상대적인 RFID 시스템 정보와 커뮤니케이션 공학의 anti-collision 프로토콜에 대한 연구

Kishan Karmacharya[1] · 최영일[2] · 최동유[1] · 한승조[1*]

[1]조선대학교

[2]조선이공대학

Comparative performance study of anti-collision protocols in RFID system Information and Communication Engineering

Kishan Karmacharya[1], Young-il Choi[2], Dong-you Choi[1], Seung-jo Han[1*]

[1] Dept. of Information & Communication Engineering, Chosun University

[2] Dept. of Computer Mechanic, Chosun College of Science&Technology

[1*]Corresponding author.(Email: sjbhan@chosun.ac.kr)

ABSTRACT

RFID is a generic term for technologies which use RF waves to identify, track, or categorize any object. A radio frequency identification (RFID) reader recognizes objects through wireless communications with RFID tags. Tag collision arbitration for passive tags is a significant issue for fast tag identification due to communication over a shared wireless channel. One of the research areas in RFID system is a tag anti-collision protocol. In this paper, various anti-collision protocols are discussed. The pros and cons of different anti-collision protocols are compared with each other and their performance is analyzed and the better performance anti-collision protocol is suggested.

Keywords

RFID, collision, anti-collision, Interrogator, Tag

I. Introduction

RFID (Radio Frequency Identification) is an automatic method of identifying unique items using radio waves. RFID is an emerging and prominent automated identification technology. A RFID reader recognizes an object through wireless communications with the tag which has unique ID information and is attached to the object. There are three basic components in RFID system: a tag, an interrogator and control software (often called middleware).

An RFID reader communicates with tags through radio frequency. So it requires no LOS and it has wider range for the identification of tags which is performed in a different manner than the bar code system. In bar code system the reader identifies a bar code through the light, so there is limitation in read rate, visibility, and contact. The basic function of an RFID tag is to store data and transmit data to the interrogator. Tags can either be active (powered by battery) or passive (powered by the reader field).

Collision is divided into two types viz readers collisions and tags collisions. Collision occurs due to sharing of same wireless channel. Collision is a major problem during the identification of the tags since either the reader may not recognize all objects or a tag identification process may suffer from long delay. Therefore, anti-collision protocols which enable the fast and correct identification regardless of the occurrence of collisions are required. Reader collisions occur when more than one tag reflect back a signal at the same time, called "multi-access". Tag collisions occur when multiple tags transmit IDs to a reader at the same time and prevent the reader from recognizing any tag. Since low-functional passive tags cannot figure out neighboring tags or detect collisions, so the development of a tag anti-collision protocol plays significant role in improving the identification ability of RFID systems.

Tag anti-collision protocols can be classified into deterministic methods and probabilistic methods in terms of the way of determining the point of transmission time. The deterministic methods are tree based protocols that determines the point of transmission on receiving a message from a reader and making a process from the message, that is, a decision of whether to respond to it. On the other hand, Probabilistic algorithms are based on the framed ALOHA scheme where the reader sends the frame length and a tag picks out a slot for the data transmission probabilistic tag anti-collision protocols use the random number generated by a tag to determine the point of transmission. Each tag generates a random number and waits for its transmission time according to the chosen number. They split the group of colliding tags into two subgroups until all tags are identified. The aloha based protocols reduce the probability of occurring tag collisions how tags respond at the different time. The taxonomy of tag anti-collision protocols is given below.

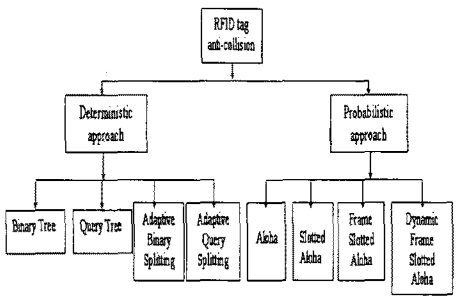

Fig.1 Taxonomy of tag anti-collision protocol

II. Tree-Based Anti-collision Protocols

In tree-based tag anti-collision protocols, a reader divides tags into two groups. A reader further divides each of them into two groups. The process of dividing tags is continued until a group contains only one tag so that each tag could be successfully identified. The dividing process of a group is continued until a reader identifies all the tags. The reader can recognize all tags in its identification range. The main advantage of tree-based tag anti-collision protocol is that all tags in identification range of the reader can be identified. An identification cycle is a process that constructs a tree from root node to leaf nodes but this algorithm requires precise tag timing synchronization to determine the position of the collision bits.

