Tag Anti-collision Algorithm for RFID Systems with Minimum Overhead Information in the Identification Process

Usama S. MOHAMMED, Mostafa SALAH

Dept. of Electrical Engineering, University of Assuit, Assiut 71516, Egypt

Usama@aun.edu.eg, engmsam1@gmail.com

Abstract. *This paper describes a new tree based anti-collision algorithm for Radio Frequency Identification (RFID) systems. The proposed technique is based on fast parallel binary splitting (FPBS) technique. It follows a new identification path through the binary tree. The main advantage of the proposed protocol is the simple dialog between the reader and tags. It needs only one bit tag response followed by one bit reader reply (one-to-one bit dialog). The one bit reader response represents the collision report (1: collision; 0: no collision) of the tags' one bit message. The tag achieves self transmission control by dynamically updating its relative replying order due to the received collision report. The proposed algorithm minimizes the overhead transmitted bits per one tag identification. In the collision state, tags do modify their next replying order in the next bit level. Performed computer simulations have shown that the collision recovery scheme is very fast and simple even with the successive reading process. Moreover, the proposed algorithm outperforms most of the recent techniques in most cases.*

Keywords

Passive RFID tag, anti collision protocol, binary tree protocol, Aloha-based protocols.

1. Introduction

RFID systems consist of networked electromagnetic readers and tags, where the readers try to identify the tags as quickly as possible via wireless communications. However, since the readers or the tags communicate over the shared wireless channel, the collision problem occurs in signal transmission of the readers or the tags, which leads to slow identification. Thus, it is a key issue to develop an efficient anti-collision protocol reducing collisions so as to identify all tags in the interrogation zone. Collisions are divided into reader collisions and tag collisions. Reader collision problems arise when multiple readers are simultaneously used. The other, most important, collision problem (approached in this paper) is the tag collision that occurs when several tags try to answer to a reader query at the same time. Passive tags take its power from reader RF signal, and use load modulation by reflecting energy from the reader for setting up communication to the reader. The tag may be designed to communicate in half duplex or full duplex mode.

2. Related Work

In RFID system, there are two approaches of tag collision resolution scheme: (1) Probabilistic algorithm which is based on ALOHA. In generally, ALOHA based protocols cannot perfectly prevent tag collisions because of the probabilistic procedure that allows random medium access in the identification process [1]. (2) Deterministic algorithm (tree based protocols) which detects collided bits and splits disjoint subsets of tags. The reader in the query tree (QT) protocols sends a query containing a prefix having a length of 1 to n bits. The tags whose prefixes match with the bits sent by the reader replies back with their tag ID. The reader asks the tags to answer if their ID matches the given prefix [2]. There are different schemes of the basic query tree protocols as in [3] - [8] for reducing the exchanged overhead information between the reader and tags, and to have shorter identification time. In [3], it works by reversing the IDs of the tags and then applying the query tree (QT) protocol. Because it is effective to classify the suffix first, if the bit string has a consecutive or identical prefix. In [4], it performs the query tree anti-collision algorithm on a smaller length of 16-bit randomly generated (virtual or temporary ID) numbers as shortcut representation of the original ID with 96 bit length. It tries to identify tags through identifying their randomly generated numbers first, then requests the tag that owns that virtual ID to start transmitting its full length real ID. In [5], bi-slotted query tree algorithm (BSQTA) and bi-slotted collision tracking tree algorithm (BSCTTA) are presented. For fast tag identification, BSQTA and BSCTTA use time divided responses depending on whether the collided bit is ‘0’ or ‘1’ at each tag ID. The reader sends n-1 length inquiring bits (prefix) once to tags instead of sending the same prefix twice with a different last bit. It reduces both prefix overhead and iteration overhead.

In [6] and [7], the ID of the tag is divided into several sections, and each section contains special sequences. Readers can get every ID by identifying each section one by one. Each section consists of two bits, which were represented according to Manchester Coding.

There are different schemes of counter based protocols reported in [9]-[14]. In [9], the tag uses its internal counter to determine when it changes the state from quiet state to active state. The reader must send a query bit of the last collision bit position to inform the tags with the last stop bit position. In [10], the basic search criterion is the depth-first search (DFS) algorithm. However, the reader command frame is long, and it must contain the bit position of the most recently occurred collision. In [11] and [12], adaptive binary splitting (ABS) uses counter to reach the goal of anti-collision, but the splitting of sets depends on the generation of the random binary number {0, 1}. So, it cannot achieve the best splitting result. The probability of occurrence of 1 or 0 is not 50%. At any moment, there won't be any splitting result, and may cause the next time-slot to be idle time-slot or collided time-slot. Consumption of time-slots and longer time-slot are the main drawback of the ABS protocol.

