An Enhanced RFID Multiple Access Protocol for Fast Inventory

You-Chang Ko, Sumit Roy
Dept. of EE , Univ. of Washington
Seattle, WA 98195 USA
{ycko,sroy}@u.washington.edu

Joshua R. Smith
Intel Research Lab.
Seattle, WA 98105 USA
joshua.r.smith@intel.com

Hyung-Woo Lee, Choong-Ho Cho
Dept. of CIS & EIE, Korea Univ.
Chochiwon, S. Korea
{hwlee,chcho}@.korea.ac.kr

***Abstract* - The relevant performance metric for successful deployment of Radio Frequency Identification (RFID) systems for tag inventory applications is the latency for reading all tags with (high) reliability. Tag collisions in response to a reader query increase the read latency of the MAC protocol; the mean latency can be considerably improved by a combination of techniques including more efficient *anti-collision* approaches as well as via *estimation of the number of backlogged tags*. We propose a novel anti-collision algorithm: *breadth-first-search with m-ary splitting* (BMSA) within a TDMA frame structure. A simple backlogged tag estimation algorithm is used in conjunction with the above to dynamically set the *splitting factor* (SF) *m*. Simulation results demonstrate the superiority of the proposed scheme over existing methods in terms of throughput/latency.**

Key words*: *RFID, Backlogged tag estimation, Anti-collision, Framed ALOHA, Tree Splitting

I. INTRODUCTION

Radio Frequency Identification (RFID) is a promising technology for identification of tagged object for real-time inventory detection and supply chain management. In passive RFID systems, a reader chooses a group of tags within its read range (via the *select* command using mask bits and specifying a contention window for the tag responses). Each tag whose ID matches the mask bits responds within one of the slots in the contention window potentially leading to tag collision. If the reader detects any collision after one read cycle, (referred to as a *round*), it issues a *query* command for a new round with the proper parameter that controls the transmission probabilities for the *backlogged tags*(n_b) whose IDs have not yet been read among the *initially selected tag pool*(n_b^0). The number of rounds continues until the number of backlogged tags approaches zero in either a *deterministic* or *statistical sense*, thereby ending the *session*. In typical inventory applications, a large initial n_b^0 communicating via a shared channel with a non-optimized multiple access protocol can severely worsen the system performance in terms of delay/throughput (channel utilization). The current RFID multiple access (MAC) protocol emphasizes the role of *anti-collision* [1~7] which, generally speaking greatly increases latency for 'full reading' of all tags reliably, such as the default binary tree-walking approach for depth first search in Gen 1 [1-3]. Hence, several enhancements to the MAC have been proposed that rely on a key architectural notion: *round–based adaptation of the contention window within a framed TDMA format* which utilizes centralized feedback from the reader of the history of recent outcomes via *estimation of the number of backlogged tags* [8~15]. Note that such adaptation within the context of tree searching has already been proposed as in the *depth-first binary split algorithm*(DBSA) [4-7]; but this cannot take advantage of backlogged tag estimation because the collision resolution process is performed immediately based on slot-based (unlike the frame based proposed here) feedback from the reader. Besides, all tags involved in a collision have to undertake a complex state management task to keep track of their retransmission turns during the collision resolution process as in Figure 1-a. To overcome these drawbacks, we propose a novel anti-collision scheme called *breadth-first-search m-ary split algorithm* (BMSA) that is applied within a TDMA frame structure as in Figure 1-b. Based on our early work[16] that introduced the first notion of BMSA we further develop a *backlogged tag estimation* technique for a delay-throughput balanced performance.

