \max(\lceil \tilde{n}/C \rceil, 2))$. The frame size of second cycle is set to $\boldsymbol{C_i} * \boldsymbol{f_g}$.

In the second cycle, the reader sends $\boldsymbol{R_2}$ including group frame size $\boldsymbol{f_g}$ and collision status Δ to all tags. On receiving $\boldsymbol{R_2}$, each active tag calculates its *collision group number* $\boldsymbol{q}$ by summing up 1's from the first bit to $\boldsymbol{\gamma}$th bit in Δ; Tag randomly selects a number $\boldsymbol{x}$ in $[1 \ldots \boldsymbol{f_g}]$ and transmit its ID in a time slot $\boldsymbol{w}$, where $\boldsymbol{w} = \boldsymbol{f_g} \cdot (\boldsymbol{q}-1)+\boldsymbol{x}$. The reader observes the reading results. If the number of collisions $\boldsymbol{C_{i+1}} \neq 0$, the frame size for the next iteration $\boldsymbol{f}$ is set to $\min(128, \max((\tilde{\boldsymbol{n}} - \boldsymbol{S_i} - \boldsymbol{S_{i+1}}), 2))$. The reader continues the arbitration process until no collision occurs.

The procedures of the CGA are listed as Table I.

Fig. 6 illustrates an example of CGA iteration. Suppose $\boldsymbol{f}$ is set to 4. At the beginning of cycle $\boldsymbol{i}$, the reader sends $\mathrm{R}_1 : \{\boldsymbol{f} = 4\}$. The active tags $\mathrm{t}_1, \mathrm{t}_2, \mathrm{t}_3, \mathrm{t}_4$, and t_5 receive the request and transmit their IDs at randomly selected time slots: 1, 3, 3, 4, and 4, respectively. After cycle $\boldsymbol{i}$, the reader obtains the reading results $\langle \boldsymbol{E_i}, \boldsymbol{S_i}, \boldsymbol{C_i} \rangle = \langle 1, 1, 2 \rangle$. It updates Δ to 0011 and performs the estimation to obtain $\tilde{\boldsymbol{n}} = 6$. The group frame size $\boldsymbol{f_g}$ is therefore set to 3.

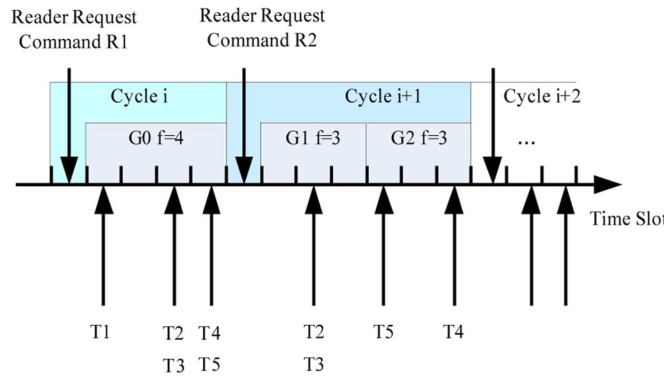


Fig. 6. A reading example of CGA.

In cycle $\boldsymbol{i}+1$, the reader sends $\mathrm{R}_2 : \{\boldsymbol{f}_{\mathrm{g}} = 3, \Delta = 0011\}$. Tags t_2 and t_3 calculate their collision groups number $\boldsymbol{q} = 1$. Tags t_4 and t_5 also calculate their collision groups as $\boldsymbol{q} = 2$. Assuming the random selections of $\mathrm{t}_2, \mathrm{t}_3, \mathrm{t}_4$, and t_5 are 2, 2, 1, and 3. Tags transmit their IDs at time slots 2, 2, 4, and 6. The reader obtains the result $\langle \boldsymbol{E}_{\mathbf{i+1}}, \boldsymbol{S}_{\mathbf{i+1}}, \boldsymbol{C}_{\mathrm{i+1}} \rangle = \langle 3, 2, 1 \rangle$ and let $\boldsymbol{f} = 3(6-1-2)$ for the next iteration.

VI. The Performance Analysis for the Anticollision Algorithms

The CGA was compared with that of DFA [2], EDFSA [4], GBBSA [7], and DyTSA [20] on the basis of whether all tags were successfully read from different tag-set sizes. The simulations were run 1000 times to obtain the average number.

We compare the read performance of CGA with that of other algorithms. The calculations of time slots are described as follows.