A new idea in [14] is introduced to reduce the probability of collision efficiently and to make fast identification. It reduces the length of the time slot by truncating unnecessary data bits to minimize the receiving time. The reader does not need to receive any data after receiving the first collided bit. A feedback message is sent by the reader to inform the tags about the type of a time-slot (collision, idle, readable, or Multiple-readable). Feedbacks are just like instructions and include operating code and some other information. The operating code in 3 bits is used. But in the other side, it has long reader instructions and the reader must inform the tag the position of collided bit in the collision code, for example (Opcode||Query String= 000||0010).

3. Proposed Parallel Binary Splitting Protocol

3.1 Parallel Binary Splitting Identification Path

The main objective of the PBS is to simplify the dialog between the reader and tags during the reading process (the reader extracts the tags' IDs). In the new PBS path, the main advantage is the parallel processing technique. Fig. 1 shows the difference between the new parallel binary splitting (PBS) and the traditional depth first search (DFS) technique which starts each new splitting path until ending at one tree leaf. The reader listens to tags' one bit at each node in the same bit level before going to the next discovered level as shown in Fig. 1b. PBS does not need query of last collision bit position to restart another path such as in DFS (Fig. 1a).

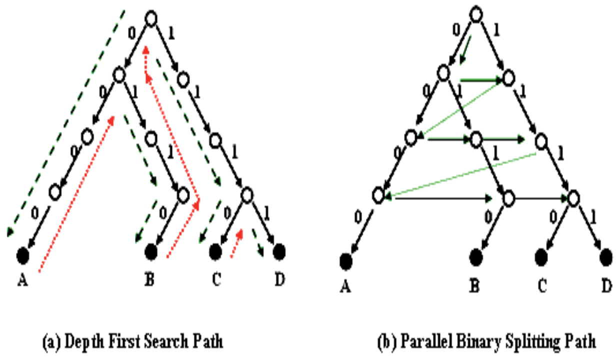

Fig. 1. The difference between the depth first search path and the parallel splitting identification path.

3.2 Reader Operation

Our protocol is based on remapping the discovered binary tree configuration after each tag and reader one bit dialog. The reader continually updates the pretended binary tree of existing tags according to tags reply step by step. The reader explores the future nodes branched from each already discovered node (subgroup) of the previous splitting level, by examining (scanning) the previously discovered paths in the past binary splitting level. Each subgroup replies the reader interrogation by sending its next bit in its assigned order. It is mainly depending on exchanging one bit sequentially between the tags and the reader (one-to-one bit dialog). The one bit reader message provides the tags with a comment conveying information about the collision state of last tags' transmitted one bit message.

Reader detects the state of the last tag reply, of the scanned subgroup, and sends the collision report:

(Detecting Collision: send 1, No Collision : send 0).

Depending on the collision report, received from the reader, each tag continually updates its recognized relative position in the assumed virtual replying queue with respect to surrounding tags. Hence, the tag determines its future replying in the next level and setting up self transmission control.

3.3 Tag Operation

By knowing the state of the last tags reply (collision or no collision) from reader report, each tag continually changes its relative position in virtual replying queue. Each tag in the reply queue waits its replying order to send the next level ID-bit. Each tag knows its current replying-order, and remains in its order in the replying queue until detecting a tag collision. The tags will classify themselves in a new subgroup according to the reader one bit signal only. Each subgroup can reply the next marked bit of its ID during its allocated order. The collision state means the insertion of an additional node in that position to the already discovered nodes. The collision state does not require retransmission or stopping tags transmission, but it leads to reordering tags relative replying orders for the next bit level. The number of discovered subgroups equals to the number of collision nodes plus one.

Tags use simple logic operation based on two counters and two registers, to achieve self transmission control and dynamically updated replying orders. The registers and counters are defined as follows:

1. Current Path Register (CPR): is used to store the current number of paths (binary branches) in that bit level. It contains the number of checked node in that binary level.
2. Next Paths Counter (NPC): is used to store the total number of continually discovered paths. It will be incremented when the reader reports a collision tag reply. Any collision means an increasing in the binary braches by one. *CPR will be loaded by NPC content at the end of each bit level splitting.*
3. Current Order Register (COR): is used to store the tag replying order with respect to the current number of paths in CPR.
4. Next Order Counter (NOC): is used to track the change in the tag replying order. It will be incremented when a new branch of lower order appears in the binary tree. *COR will be loaded by NOC content at the end of each bit level splitting.*

The tag operation can be described as follows:

1. Receiving the reader starting command.
2. Initially, each tag starts by thinking that it is the only tag in the reader range and resets the counters and the registers.
3. One-bit tag response in its replying order, one-bit reader report will follow that.
4. Scanning the previously discovered paths (nodes) in the past splitting level. Each subgroup sends the current marked bit in the current bit level.
5. Tags know the collision state from the reader report at each node. Tags modify its control counters as follow:

a- IF "No Collision": THEN no change in its order and the total number of paths.

b- IF "Collision" : THEN
*increment the total number paths (increment NPC).
* "IF the tag is not scanned in the current bit level (i.e. it is waiting its replying order)"
OR
"IF it is the tag replying order and participating in the current tag collision by sending its marked bit which is one" THEN:
*incrementing its replying order in the next splitting level (increment NOC).

