

Adaptive Binary Splitting for Efficient RFID Tag Anti-Collision

Jihoon Myung, *Student Member, IEEE*, Wonjun Lee, *Senior Member, IEEE*, and Jaideep Srivastava, *Fellow, IEEE*

Abstract—Tag collision arbitration for passive RFID tags is a significant issue for fast tag identification. This letter presents a novel tag anti-collision scheme called *Adaptive Binary Splitting (ABS)*. For reducing collisions, *ABS* assigns distinct timeslots to tags by using information obtained from the last identification process. Our performance evaluation shows that *ABS* outperforms other tree based tag anti-collision protocols.

Index Terms—Collision resolution, RFID, tag anti-collision, tag identification.

I. INTRODUCTION

IN RFID, tag-to-reader signals transmitted simultaneously cause collision called tag collision, which hinders fast tag identification. Since low-functional passive tags can neither detect collisions nor figure out neighboring tags, tag anti-collision protocols for passive RFID tags are important for tag identification.

Tag anti-collision protocols are separated into aloha-based and tree-based. Aloha-based protocols such as aloha, slotted aloha, and frame aloha [1], [2], [3] reduce the occurrence probability of tag collisions since tags transmit at distinct times. However, aloha-based protocols cannot completely prevent collisions and hence they have ‘tag starvation problem’ that a specific tag may not be identified for a long time. In tree-based protocols which are based on the collision resolution algorithm studied in [4], tags, which transmit at the same time, form a set. When a set causes collision, the mechanisms split it into two subsets and attempt to recognize two subsets in turn. The binary tree protocol, which uses random numbers for splitting, is adopted as the standard for RFID anti-collision in ISO/IEC 18000 Part 6 [3]. The query tree protocol [5] splits a set of tags by the reader’s queries. Although tree-based protocols do not cause tag starvation, they have relatively long identification delay due to the splitting procedure starting from one set including all tags.

We propose an *adaptive binary splitting (ABS)* scheme, which is an improvement on the binary tree protocol. For decreasing collisions, *ABS* starts the splitting procedure from several sets of tags by using information obtained from the

last identification process. We presented the same approach on the query tree protocol in [6]. *ABS* facilitates tag identification with a short delay, while recognizing all tags.

II. THE PROPOSED

ADAPTIVE BINARY SPLITTING SCHEME

We assume that the channel is slotted and a reader performs identification processes repeatedly for object tracking and monitoring. In a timeslot, tags transmit ID to the reader and the reader then transmits a feedback. A feedback informs all tags of the result of the reader’s reception, namely, *idle* (no tag signal), *readable* (only one tag signal) or *collision* (multiple signals).

Tag t_i decides the timeslot for the transmission with a *progressed-slot counter*, P_C , and an *allocated-slot counter*, $A_C(i)$. P_C means the number of tags recognized by the reader in the ongoing process. At the beginning of the process, P_C is initialized with 0. $A_C(i)$ signifies the timeslot for t_i ’s transmission, i.e., t_i transmits when $P_C = A_C(i)$. The tags, which have the same value of the allocated-slot counter, form a set. Tag collision occurs if a set include multiple tags. According to the feedback, tags act as follows.

- *Readable*: Tags add 1 to P_C .
- *Idle*: Tag t_i decreases $A_C(i)$ by 1 to pull the schedule of the transmission if $P_C < A_C(i)$.
- *Collision*: To split a set of tags, tag t_i generates a random binary number and adds it to $A_C(i)$ if $P_C = A_C(i)$. Note that colliding tag t_i has $A_C(i)$ equal to P_C . Since P_C is not changed, the first subset (tags which generate 0) retransmits at the following timeslot and the second subset (tags which generate 1) retransmits after the first subset is recognized. To prevent the second subset and another set of tags, which have already had the allocated-slot counter of $A_C(i) + 1$, from integrating, tag t_j adds 1 to $A_C(j)$ if $P_C < A_C(j)$.

At the end of the process, every tag has a unique allocated-slot counter. Therefore, maintaining the allocated-slot counter at the boundary of two consecutive processes enables the splitting procedure to start from multiple sets of tags. Fig. 1 shows tag identification of *ABS* when there are three tags and all the tags have the allocated-slot counter of 0 before tag identification.

For terminating the identification process at once after identifying all tags, the reader acts as the tag which has the largest allocated-slot counter. Reader r concludes all tags to have been recognized and terminates the process if $P_C = R_C(r)$ ($R_C(r)$ is reader r ’s allocated-slot counter).