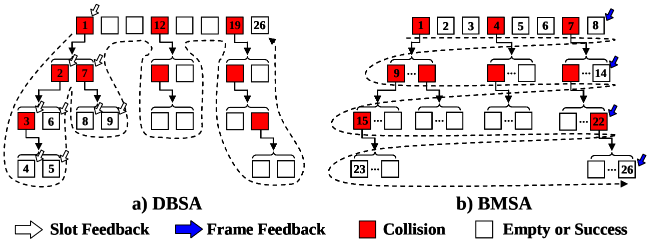


Figure 1 DBSA vs. BMSA

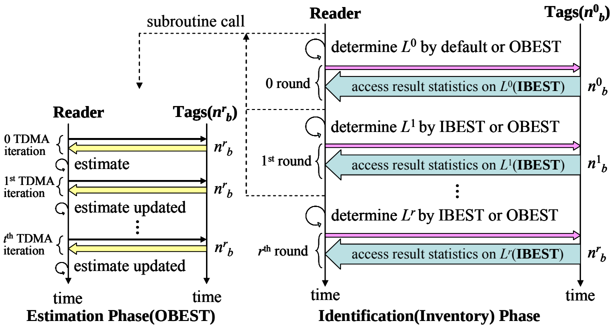


Figure 2 IBEST vs. OBEST

Since the maximum throughput of framed ALOHA scheme is achieved when a reader sets the frame size(L) as a function of the number of backlogged tags n_b, it is important to obtain as accurate an estimate $\hat{n}_b$ as feasible. We classify the backlogged tags estimation approaches into two classes:

1) *In-round Backlogged-tag ESTimation*(IBEST): $\hat{n}_b$ is obtained based on the access statistics in the previous round of the identification(ID read) phase itself, i.e. no separate querying for estimation is needed[8~13].
2) *Out-round Backlogged-tag ESTimation*(OBEST): To increase accuracy of $\hat{n}_b$, a *separate* query phase independent of the ID read is performed prior to starting the next identification round. This requires extra time overhead, as described in [14,15].

There clearly exists a tradeoff between estimator accuracy and the time needed to achieve it - accuracy in IBEST is coarse while a higher precision estimate can be attainable in OBEST at the expense of extra time overhead. As shown in Fig. 2, the backlogged tag estimator at identification round r is denoted by $\hat{n}_b^r$ which in turn is used to determine the frame size L^r. The existing standards based on dynamic framed-ALOHA leave details of the frame update mechanism to vendor implementation[1~3]. We define the *session delay*(Sd) to recognize the population of n_b^0 tags by the reader as follows:

$$Sd_{OBEST} = Ed_{OBEST} + Id_{ha}$$
$$Sd_{IBEST} = Id_{ca}$$

The session delay in OBEST(Sd_{OBEST}) is composed of the *estimation delay*(Ed_{OBEST}) plus the *identification delay with high accuracy*(Id_{ha}), whereas the session delay in IBEST(Sd_{IBEST}) is simply the *identification delay with coarse accuracy*(Id_{ca}). In Sd_{OBEST} however, the gain by way of time saving in the identification phase due to the high accuracy of $\hat{n}_b$, is far less than the overhead time required (Ed_{OBEST}) in the estimation phase for achieving that precision. In other words, $Id_{ca} > Id_{ha}$, but $Ed_{OBEST} \gg (Id_{ca} - Id_{ha})$ which we illustrate by an example in the following section. Hence BMSA uses a simple yet effective IBEST method to drive the dynamic updating of the various *splitting factors* (SF) m. For performance evaluation, we examine several existing backlogged tag estimation algorithms as a baseline, and show that the BMSA outperforms DBSA as well as framed-ALOHA scheme in terms of session delay and throughput via simulations.

II. RELATED WORK

2.1. In–round Backlogged-tag Estimation

To eliminate the instability problem of the fixed framed ALOHA, Schoute developed dynamic framed ALOHA that adjusts the frame length according to the estimate of the number of backlogged nodes [8]. Schoute developed *a posteriori* estimate for $\hat{n}_b$ equal to $2.39n_c$, where n_c is the observed number of collided slots in the frame. In [9,10] the *fixed collision rate*(FCR) method has been proposed based on two objectives: a) the maximum throughout is achieved when the number of access slots in a frame equals the number of competing nodes, and b) the slot collision rate at maximum throughput is maintained close to the constant $1\text{-}2/e \approx 0.26$. Thus the reader can adjust the frame size in each round in such a way so as to keep the frame collision rate at a constant level of 0.26.