6. Registers (COR, CPR) are updated by the contents of the corresponding counters (NOC, NPC) at the end of each bit level, to start the next bit level with the modified orders. The changing in the paths and the orders will not be considered until the start of the next splitting binary level. (During the scan level, the next orders are estimated from the contents of the two registers, not the counters).
7. Scanning the next splitting level and repeat the process starting from step 4 until completing "*n*" level of the ID length.

3.4 Demonstration Example

Assuming that, there are four tags to be identified, as an example, {A, B, C, D} = {0000, 0110, 1110, 1111}. Fig. 2 and Tab. 1 describe in details the process of node exploration with the updated orders. The assigned tag order is showed step by step. It consumes one bit for each node for tag response and one bit for reader to report the type of each node (collision or no collision).

*The overall bit transferred between tags and reader = 9 bit tag response + 9 bit reader reply = **<u>18 bit.</u>**

*In general, the number of exchanged bits according PBS equals to double of the number of binary tree node of existing tags except the leaves tags.

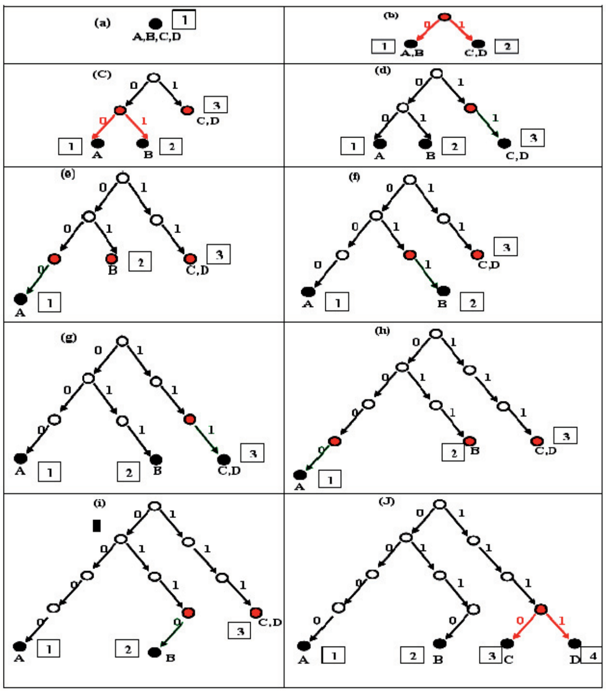

Fig. 2. Parallel splitting scan for path exploration.

step	Reader Reply	Reply of TAG (After Reader Response)				NOC (Next Order Counter)			
		A (0000)	B (0110)	C (1110)	D (1111)	A	B	C	D
a	Start	0	0	1	1	1	1	1	1
b	1	0	1	---	---	1	1	2	2
c	1	---	---	1	1	1	2	3	3
d	0	0	---	---	---	1	2	3	3
e	0	---	1	---	---	1	2	3	3
f	0	---	---	1	1	1	2	3	3
g	0	0	---	---	---	1	2	3	3
h	0	---	0	---	---	1	2	3	3
i	0	---	---	0	1	1	2	3	3
J	1	---	---	---	---	1	2	3	4

Tab. 1. The anti-collision process of the proposed algorithm (--: silent)

4. Modified PBS Protocol

In this section, the modification of the PBS protocol is suggested. It is fast parallel binary splitting and will be denoted as FPBS. Although, the PBS path minimizes the dialog between the reader and tags to only one bit tag response followed by one bit reader reply, but the reader one bit response for each one bit of tags' reply is considered large overhead information that must be reduced. The one bit reader response is used for reporting the collision state (1 for saying collision, 0 for no collision). The FPBS further minimizes the exchanged bits by confining the need of sending reader report to the collision condition only.

4.1 Collision Tracking Assumption

The reader can analyze the response of tags clearly and detect collision by using the Manchester code characteristics. The reader checks whether a collision occurs or not in each bit on the received sequences. The reader does not need to receive any data after receiving the first collided bit.

The reader can truncate unnecessary data bits to reduce the receiving time. ***The reader transmits an ACK signal to stop tags transmission if there is a collision.*** It is a practical assumption and can be found in [5], [13], and [14]. This assumption implies an increased cost of full duplex communication tag ability.