Manuscript received September 28, 2005. The associate editor coordinating the review of this letter and approving it for publication was Prof. Friedrich Jondral. This work is jointly supported by a grant from SK Telecom, Korea [Project No. KU-R040572] and by a grant by KOSEF [Grant No. R01-2005-000-10267-0].

W. Lee and J. Myung are with the Dept. of Computer Science and Engineering, Korea University, Seoul, Korea (e-mail: wlee@korea.ac.kr).

J. Srivastava is with the Dept. of Computer Science and Engineering, University of Minnesota, Minneapolis, USA.

Digital Object Identifier 10.1109/LCOMM.2006.03031.

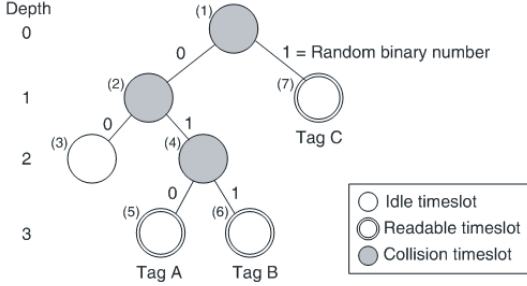


Fig. 1. Tag identification of ABS; a circle indicates a set of tags, and a number in parentheses denotes the number of the timeslot in which the operation (reading tag ID or changing the allocated-slot number) is executed.

In an environment of multiple readers, a tag can also be recognized by reader r with its allocated-slot counter decided by another reader. When an identification process starts, reader r provides tags with $R_C(r)$. Tag t_i changes $A_C(i)$ into a random number from 0 to $R_C(r) - 1$ if $A_C(i) \geq R_C(r)$.

III. ALGORITHMIC PERFORMANCE ANALYSIS

We analyze the average identification delay of ABS through the derivation of delay at the binary tree protocol.

Definition 1: Let $S_{r,i}$ be the set of all the tags recognized by reader r in the i th identification process of reader r . Identification delay of recognizing $S_{r,i}$, $d_{total}(S_{r,i})$, is

$$d_{total}(S_{r,i}) = \sum_{x=1}^{T(S_{r,i})} (d_{reader} + d_{tag}) \approx T(S_{r,i}) \cdot d_{timeslot} \quad (1)$$

where $T(S_{r,i})$ is the number of timeslots required for recognizing $S_{r,i}$, d_{reader} is the delay of delivering the reader's feedback, d_{tag} is the delay of delivering tag ID, and $d_{timeslot}$ is the average delay of the timeslot. Identification delay is determined by $T(S_{r,i})$.

To consider tag mobility, we categorize tags. For the i th identification process of reader r , *staying tag* t_s is $t_s \in S_{r,i} \cap S_{r,i+1}$, *arriving tag* t_a is $t_a \in S_{r,i+1} - S_{r,i}$, and *leaving tag* t_l is $t_l \in S_{r,i} - S_{r,i+1}$. Let $S_{r,i} = \{t_1, t_2, \dots, t_n\}$, B_α be the set of arriving tags, $\{t_{n+1}, t_{n+2}, \dots, t_{n+\alpha}\}$, and B_l be the set of leaving tags, $\{t_{f(1)}, t_{f(2)}, \dots, t_{f(\beta)}\}$ ($1 \leq f(x) \leq n$).

Theorem 1: Let $T_{ABS}(S_{r,i+1} | S_{r,i})$ be the number of timeslots consumed by ABS recognizing $S_{r,i+1}$ after recognizing $S_{r,i}$. When $S_{r,i+1} = S_{r,i}$,

$$T_{ABS}(S_{r,i+1} | S_{r,i}) = n \quad (2)$$

Proof: During the i th identification process, ABS makes n sets and each set includes only one tag. In the $i+1$ th process, tags transmit ID in increasing order of the allocated-slot counter and there exist only readable timeslots. \square

Theorem 2: When $S_{r,i+1} = S_{r,i} - B_l$,

$$T_{ABS}(S_{r,i+1} | S_{r,i}) = T_{ABS}(S_{r,i} | S_{r,i}) = n \quad (3)$$

Proof: Before the $i+1$ th identification process, a set of a leaving tag becomes empty. After some tags reach out of the reader's range, tag identification at ABS generates idle timeslots. To eliminate idle timeslots, ABS requires one process. Therefore, the following identification process has idle timeslots caused by leaving tags. \square