1) CGA: The reader sends two requests and use $\boldsymbol{f}$ slots for the first cycle. The reader performs the estimation and uses $\boldsymbol{C_i} * \boldsymbol{f_g}$ slots for the next cycle. Therefore, $1+\boldsymbol{f}+1+\boldsymbol{C_i}*\boldsymbol{f_g}$ time slots are required for iteration.
2) DFA: The reader sends one request and uses $\tilde{\boldsymbol{n}} - \boldsymbol{S}$ time slots for tags to reply. The reader performs ME for the setting of the next frame size. Therefore, $1 + \tilde{\boldsymbol{n}} - \boldsymbol{S}$ time slots are required for iteration.
3) EDFSA: The reader sends one request and uses $\boldsymbol{f}$ slots for tags to reply according to the estimation and the lookup table. Therefore, $1 + \boldsymbol{f}$ time slots are required for single iteration.
4) GBBSA: The reader sends one request command and the tags that randomly select zero value reply their RSs at once in the next time slot. These RSs arrive at the same time and form a new RS. The reader scrolls IDs and requests for tag's reply according to the new RS which bits are not equal to zero. The GBBSA requires 2 (for request and reply RSs) $+2(\boldsymbol{S}+\boldsymbol{C})$ (for scrolling IDs) time slots for iteration.
5) DyTSA: The reader sends one request command and uses $\boldsymbol{f}$ slots for tags to reply. New nodes are created according to the number of collisions and they are inserted into the tree. The DMDV estimation is performed to decide the frame size of node. If the sibling nodes are completely solved, the frame size is set to the average tag-set size number of

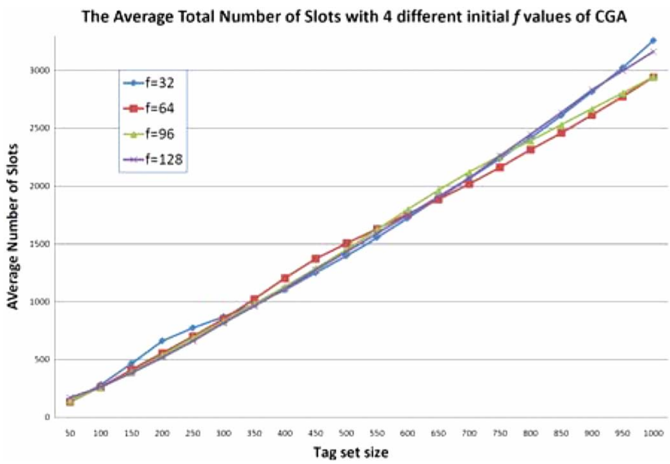


Fig. 7. Comparison of the simulation results for CGA with four different initial f values.

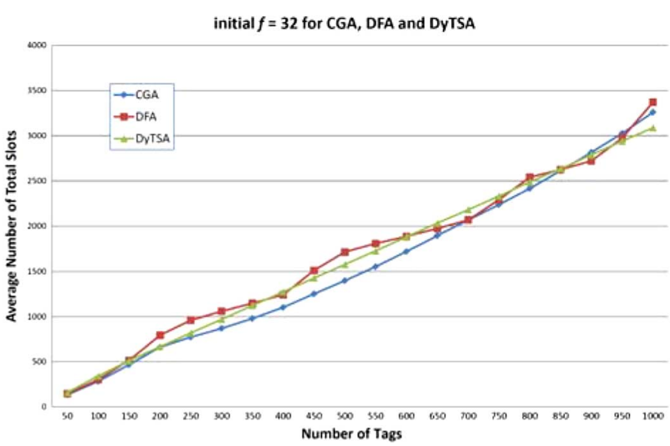


Fig. 8. Comparison of the simulation results for CGA, DFA, and DyTSA. The initial frame sizes were set to 32.

the sibling nodes. The arbitrations are repeated recursively in depth-first order. Therefore, DyTSA requires $1 + f$ slots for iteration.

The different initial frame size affects the read performance. Fig. 7 shows the simulation results of the CGA with different sizes of initial frame f. The hump on the line indicates the influence of full collision as small initial f meets large n. The experiment also indicates that small initial frame size could use less slots than large initial frame size ones to read all tags. For example, when $650 \leq n \leq 1000$, initial frame size $f = 64$ outperforms $f = 128$. For any tag population within that range, full collision is possible to occur. The CGA estimates $\tilde{n} = 1338$ with $f = 64$ or $\tilde{n} = 2759$ with $f = 128$. In such extreme condition, the CGA could be more efficient with the estimation close to the tag population.