4.2 FPBS Operation

According to the PBS path, the "No Collision" state does not require any modifications in the assigned tags relative orders. It is only the collision state that requires changing in the relative orders and the control counters. It will be assumed that the "No Collision" is the default state. The tags send their marked bit in its previously assigned orders until receiving reader acknowledge (notifying) of collision state. If a data collision takes place, the reader sends acknowledge, else, no reader action is considered. By receiving the collision acknowledge from the reader, all tags modify their next replying orders in the next bit level, without any change in the current replying orders.

The new self assigned replying orders will be considered in the next bit level. As soon as receiving reader's collision ACK, all tags stop transmission and change its control counters. Fig. 3 shows the time diagram of both PBS and FPBS. The collision bit does not need to be transmitted again. It leads to modifications in the internal counters of all tags involved in the identification session. As shown in Fig. 3b, the tags continue in transmitting information without reader interruption until the reader detects collided bit. Hence the reader sends ACK to inform the tags that it is the time to change the orders due to the insertion of a new subgroup.

Fig. 3. Time diagram: (a) PBS, (b) FPBS.

5. Performance Analysis

In this section, the performance of FPBS algorithm will be discussed. Fig. 4 shows the exchanged bit stream between the reader and the four tags to be identified, as an example, {A, B, C, D} = {0000, 0110, 1110, 1111} by PBS and FPBS protocols.

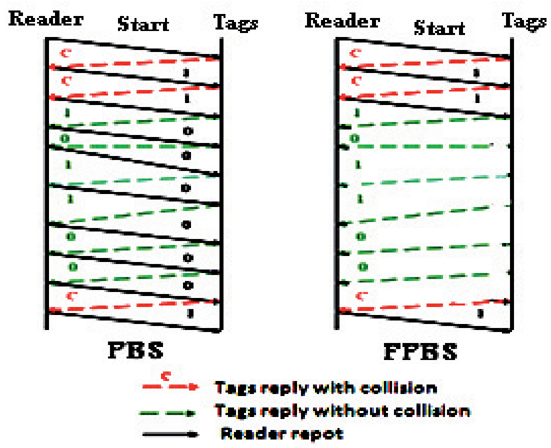

Fig. 4. Transmitted bit stream between the reader and tags for parallel splitting algorithm by PBS & FPBS.

In the EAA algorithm [13], the total number of feedback bits and the reader response is 19 and 11; respectively. It uses 30 bits to identify the four tags in our example. (19+11=30 bit).

In the proposed PBS algorithm, it consumes one bit for each node (tag response), and one bit for reader to report the state (collision or no collision). The tree has 9 nodes (except the last nodes). The number of transmitted bits by the tag equals the transmitted bits by the reader and equals 9 bit. Then, the overall bit transferred between = 18 bits.

In the proposed algorithm (FPBS), the number of transmitted bits by tag is 9 bits and the number of collision nodes is 3 nodes which equals the number of transmitted bits by the reader.

Then, the identification time can be estimated as the time of transferring 9+3=12bits.

Recently, in NEAA [14], another example of identifying seven tags A, B, C, D, E, F, and G in the interrogating zone is proposed. Their tag IDs are “0000”, “0001”, “0010”, “0110”, “1001”, “1010”, and “1110”, respectively as shown in Fig. 5.

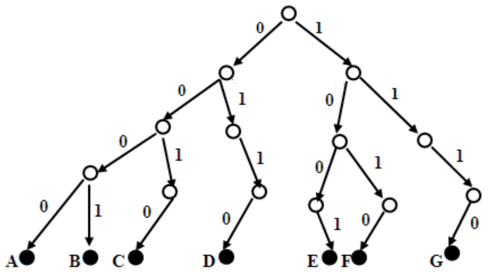

Fig. 5. The diagram of the binary tree of the current 7 tags.

The total number of feedback bits and the total number of response bits are 23 and 19, respectively according to **NEAA** protocol [14]. The total number of transmitted bits between tags and reader are **23+19=42 bits**.

However, in **FPBS**, it needs 13 bits as tags response, and 6 bits as reader collision ACK (note that: number of tags = number of collision nodes + 1). Hence, the total exchanged bits are equal to **13+6=19 bits.**

6. The Upper Limit (Bound) of the Bit-Exchanged between the Reader and Tags

It is important to note the following:

*The number of transmitted bits from the tags equals to the number of the nodes in the binary tree.

*The number of transmitted bits from the reader equals to the number of the collisions in the binary tree.

*The number of tags in the binary tree equals to the number of collisions plus one. Hence, the number of the reader transmitted bits (collision reports) almost equals to the number of identified tags.