A. Binary tree protocol

Binary Tree Protocol requires tags to remember the previous inquiring results, thus reduces the average inquiry time. The binary tree protocol uses the pseudo-random number generator to divide tags into two groups. The counter variable in each tag is used for identifying each group. At the beginning of identification operation, the reader sends a message which notifies the start of its cycle to tags. All tags receiving this message generate random numbers of 0 or 1. Tags set their counter values by adding the generated random number to their counter values.

Tags are divided into two groups: one group has 0 in their counter values; the other group has the counter values of 1. The group with the counter value of 0 tries to transmit and wait for the reply from the reader. If a collision occurs, tags which tried to transmit their ID codes in the previous cycle are divided into two groups by using a pseudo-random number, and tags that did not try the ID transmission increase the value of their counters by 1. If there are no collisions, all tags decrease the value of their counters by 1.

In this protocol, a completely identified tag is eliminated. In every tag, there is a pointer. Every time the tag is reset, the pointer points to the highest bit of the tag's ID. With the ongoing of inquiring, it moves towards the lowest bit. During inquiring, the reader sends one inquiring bit at one time. The tags whose pointed bit is the same as the inquiring bit will send back their next bit to the reader while those who aren't goes to the "quiet" state and will not answer the remaining inquires in this round of inquiring until one tag has been eliminated and then all the remaining tags are reset. If the reader senses a non-collision answer, it uses it as its next-step inquiring bit. Otherwise if a collision is sensed, reader uses '0' as next-step inquiring bit. Thus for every round of inquiry, one tag, and only one tag will be identified, when its pointer finally gets to the lowest bit of the identified tag. Then the identified tag will be eliminated and all tags that have already entered the state of "quiet" will be reset, followed by a new round of inquiring beginning from the highest bit. After I round of inquiry, the IDs in the k tags will all be identified.

B. Query tree protocol

The query tree protocol is memory-less protocol. A reader transmits a query to tags. The query contains the prefixes of the tag identification (ID) codes. All tags within the range of a reader compare the query of the reader with their ID codes and transmit their ID codes to the reader when the result of the comparison is true. This protocol uses a query of reader and prefixes of tag ID codes to divide tags into two groups. Tags in one group transmit their ID codes to the reader while tags in the other group wait for the next query of the reader.

The content of a query is the identifier of each group. The reader repeats dividing tags into two groups until the number of tags in a group is one. When the number of tags in a group is one, the reader is successful in identifying the tag. This identification process can be considered as constructing a searching tree based on tag ID codes. The reader increases the length of the query until the identification cycle is completed. It is slower than binary tree protocol for tag identification.

The operation of the reader can be implemented by using a data structure (e.g. queue or stack). The queries set to 0 and 1 are stored in the data structure initially. When a collision occurs, the reader makes two queries whose lengths are 1 bit longer than the queries which cause the collision by concatenating query and the extra bit (0 and 1). Then the reader inserts itself to the stack or queue. When a readable slot or an idle slot occurs, the reader gets a next query from the data structure without any further processing. In case of using a stack for the data structure, the search is depth first search like the binary tree.

C. Adaptive binary splitting

The adaptive binary splitting protocol arranges the tags' transmissions via consecutive communications between a reader and tags. If a tag is within the reader's identification range, it is able to communicate with the reader directly. Tags transmit their own ID and then the reader detects collision. The reader always informs all tags whether or not the tag to reader signals collides. When this invert signals lead to collision, the colliding tags randomly select a binary number 0 or 1. Based on this selected number, a set of the colliding tags is split into two subsets. By continuing this splitting procedure until tags enable to transmit without collision, the reader can recognize all the tags. Since each recognized tag gets an exclusive time for transmission, ABS can reduce the number of collisions of the tag-to-reader signals and identify tags fast in the next process of tag identification. The tag maintains progressed slot counter (PSC) and allocated slot counter (ASC). PSC represents the number of timeslots passed in a frame and is initialized with 0 at the beginning of the frame. All the tags always have the same values of PSCs. The values of PSC are increased by 1 when a tag is successfully recognized by the reader. ASC signifies the sequence that the tag can access a channel for the transmission. The tag has one of three states: wait state, active state and sleep state. The tag transmits its ID to the reader only in the active state in which the tag has the allocated-slot number equal to the progressed-slot number.