2.2. Out–round Backlogged-tag Estimation

Recent work by Kodialam [15] presents an enhanced unified estimator based on more complete information; it uses both the number of empty and collided slots in a frame. It outperforms Schoute [8] and Vogt [12] in terms of estimator accuracy at the cost of requiring formidable time overhead(Ed_{OBEST}). This is mainly due to the fact that the unified estimator produces a lower variance estimate over a wider range of n_b than an estimator that only counts the number of collided slots. However, this improvement carries a price; according to the simulation result in Table 2 of [15], 7,498 slots are required to estimate 500 tags with 0.2% confidence interval using the Philips I-Code system (56-bit long tag ID and 56Kbps data rate). This translates to about 1,340 ms for estimation phase for a slot duration of 10 bits corresponding to 56 Kbps.

III. BREADTH-FIRST *m*-Ary SPLIT ALGORITHM

The conventional tree based anti-collision schemes[4~7] in Gen 1 RFID systems relies on depth-first binary splitting algorithm (DBSA) where a collided slot undergoes successive binary splitting till the collision is resolved as shown in Figure 1-a. DBSA requires complex state management by the tags involved in the collision to keep track of their retransmission access slots which depend on slot based feedback during a collision resolution interval. The approach also does not scale well for large number of tags because accurate estimation of the number of backlogged tags is not feasible due to slot-based feedback. In BMSA algorithm, the reader directs tree splitting within a *frame structure* based on sequential processing of slots as in Figure 1-b. Note that BMSA is different from framed ALOHA schemes since the m-slot sub ranges allocated for each collided slot are mutually exclusive in the next round. The BMSA operates as follows:

i) *Initialization*: Reader broadcasts the *select* command with mask bits and the initial frame size, L^0 for the first round.
ii) Each tag compares the mask bits to it's own ID and upon a match, chooses a slot number at random uniformly within L^0 for channel access.
iii) If any collision is detected during the r-th round, the reader issues the *query* command that contains the *access result map* with the outcome of all slots in the frame, and the splitting factor m for the next round to adjust the contention window via $L^{r+1} = mn_c^r$, where n_c^r is the number of collided slots in the r-th round.
iv) Each tag confirms the access result for it's slot in the previous round. If collided, the tag calculates the allowable m-slots whose indexes are given by $[S_{m\theta_i+1}, S_{m(\theta_i+1)}]$, where S_j is the j-th slot in the r+1 the round, $1 \leq m\theta_i+1 < m(\theta_i+1) \leq L^{r+1}$ and θ_i is the number of collided slots that occurred prior to the slot chosen by the tag in the r-th round obtained from the *access result map*. The tag chooses a slot in the above index range by either generating a random number or converting its ID to an m-ary digit, i.e., choosing the ($m\theta_i+1+q^s$)-th slot, where $0 \leq q^s \leq m-1$, and q^s is the s-th digit in the m-ary ID string.

Once a session starts iii) and iv) alternate until no further collision is detected in the frame. The important advantages in BMSA are two-fold: protocol efficiency is enhanced, whereby it can effectively read a larger tag population (n_b) for a desired quality metric (e.g. read latency). Secondly, from the perspective of algorithm implementation complexity, it replaces the more complex state management task imposed on a tag in DBSA with the simpler retransmission rule for all collided tags broadcast by the reader described earlier.