***The total number of exchanged bits = number of tree nodes (i.e. except leaves) + number of tags (i.e. tree leaves) = number of the binary tree nodes.**

Hence, it is easy to compute the consumed number of bits according our FPBS protocol by counting the number of tree nodes of the existing tags.

The upper limit (L) is considered the maximum number of exchanged bits between tags and its reader (interrogator). It must be less than the number of existing tags multiplied by the length of the Tag ID.

For example: if you have at random **500 tag** (with the tag ID length = 96 bit). Then,

- The number of the exchanged bits must be less than 500*96 =48000 bits. However, under the bit rate of 80 kb/s, we need **0.6 second** to complete the identification process in its worst case.
- The number of the binary tree nodes is 43532 bits. It has 500 collisions. Hence, the total exchanged bits are equal to 44032 bits (the total number of nodes including the leaves nodes). It can be identified using the FPBS algorithm in **0.55 second** under the bit rate 80 kb/s.

7. Simulation Results

In this section, the performance of the proposed FPBS algorithm is performed in “Matlab”. Fig. 6 shows comparison among the PBS, the FPBS algorithm and the Dynamic Bit Arbitration (DBA) [8] algorithm when the tag has ID length of 32 bit and the number of tags is increased from 50 to 500 tags. Fig. 7 shows the average number of identified tags per second under the following simulation conditions:

In the field of the reader, the number of tags is increased from 2 to 512 and the length of the tag IDs is 96 bits. Both tag-to-reader data-rate and reader-to-tag data-rate are chosen to be 80 kbps. By generating 900 tags (with ID long =96 bit) randomly, the binary tree of the FPBS protocol will have 77598 internal nodes which equals the number of tags transmitted bits. The reader will send 900 bit as collision reports. The total exchanged bits among the reader and tags are 78498 bits. It consumes 0.9812 second under bit rate of 80 kb/s.

Fig. 8 shows the average required overhead-bit for one tag identification (as the cost due to prefix and iteration overhead). In the worst case, if we consider that reader overhead equals one bit per tag. Then, the proposed FPBS consumes the least reader overhead in the tag identification.

Fig. 9 shows the results of identifying 256 tags with variable ID length from 8-bit to 64-bit. Assume that the time of transmitting one bit is 5 µs. Then, if the proposed protocol is used to identify the 100% tag density of 8 bit ID length (i.e.2^8=256 tags are in the interrogator’s operating range), the reconstructed binary tree has 255 internal nodes (without including the tree leaves). It needs one bit tags' reply at each node plus 255 collision reader report. There is a collision in each node. The equivalent number of transferred bits is 510 bit. The total identification time = 510*5µs = 2550 µs. It provides 10 µs per one tag identification according to the proposed FPBS. However, the NEAA algorithm in [14] consumes 16.8 µs to identify one tag.

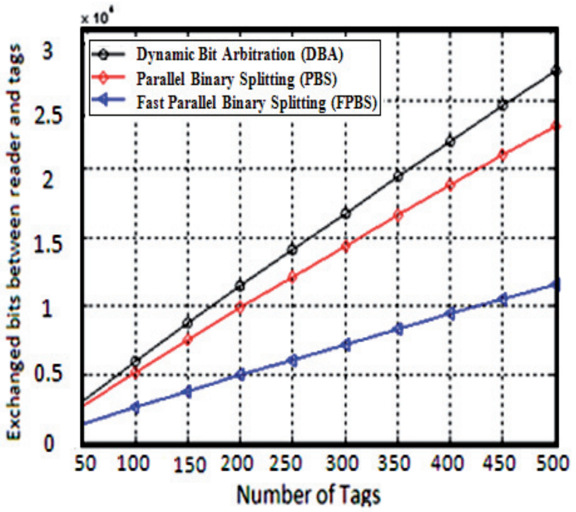

Fig. 6. Total transferred bits vs. the number of tags, for random IDs.

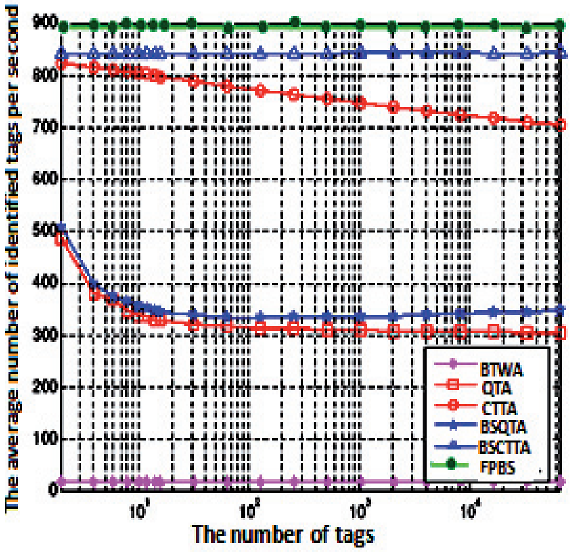

Fig. 7. The average number of identified tags per second.