Figs. 8–10 presents the average required time slots for CGA, DFA, and DyTSA with three different sizes of initial f. EDFSA and GBBSA both have the frame size of 128 are compared as Fig. 11. Note that both DyTSA and CGA depend on the estimation of tag-set size. DyTSA follow depth-first tree order and the reader has to calculate and broadcast the node frame size for every child node. In CGA, the colliding tags can calculate

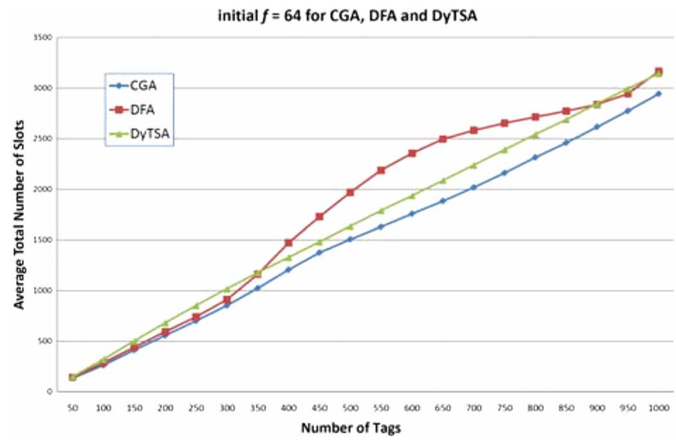


Fig. 9. Comparison of the simulation results for CGA, DFA, and DyTSA. The initial frame sizes were set to 64.

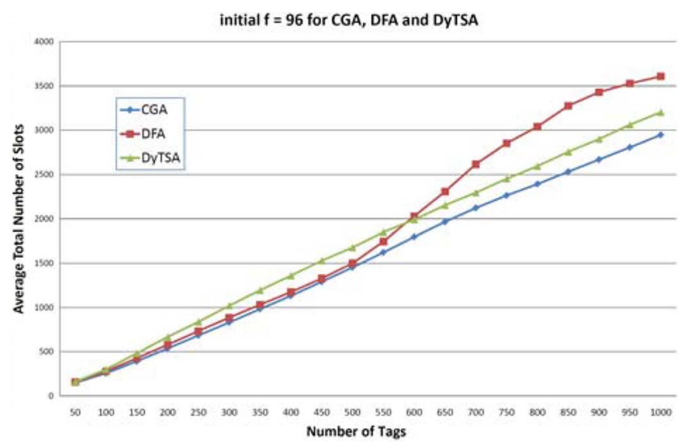


Fig. 10. Comparison of the simulation results for CGA, DFA, and DyTSA. The initial frame sizes were set to 96.

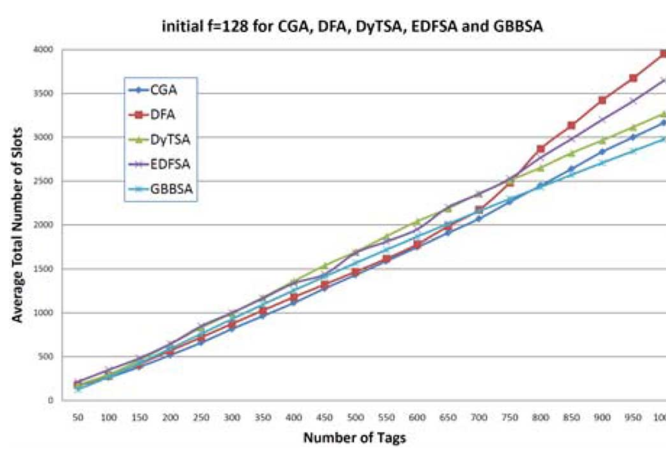


Fig. 11. Comparison of the simulation results for CGA, DFA, DyTSA, EDFSA, and CBBSA. The initial frame sizes were set to 128.

its own collision group from Δ. The experiments show that, in most cases, the CGA uses fewer slots to read all the tags and can have a better performance.