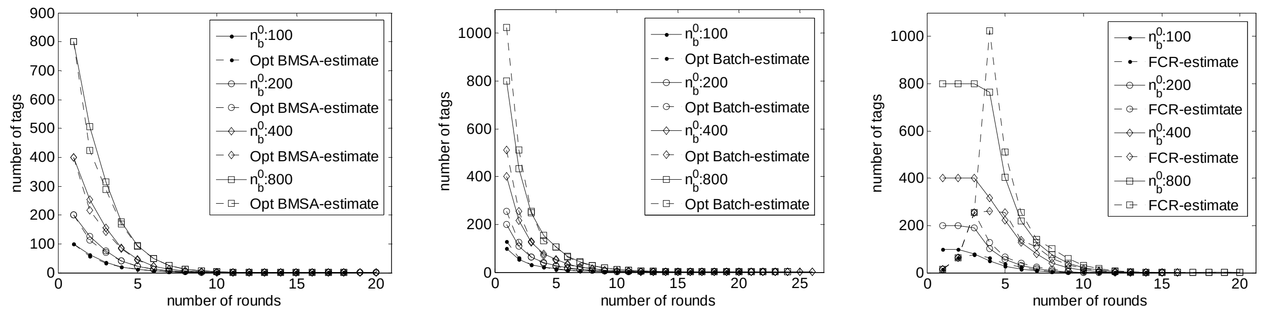
IV. PERFORMANCE EVALUATION

4.1. Background of Simulation

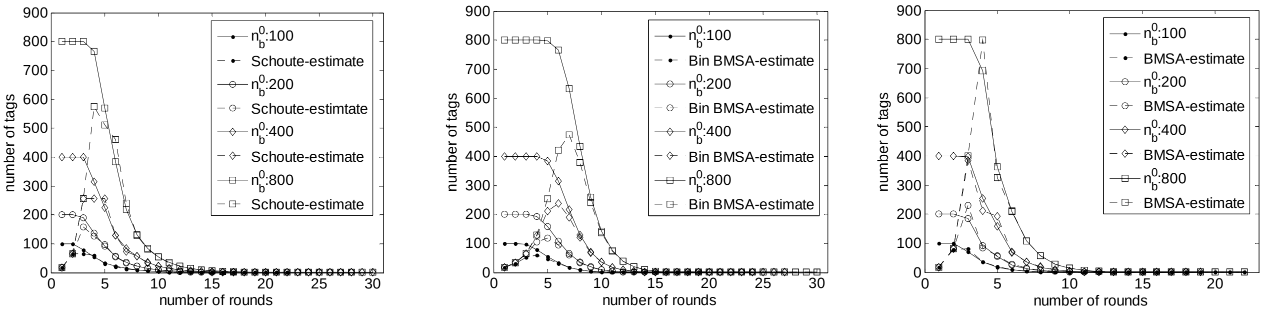
Our assumptions for the subsequent performance evaluation via simulation are as follows:

- The channel or air interface is assumed perfect, i.e. no packet losses due to background noise when a single transmission occurs.
- In the event of multiple simultaneous transmissions (collisions), all packets are implicitly lost, i.e. any *capture effect* is not considered.
- The tag that receives feedback of success removes itself from any further channel contention (one-time reading of ID).
- We use $L^0 = 16$ slots in all cases.

Generally there are two ways to implement the framed ALOHA protocol: *sequential* vs. *non-sequential(batch)* approaches to adapting the frame size. In sequential processing, an update for the frame length L for the next round is obtained whenever a prior criterion as an estimation scheme, i.e., the first p slots, where $1 \leq p < L$, are either all collided or all empty is met. If the criterion is not met within the p slots, the same L is used in the next round. In frame-oriented (batch) processing, the outcome from the entire current frame is observed(p=L) prior to estimating an updated L for the next round. In the *sequential* method, only some of the outstanding tags that choose the first p slots in the frame can retransmit during a round, while all n_b can have a retransmission chance during a round in the *batch* processing. The EPCglobal Class1 Generation2 (EPC C1G2) system supports both options. In our implementation of a *sequential* approach, the reader arbitrarily stops at an intermediate slot time before the end of current frame, given by 2^Q where Q is an integer in the range [0, 15], and issues a new L with an updated Q based on the outcomes till that point. Secondly it updates Q after the L time completely expires(*batch* approach). In our simulation, Q is updated as long as the first 2 slots in L are either empty or collided. In our *batch (non-sequential)* simulations, we consider the frame size determined by $L = 2^k$, where $k = 0, 1, 2, ..,$ i.e., $L = \min(| 2^{Q-1} - \hat{n}_b |, | 2^Q - \hat{n}_b |)$, where $2^{Q-1} \leq \hat{n}_b \leq 2^Q$ as in EPC C1G2.