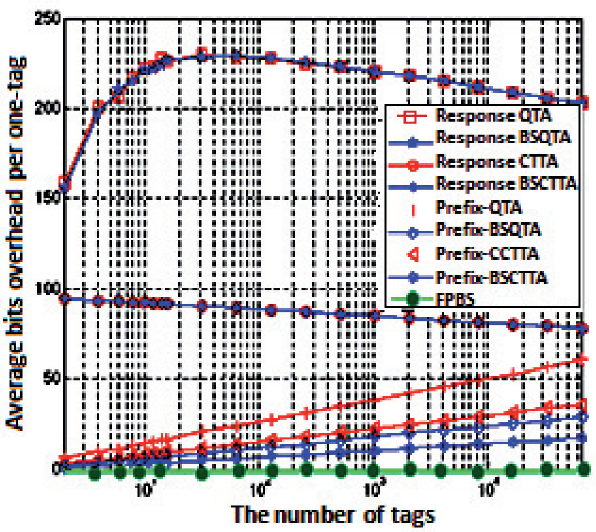

Fig. 8. Search cost (Bits): average required bits for one-tag identification.

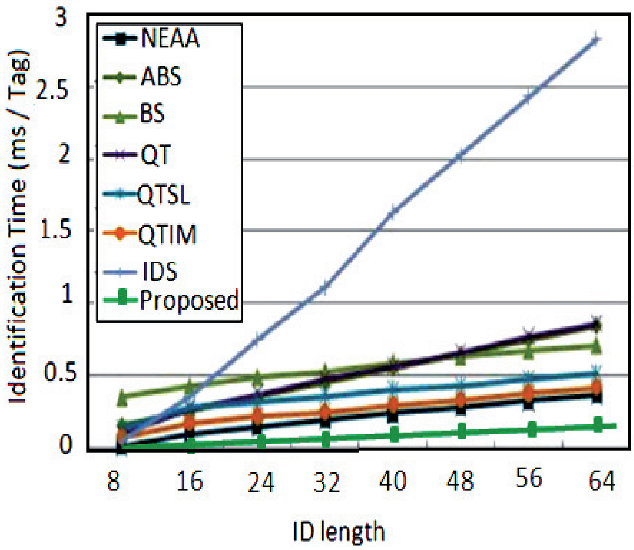

Fig. 9. Average identification time for each algorithm.

8. FPBS Performance in Successive Sessions

In this section, the performance of the proposed anti-collision algorithm will be explored in the case of some tags are arriving or leaving in successive reading cycles.

8.1 Performance of ABS and BA Protocols

Adaptive binary splitting technique (ABS) [11] can avoid collisions among staying tags, but it cannot prevent arriving tags from colliding with staying tags. The Blocking ABS (BA) protocol in [15] makes use of tag's counters to save the order of reply to recognize tags in the current interrogation session (frame), hence the tag preserves the obtained identification order from the last frame to avoid unnecessary collisions and idle cycles generated from identifying the staying tags in the current frame. Moreover, it avoids the collisions between the staying tags and the newly arriving tags. The reader starts the successive session by informing the tags the number of recognized tags in the last frame. Hence, all arriving tags will change their order (counters) to random number larger than the number of the recognized tags in the last frame and smaller than the total predicted number of tags in the current frame. However, the BA algorithm has longer reader response, random splitting rule and the more complex tag operation, hence larger overhead per tag [12].

8.2 FPBS Performance

The proposed FPBS achieves fast performance in the first and in the successive reading rounds. Fig. 10 shows the example of the binary tree of the four tags that is studied in section 5. Each tag stores its relative-order in the current order register (COR). The total number of the identified tags is found in the current path register (CPR).

For example, tag “C” has the COR=3. All tags have CPR=4 which is the total count of identified tags at the session end.

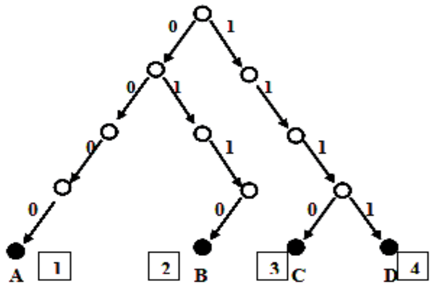

Fig. 10. The identified tree in the last reading session.