The normalized throughput is defined as the number of tags divided by the number of required time slots that are used to measure the channel's performance. The initial frame size

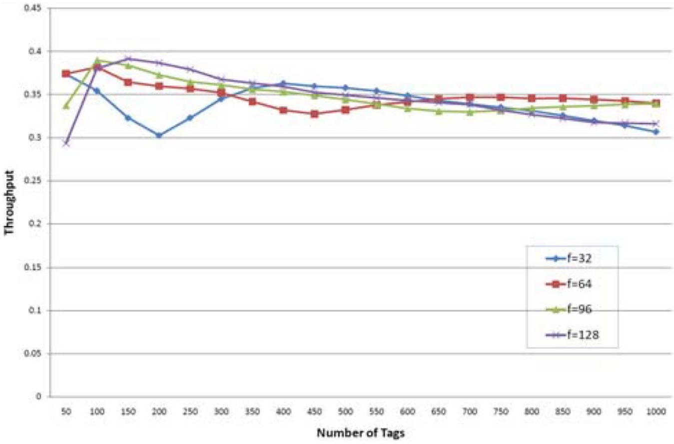


Fig. 12. The comparison for different initial frame size of CGA on normalized throughput.

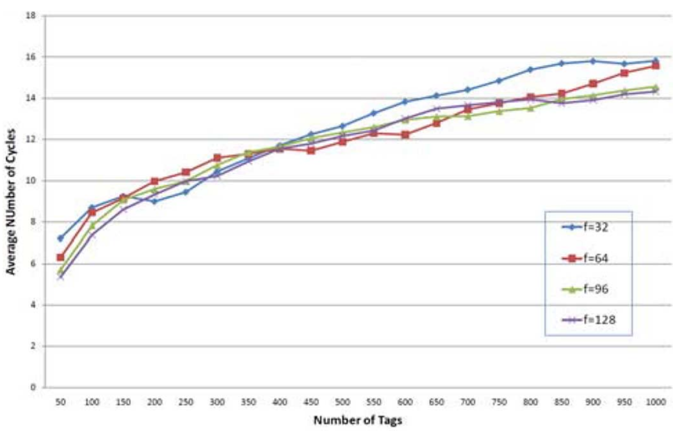


Fig. 13. Comparison of the average number of cycles for CGA with different initial frame size.

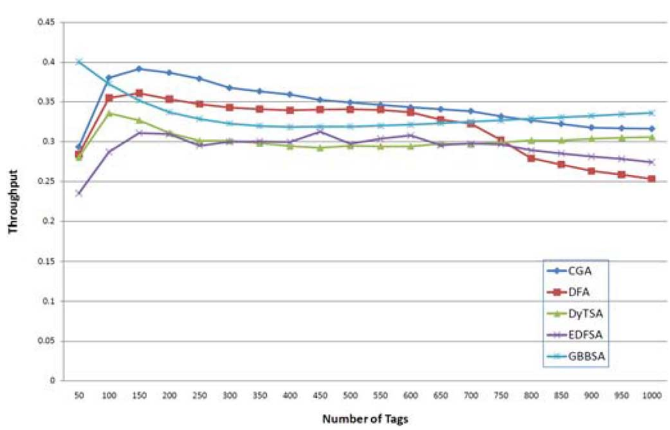


Fig. 14. Comparison of the normalized throughput for CGA, DFA, DyTSA, EDFSA, and CBBSA. The initial frame sizes were set to 128.

also influences the throughput. Fig. 12 shows the maximum throughput occurs when the tag-set size is approximately equal to frame size. It also shows the effect of overestimation at full collision and its influence on the throughput. For example, when initial $\boldsymbol{f} = 32$ and $n = 200$, CGA estimates and allocates the frame size to 719 that decreases the throughput. This value is obviously too large to solve the collisions. As we can see in Fig. 13, the average number of cycles is therefore decreasing. However, CGA can efficiently solve the collisions and its throughput is outperformed to other algorithms in most cases, as shown in Fig. 14.

VII. Conclusion

This paper presents efficient estimation and anticollision algorithms; for the estimation and the arbitration to an unknown quantity of tags. The proposed estimation algorithm has low average estimation errors in a high tag density environment. When full collision occurs, our proposed anticollision algorithm can effectively solve and use less time slots to read all tags. The simulation results show that proposed algorithms outperform others.

References

[1] J.-R. Cha and J.-H. Kim, "Novel anti-collision algorithms for fast object identification in RFID systems," in *Proc. Int. Conf. Parallel and Distrib. Syst. Comput.*, 2005, vol. 2, pp. 63–67.

[2] 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.*, no. 99, pp. 1–7, 2008.

[3] K. Finkenzeller, *RFID Handbook*, 2nd ed. New York: Wiley, 2003.