Lemma 1: Let $C_{binary}(n)$ be the number of collisions caused by the binary tree protocol recognizing n tags. The

| Counter | P_C | A_C | | | R_C |
|-----------------|-------|-------|-------|-------|--------|
| | | Tag A | Tag B | Tag C | Reader |
| 1 (Collision) | 0 | 0 | 0 | 0 | 1 |
| 2 (Collision) | 0 | 0 | 0 | 1 | 2 |
| 3 (No response) | 0 | 1 | 1 | 2 | 3 |
| 4 (Collision) | 0 | 0 | 0 | 1 | 2 |
| 5 (Identified) | 0 | 0 | 1 | 2 | 3 |
| 6 (Identified) | 1 | 0 | 1 | 2 | 3 |
| 7 (Identified) | 2 | 0 | 1 | 2 | 3 |

total number of timeslots consumed by the binary tree protocol for n tags, $T_{binary}(n)$, is

$$T_{binary}(n) = 2C_{binary}(n) + 1 \quad (4)$$

Proof: Tag identification of the binary tree protocol can be represented by a full binary tree because it splits only the set of colliding tags into two subsets. Therefore, all the intermediate nodes in the tree correspond to collisions and all the leaf nodes correspond to either idle timeslots or readable timeslots. \square

Lemma 2: For any n ,

$$T_{binary}(n) = 1 + 2 \sum_{k=0}^{\infty} \{2^k - p(k)^{n-1} (2^k + n - 1)\} \quad (5)$$

where $p(k) = 1 - 2^{-k}$.

Proof: Let $I_{binary}(n, k)$, $R_{binary}(n, k)$ and $C_{binary}(n, k)$ denote the number of idle timeslots, readable timeslots and collisions, respectively, in the depth k of the tree produced by the binary tree protocol recognizing n tags. $C_{binary}(n)$ is

$$C_{binary}(n) = \sum_{k=0}^{\infty} C_{binary}(n, k) \quad (6)$$

Let $p(k) = 1 - 2^{-k}$ denote the probability that a tag does not transmit in a timeslot of the depth k . The m th timeslot of the depth k , $s_{k,m}$ ($1 \leq m \leq 2^k$), is the idle timeslot if n tags select timeslots except $s_{k,m}$ in the depth k .

$$I_{binary}(n, k) = 2^k p(k)^n \quad (7)$$

$s_{k,m}$ is the readable timeslot if only a tag selects $s_{k,m}$ and other tags select timeslots except $s_{k,m}$ in the depth k .

$$R_{binary}(n, k) = np(k)^{n-1} \quad (8)$$

Therefore,

$$\begin{aligned} C_{binary}(n, k) &= 2^k - I_{binary}(n, k) - R_{binary}(n, k) \\ &= 2^k \{1 - p(k)^k - n2^{-k} p(k)^{n-1}\} \end{aligned} \quad (9)$$

Lemma is proved by (4), (6) and (9). \square

Theorem 3: When $S_{r,i+1} = S_{r,i} + B_a$,

$$\begin{aligned} T_{ABS}(S_{r,i+1} | S_{r,i}) &= n \\ &+ 2n \sum_{k=0}^{\infty} \left\{ 2^k - p(k)^{\alpha/n} \left(2^k + \frac{\alpha}{n} \right) \right\} \end{aligned} \quad (10)$$

Proof: After the i th process, under ABS, there exist n sets, each set includes only one staying tag, and $R_C(r) =$

n . Without loss of generality, we assume that allocated-slot counters of arriving tags are uniformly distributed over the interval $[0, n - 1]$. Therefore, $\alpha n^{-1} + 1$ tags are assigned to each set at the beginning of the i th process.

$$T_{ABS}(S_{r,i+1} | S_{r,i}) = nT_{binary}(1 + \alpha n^{-1}) \quad (11)$$

Theorem is proved by lemma 2. \square

Theorem 4: When $S_{r,i+1} = S_{r,i} - B_l + B_a$,

$$\begin{aligned} T_{ABS}(S_{r,i+1} | S_{r,i}) &= n \\ &+ 2n \sum_{k=0}^{\infty} \{2^k - p(k)^{\alpha/n} (2^k + \alpha n^{-1})\} \\ &- 2\alpha\beta n^{-1} \sum_{k=0}^{\infty} \{p(k)^{\alpha/n} (2^k - 1)^{-1}\} \quad (12) \end{aligned}$$

Proof: After the i th process, β sets become empty by leaving tags and then α arriving tags are assigned to n sets uniformly.

$$\begin{aligned} T_{ABS}(S_{r,i+1} | S_{r,i}) &= T_{ABS}(S_{r,i} - B_l + B_a | S_{r,i}) \\ &= (n - \beta)T_{binary}(1 + \alpha n^{-1}) + \beta T_{binary}(\alpha n^{-1}) \quad (13) \end{aligned}$$