To use the advantages of the order based FPBS protocol, we can define two types of reader sessions:

Checking session: the reader checks the existence of the recognized tags in the last session. The reader starting command contains the total count of the identified tags “CPR”. All tags that have the same “CPR” will respond with one bit in its relative order (allocated time slot) to inform the reader with its existence. The reader and tags are operating in simpler one-to-one dialog in a frame with CPR time slots.

For example, if the reader starts checking session for the tree shown in Fig. 10, the “CPR” represents a frame with four time slots. Each tag should respond by one bit in its time slot. If the tag responds to the reader, the reader replies one bit ACK “0” to say “no change in the relative orders”. However, if tag “C”, for example, is going out the reader range, its allocated time slot will be idle, and the reader will send one bit ACK “1” to say “there is a leaving tag in that time slot”, and the other tags will update its relative orders. Then, tag “D” will modify it’s COR to be 3 instead of 4. All tags (A, B, D) will modify its CPR register to be 3.

In general, it is shortening the need of transmitting bit to 96 bit for every tag ID to one bit response representing its existence in its allocated time slot.

PBS reading session: the reader command starts the parallel splitting in the binary tree for the newly arriving tags.

* If the command contains “CPR”=1, then all tags will reset its counter and registers, and operate in the normal parallel splitting.

* If the command contains “CPR” > 1, then:

- *The tags that have the same “CPR”* will remain silent, because they were recognized in the previous session. Hence, staying tags are prevented from colliding with arriving tags. Staying tags will listen to the reader responses in the splitting process of the arriving tags, and update its “CPR”. The “CPR” is incremented at each collision report. The new “CPR”= the old “CPR”+ the number of reader collision reports +1.

- *The tags that have different “CPR”* will reset its counter and registers, and operate in the normal parallel splitting. However, at the end of the session, it will add the “CPR” value that received from the reader command to both “CPR” and “COR” registers. This action will combine the two subgroups of the staying and the arriving into one recognized group that can be checked in successive reading sessions.

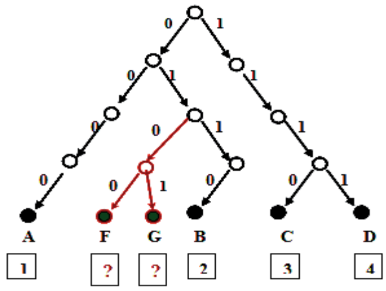

Fig. 11. The new tree in the next reading session.

Suppose that there are two arriving tags (F=0100 and G=0101), for example, as shown in Fig. 11. The reader sends command that contains the previous “CPR”=4 and the four tags (A, B, C, D) will remain silent. Then, tags will listen to the number of reader collision reports. Tags (F and G) have one collision node; hence all tags will update its CPR to be 6. The relative orders (COR) of the two tags F and G will be 5 and 6, respectively.

9. Conclusion and Future Work

This paper presents fast tag anti-collision algorithm based on parallel binary splitting (FPBS). The proposed algorithm overcomes the prefix and iteration-data overhead of the previously protocols. The tag achieves self transmission control using two counters and two registers. The tag operation needs only the collision notification that is provided by the reader in the identification process. The tags modify its replying order in the next bit level according to the collision condition. The major advantages of the proposed scheme are the low implementation complexity and the minimum number of transferred bits between the reader and the tags in the identification process with the same bandwidth. It consumes one bit per tag as a reader overhead. So, the number of transmitted bits is equal to the number of the binary tree nodes of the existing tags. It also presents faster performance in the successive reading cycles. The proposed algorithm can check the existence of the previously recognized tags very fast. Moreover, it prevents the current tags from collision with the new tags using a minimum overhead. The performance analysis shows that the proposed full duplex operation technique outperforms most of the recent techniques in most cases.

Hardware implementation is suggested as future work for realizing the operation of the proposed protocols by using FPGA, and estimating the number of logic circuits consumed by the protocol.

References

[1] BONUCCELLI, M. A., LONETTI, F., MARTELLI, F. Tree slotted aloha: a new protocol for tag identification in RFID networks. In *Proc. of IEEE Int. Symposium on a World of Wireless, Mobile and Multimedia Networks,* 2006, p. 603-608.

[2] SAHOO, A., IYER, S., BHANDARI, N. *Improving RFID System to Read Tags Efficiently*. Kanwal Rekhi School of Information Technology, Indian Institute of Technology Bombay, approved for the degree of Master, June, 2006.

[3] JUNG-SIK CHO, JEA-DONG SHIN, SUNG K. KIM RFID Tag anti-collision protocol: Query tree with reversed IDs. In *The 10th International Conference on Advanced Communication Technology (ICACT)*. Feb. 2008, vol. 1, p. 225–230.