[4] S.-R. Lee, S.-D. Joo, and C.-W. Lee, "An enhance dynamic framed slotted ALOHA algorithm for RFID tag identification," in *Proc. Int. Conf. Mobile and Ubiquitous Systems: Networking and Services*, 2005, pp. 166–174.

[5] D. E. Knuth, *The Art of Computer Programming*. Reading, MA: Addison Wesley, vol. 2.

[6] H. Vogt, "Efficient object identification with passive RFID tags," in *Proc. Int. Conf. Pervasive Computing*, 2002, pp. 98–113.

[7] C.-P. Wong and Q.-Y. Feng, "Grouping based bit-slot ALOHA protocol for tag anti-collision in RFID systems," *IEEE Commun. Lett.*, vol. 11, no. 12, Dec. 2007.

[8] *UHF Air Interface Protocol Standards*, EPCglobal,C1G2, 2005.

[9] *Parameters for Air Interface Communications at 860 MHz to 960 MHz Information Technology—Radio Frequency Identification for Item Management*, ISO/IEC 18000-6, 2004.

[10] J. Waldrop, D. W. Engels, and S. E. Sarma, "Colorwave: An anticollision algorithm for the reader collision problem," in *Proc. IEEE ICC*, 2003, vol. 2, pp. 1206–1210.

[11] C.-F. Lin and F. Y. S. Lin, "A simulated annealing algorithm for RFID reader networks," in *Proc. IEEE WCNC*, 2007, pp. 1669–1672.

[12] D. W. Engels and S. E. Sarma, "The reader collision problem," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2002, vol. 3, pp. 6–9.

[13] "Taiwan customs officials adopt RFID-enabled container seals," *RFID Journal* 2009. [Online]. Available: http://www.rfidjournal.com/article/view/4727,

[14] D.-H. Shih, P.-L. Sun, D. C. Yen, and S.-M. Huang, "Taxonomy and survey of RFID anti-collision protocols," *Comput. Commun.* 2006, pp. 2150–2166.

[15] C. Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification," in *Proc. 4th Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, 2000, pp. 75–84.

[16] J. Myung, W. Lee, and T. K. Shih, "An adaptive memoryless protocol for RFID tag collision arbitration," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 1096–1101, Oct. 2006.

[17] D. K. Klair, K. Chin, and R. Raad, "On the accuracy of RFID tag estimation functions," in *Proc. ISCIT*, Oct. 17–19, 2007, pp. 1401–1406.

[18] M. S. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in RFID systems," in *Proc. MOBICOM*, 2006, pp. 322–333.

[19] M. A. Bonuccelli, F. Lonetti, and F. Martelli, "Instant collision resolution for tag identification in RFID networks," *Elsevier Ad Hoc Networks* pp. 1220–1232, Nov. 2007.

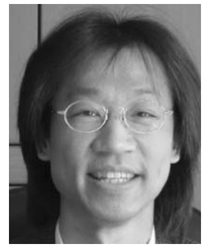
[20] G. Maselli, C. Petrioli, and C. Vicari, "Dynamic tag estimation for optimizing tree slotted aloha in RFID networks," in *Proc. MSWiM*, 2008, pp. 315–322.

[21] F. C. Schoute, "Dynamic frame length ALOHA," *IEEE Trans. Commun.*, vol. COM-31, no. 4, pp. 565–568, Apr. 1983.

[22] *WCO SAFE Framework of Standards*, , Jun. 2007, World Customs Organization.

Chun-Fu Lin is working towards the Ph.D. degree at the Department of Information Management, National Taiwan University, Taipei. since 2003.

He is also a Taiwan Customs Officer. Currently, he works on the Taiwan government's project to implement the applications of RFID e-Seal in cross-border supply chain security. His research interests include network optimization, network planning, anticollision algorithm, and RFID networks.



Yeong-Sung (Frank) Lin received the B.S. degree in electrical engineering from the Department of Electrical Engineering, National Taiwan University, Taipei, in 1983, and the Ph.D. degree in electrical engineering from the Department of Electrical Engineering, University of Southern California (USC), Los Angeles, in 1991.

After graduating from the USC, he joined Telcordia Technologies (formerly, Bell Communications Research, abbreviated as Bellcore) in New Jersey. In 1994, he joined the faculty of the Electronic Engineering Department, National Taiwan University of Science and Technology. Since 1996, he has been with the faculty of the Information Management Department, National Taiwan University. His research interests include network optimization, network planning, network survivability, performance evaluation, high-speed networks, distributed algorithms, content-based information retrieval/filtering, biometrics and network/information security.