\square

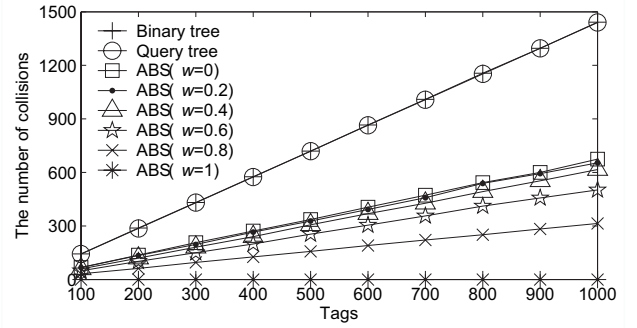
IV. SIMULATION RESULTS AND PERFORMANCE COMPARISON

We evaluate identification delay and tag communication overhead at *ABS* compared to the binary tree protocol and the query tree protocol. Identification delay is defined as the number of timeslots required for all tags. The fast identification is the most significant factor in the tree-based protocols because they do not cause tag starvation problem. Tag communication overhead is defined as the average number of bits transmitted by a tag for identification. This influences the amount of power consumption and it must be low due to the lack of the power source of tags.

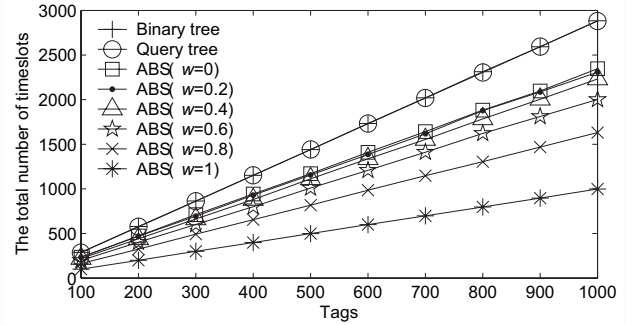
Fig. 2 shows the simulation results obtained by changing the number of tags and the ratio of staying tags to recognized tags. Let n denote the number of tags recognized in the last process. Let γ , α and β be the number of staying tags, arriving tags, and leaving tags, respectively. For a given value w , we set $\gamma/n = w$ and $\alpha/n = \beta/n = 1 - w$. *ABS* has the fastest identification and the smallest overhead of tags. Restraint on collisions enables fast identification at *ABS*. Tag communication overhead at *ABS* is not affected by the number of tags. When $w=1$, *ABS* shows the best performance and recognizes tags without both collisions and idle timeslots.

V. CONCLUSION

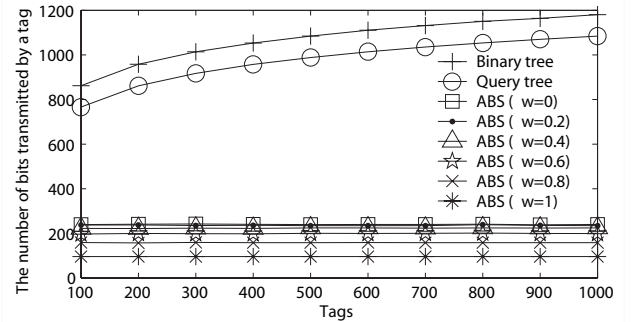
Tag collision is a major factor in deferring tag identification of RFID systems. We develop a novel and enhanced tree-based tag anti-collision scheme to reduce collisions by exploiting information obtained from the last identification process. A simulation based evaluation shows that *ABS* significantly reduces delay and tag communication overhead for the tag reading process.



(a) Collisions caused by tag identification



(b) Identification delay of recognizing all tags



(c) Tag communication overhead

Fig. 2. Performance comparison with varying the number of tags and the ratio of staying tags (w).

REFERENCES

- [1] J. Zhai and G. Wang, "An anti-collision algorithm using two-functioned estimation for RFID tags," in *Proc. ICCSA'05*, LNCS 3483, pp. 702-711, May 2005.
- [2] "EPCTM radio-frequency identification protocols class-1 generation-2 UHF RFID protocol for communications at 860MHz-960MHz Version 1.0.8," EPCglobal, Dec. 2004.
- [3] "Information technology automatic identification and data capture techniques - radio frequency identification for item management air interface - part 6: parameters for air interface communications at 860-960 MHz," ISO/IEC FDIS 18000-6, Nov. 2003.
- [4] J. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inform. Theory*, vol. 25, pp. 505-515, Sept. 1979.
- [5] F. Zhou *et al.*, "Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems," in *Proc. ACM ISLPED'04*, pp. 357-362.
- [6] J. Myung and W. Lee, "An adaptive memoryless tag anti-collision protocol for RFID networks," *IEEE INFOCOM'05*, Poster Session, Mar. 2005.