[4] JI H. CHOI, DONGWOOK LEE, HYUCKJAE LEE Query Tree-Based Reservation for Efficient RFID Tag Anti-Collision. *IEEE Communications Letters,* Jan. 2007, vol. 11, no. 1, p. 85-87.

[5] OKKYEONG BANG, CHOI, J. H., LEE, D. W., LEE, H. J. Efficient novel anti-collision protocols for passive RFID tags. *Auto-ID Labs White Paper WP-HARDWARE-050 March* ***2009***. Available at: www.autoidlabs.org.

[6] SHIYU LI, QUANYUAN FENG A novel anti-collision algorithm in RFID system. In *Electromagnetics Research Symposium*. Cambridge, USA, 2008, p. 207-210.

[7] CHEN-CHUNG LIU, YIN-TSUNG CHAN An anti-collision protocol of RFID based on divide and conquer algorithm. In *The 26th Workshop on Combinatorial Mathematics and Computation Theory*, 2009.

[8] ZIMING GUO, BINJIE HU A dynamic bit arbitration anti-collision algorithm for RFID system. In *IEEE International Workshop on Anti-Counterfeiting, Security, Identification*. 2007, p. 457 – 460.

[9] JUN-BO GUO, ZHEN-HUA DING ID-binary tree stack anti-collision algorithm for RFID. In *Proceedings of the 11th IEEE Symposium on Computers and Communications (ISCC '06)*. Cagliari (Italy), 2006, p. 207 - 212.

[10] SUNG HYUN KIM, POOGYEON PARK An efficient tree-based tag anti-collision protocol for RFID Systems*. IEEE Communications Letters,* 2007, vol. 11, no. 5, p. 449-451.

[11] JIHOON MYUNG, WONJUN LEE, JAIDEEP SRIVASTAVA Adaptive binary splitting for efficient RFID tag anti-collision. *Communications Letters, IEEE*, Mar 2006, vol. 10, no. 3, p.144-146.

[12] JIHOON MYUNG, WONJUN LEE, JAIDEEP SRIVASTAVA, SHIH, T. K. Tag-splitting: adaptive collision arbitration protocols for RFID tag identification. *IEEE Transactions on Parallel and Distributed Systems*, 2007, vol. 18, no. 6, p. 763 – 775.

[13] WEI-CHIH CHEN, SHI-JINN HORNG, PINGZHI FAN An enhanced anti-collision algorithm in RFID based on counter and stack. In *Proceedings of the Second International Conference on Systems and Networks Communications (ICSNC 2007*), Issue 25-31, 2007.

[14] YUAN-HSIN CHEN, SHI-JINN HORNG, RAY-SHINE RUN, JUI-LIN LAI, RONG-JIAN CHEN, WEI-CHIH CHEN, YI PAN, TERANO TAKAO A novel anti-collision algorithm in RFID systems for identifying passive tags. *IEEE Transaction on Industrial Informatica*, Feb. 2010, vol. 6, no. 1.

[15] YUAN-CHENG LAI, CHIH-CHUNG LIN A blocking RFID anti-collision protocol for quick tag identification. In *International Conference on Wireless and Optical Communications Networks, WOCN '09*. Cairo (Egypt), May 2009.

About Authors...

USAMA SAYED MOHAMMED received his B.Sc. and M.Sc. degrees from Assiut University, Assiut, Egypt in 1985 and 1993, respectively, and Ph.D. degree from the Czech Technical University in Prague, Czech Republic in 2000, all in Electrical Engineering. From 1988 to 1996, he was at the Faculty of Engineering, Assiut University, working as an Assistant Lecturer. From February 1997 to July 2000, he was a research assistant at the Department of Telecommunications Technology, Czech Technical University in Prague, Czech Republic. From December 1999 to March 2000, he was a research assistant in the University of California in Santa Barbara (UCSB), USA. From November 2001 to April 2002, he was a post Doctoral Fellow with the Faculty of Engineering, Czech Technical University in Prague, Czech Republic. Since February 2006, he has been an Associate Professor with the Faculty of Engineering, Assiut University, Egypt. He authored and co-authored more than 90 scientific papers. Usama has been selected for the inclusion in 2010 Edition of the Marquis Who's Who in the World. His research interests include telecommunication technology, wireless technology, wireless networks, RFID, image coding, speech coding, statistical signal processing, blind signal separation, and video coding.

MOSTAFA SALAH received his B.Sc and M.Sc. degree from Assiut University, Assiut, Egypt in 2003. He is currently a M.Sc. student at the Department of Electrical Engineering, Assiut University, Egypt. He worked with the Egypt Telecom and is continuing his research activity related to the wireless network protocols. His research interest is in the field of RFID. His main interest is the study of tag and reader anti-collision protocols in RFID systems.