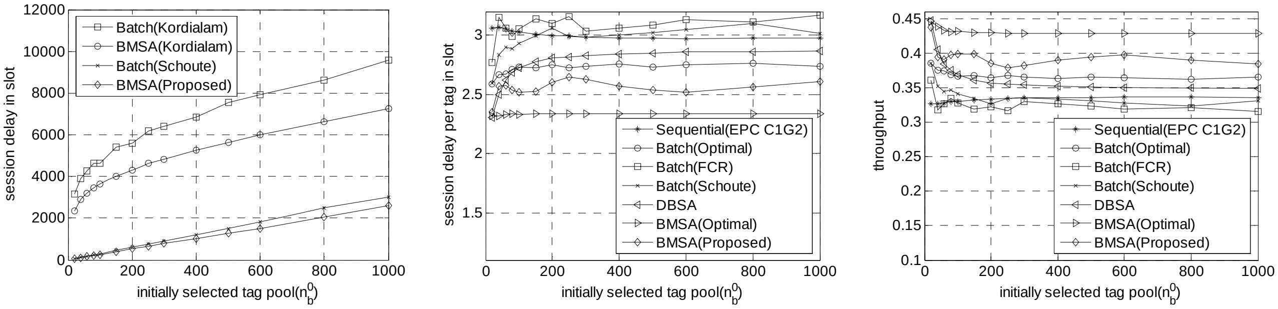


a) Dynamic BMSA(Optimal) b) Batch(Optimal, Kordialam) c) Batch(FCR)



d) Batch(Schoute) e) Binary BMSA f) Dynamic BMSA(Proposed)

Figure 3 Backlogged tags estimation



a) IBEST vs. OBEST delay b) IBEST delay c) IBEST throughput

Figure 4 Session delay and throughput

4.2. Backlogged Tag Estimation

Figure 3 shows the differences between n_b^r and $\hat{n}_b^r$ (=L^r) in *batch* framed ALOHA schemes based on FCR and Schoute, and BMSA, The reader starts with $L^0 = 16$ and obtains an estimate during a session for various size of n_b^0 such as 100, 200, 400 and 800. Figure 3-a, 3-b illustrate, respectively, the optimal cases for BMSA and *batch* framed Aloha, where the reader is assumed to know the exact n_b^r before starting each round. In optimal BMSA, m for the (r+1)-th round can be determined by $rnd\ (n_b^r / n_c^r)$, where $rnd(z)$ rounds z off to the nearest integer. With Kordialam scheme one can set L as close to the L in *batch* optimal case as possible. Figure 3-e is the case when m is fixed by 2 in BMSA. As seen in Figure 3-f, the proposed scheme among the estimation schemes achieves the smaller error in the shortest time regardless of the size n_b^0.

4.3. Session delay and Throughput

Figure 4-a shows the session delay for IBEST and OBEST schemes. The y-axis in Figure 4-b shows the mean number of required slots per tag until it is identified. For EPC C1G2 [1] parameters, 500 tags can be identified in (a) 430ms + (b) 500ms = 930ms using *sequential* method; where the first component is the time needed for slot reservation due to contention

resolution – according to the simulation in Figure 4-b, it needs about 1,500 slots. The same scenario with I-Code system in Section 2.2, with 1,500*16-bit slots (*RN*16 [1] for reservation) would need 430ms. The second component is the tag ID transmission period: 500*56-bit slots requires 500ms. Thus the *sequential* IBEST scheme incurs much lower(930ms) delay compared to 1,340 ms for estimating 500 tags in OBEST approach using the Kodialam estimator. Thus the cost of acquiring an accurate estimate of the number of backlogged tags L outweighs the potential benefits to reducing the tag identification delay. We see that the session delay for Schoute and FCR degrades as n_b^0 increases. This is because the Poisson assumption on those schemes cannot accommodate a large number of n_b well. In tree splitting algorithms, the binary slot splitting cannot effectively deal with a large number of n_b. Optimal BMSA shows the delay lower bound for BMSA. For frame adaptation, the proposed BMSA also makes use of the frame collision rate(n_c^r / L^r) by which the m for the (r+1)-th round varies from *binary* to *quinary*. As seen in the Figure 4-b and 4-c, the proposed scheme can increase overall performance by approximately 10%, 15%, and 18% when compared to DBSA, EPC C1G2, and *batch* framed ALOHA scheme, respectively, in terms of session delay and throughput. We note that the performance of proposed scheme is even better than the *batch* framed ALOHA optimal case.

V. CONCLUSION

The main drawbacks of the conventional tree splitting schemes such as *depth-first binary split algorithm* (DBSA) concern scalability for a large number of tags and resulting complex state management. To overcome these we proposed a *breadth-first m-ary split algorithm*(BMSA) within a TDMA frame structure. For backlogged tag estimation, we used a simple method that doesn't require any extra overhead. The contributions of the proposed scheme are mainly two-fold: 1) BMSA can effectively accommodate a large number of tags via controlling m dynamically during a session, and 2) because retransmission time is deterministically calculated based on reader feedback, the global synchronization process between the reader and tags is much simpler than that of DBSA resulting in lower protocol implementation complexity. The simulation results show that the proposed scheme reduced the identification delay about 10%, 15% and 18% when compared to DBSA, EPC C1G2, and *batch* framed ALOHA scheme, respectively.

REFERENCE

[1] EPCglobal Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1.0.9, 2005.

[2] "860-930 MHz Class I RFID Tag RF & Logical Communication Interface Specification v. 1.0.1", *MIT Auto-ID Center*, Nov. 2002.

[3] Philips Semiconductor, I-Code1 System Design Guide: Technical Report, May, 2002.

[4] C. Law, K. Lee, and K. Siu, "Efficient memoryless protocol for tag identification," Tech. Rep. of MIT Auto-ID Center, MIT-AUTOIDTR-003.pdf, 2000.

[5] M. Jacomet, A. Ehrsam, and U. Gehrig, "Contactless identification device with anticollision algorithm," *IEEE Conference on Circuits, System, Computers and Communications*, Greece, 1999.

[6] Jihoon Myung and Wonjun Lee, "Adaptive Binary Splitting: A RFID Tag Collision Arbitration Protocol for Tag Identification", *IEEE International Conference on Broadband Networks*, 2005.

[7] D.R. Hush and C. Wood, "Analysis of Tree Algorithm for RFID Arbitration", *IEEE International Symposium on Information Theory*, 1998.

[8] Frits C. Schoute, "Dynamic Frame Length ALOHA", *IEEE Transactions on Communications*, Vol.COM-31, No.4, April 1983.

[9] Chih-Peng Li, Tsui-Tsai Lin, "Fixed Collision Rate Back-off Algorithm fir Wireless Access Networks", *IEEE VTS* 2002.

[10] J. R. Cha and J.H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system", *IEEE International Conference on Parallel and Distributed Systems*, 2005.

[11] B. Zhen, M. Kobayashi and M. Shimizu, "Framed ALOHA for multiple RFID objects identification", *IEICE Transactions on Communications*, vol. E88-B(3), Mar 2005.

[12] H. Vogt, "Efficient Object Identification with Passive RFID Tags", *LNCS 2414*, Springer-Verlag, 2002.

[13] Christian Floerkemeier, "Transmission Control Scheme for Fast RFID Object Identification", *IEEE PERCOMW* 2006.

[14] A. Krohn, T.Zimmer, M. Beigl and C.Decker, "Collaborative Sensing in a Retail Store Using Synchronous Distributed Jam Signalling", Pervasive 2005, *LNCS* 3468, Springer-Verlag, May 2005.

[15] M. Kodialam, T. Nandagopal, "Fast and Reliable Estimation Schemes in RFID Systems", *ACM MobiCom* Sep. 2006.

[16] You-Chang Ko, et al. "Collision Reduction Random Access Using *m*-ary Split Algorithm in Wireless Access Network", *LNCS* 3510, Springer-Verlag, May, 2005.

---

[1] A tag in EPCglobal system generates a 16-bit random number(RN16) to backscatters it in contention mode.