D. Adaptive Query Splitting

AQS recognizes tags using queries sent by a reader. A query includes a bit string. The tag responds with its ID when its first bits of the ID are equal to the bit string of the query. Collisions are resolved by two 1-bit longer queries. The reader has queue Q which maintains bit strings for queries. At the beginning of the frame, Q is initialized with queries of all the leaf nodes in the tree of the last frame. To do this, the reader also has candidate queue CQ, which compiles queries of readable cycles and idle cycles of the ongoing frame. When a new frame starts, the reader initializes Q with bit strings in CQ and makes CQ empty. Accordingly, the reader does not transmit queries which caused tag collisions in the last frame. Tags still require very simple functions such as matching their IDs with the queries and are recognized with few collisions.

III. Probabilistic Anti-collision Protocols

The Aloha protocol is a very simple TDMA protocol: a tag begins transmitting as soon as it is ready and has data to send. The implicit start of the exchange between the tags and a reader, with the tags automatically sending their IDs upon entering a powering field, is one of the most basic properties observed in Aloha protocols. This is referred to as a "Tag-Talks-First" behavior, the opposite of which would be a "Reader-Talk-First" behavior.

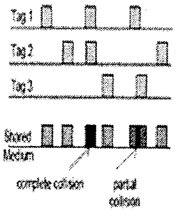

Fig. 2. (pure) Aloha example

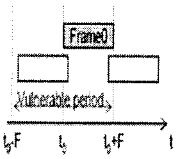

Fig. 3. Aloha frame vulnerability

If by misfortune some other tag has data to send around that time (whether earlier or later) and the interval during which the two tags transmit overlaps, then a complete or partial collision occurs and the vulnerability period is 2F.

Several extensions have been proposed in order to increase Aloha's feasibility and efficiency. The first technique is named "Switch-off" under which successfully decoded tag responses result in the tag automatically entering a Quiet state where it no longer transmits its ID to the reader.

The second method called "Slow-down" is a compromise between pure Aloha and the "Switch-off" extension, whose goal is to diminish tags' reply frequencies. This is accomplished by the reader sending a certain tag a slow-down command when it feels overwhelmed by responses from this particular tag. The singled-out tag then adapts the randomness of its backoff algorithm, such that the rate at which it transmits its ID is reduced.

The third method "Carrier Sense" is meant as a way to confer to the tags a means of listening to the medium and determine if a transmission is currently in progress. The reader uses its capacity to listen to the medium in order to convey extra information to the tags. A special MUTE command is broadcast to the remaining tags in the reader's field as soon as possible after a transmission is detected. The earlier a MUTE is sent, in effect silencing tags for the (predetermined) length of an ID transmission, the smaller the probability that another transponder has started a colliding transmission.

B. Slotted-Aloha

In the Slotted-Aloha protocol time is divided into discrete time intervals, called slots. Slotted ALOHA algorithm is the tag identification method that each tag transmits its serial number to the reader in the slot of a frame and the reader identifies the tag when it receives the serial number of the tag without collision.

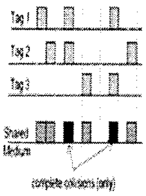

Fig. 4. Slotted-Aloha example

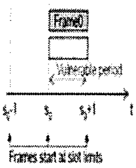

Fig. 5. Slotted-Aloha frame vulnerability

In this protocol either the packets are completely collided or do not collided at all. The problematic partial collisions that are observed in Aloha are eliminated. The vulnerability time in this case is half that of the Aloha protocol. The disadvantage however, is that such a scheme requires a synchronization mechanism in order for the slot-begin to occur simultaneously at all tags. This is accomplished either dynamically by having the reader send out slot-delimiting beacons, or statically using a pre-defined timer internal to the tags.

The terms "Muting", "Switch-off" and "Terminating" are used alternatively to express the idea of a Quiet state. The "Terminating" extension is similar to the "Switch-Off" method mentioned for Aloha, in that a successfully decoded response leads the tag to automatically enter a Quiet state in which it no longer transmits its ID. The main advantage of tags switching to a Quiet state is that unnecessary collisions due to tags replying indefinitely are avoided.

A more interesting extension to the Slotted-Aloha protocol is referred as "Early End". The slots are delimited by beacons sent by the reader known as SOF (start-of-frame) and EOF (end-of-frame. A silent period caused by the tags randomly waiting out several slots can be reduced to a fraction of the time wasted if all slots were waited out to their full length, thereby reducing the overall tag identification time.

C. Frame-Slotted Aloha

A Frame-Slotted Aloha protocol is built by taking Slotted-Aloha and the discrete time division one step further by grouping several slots into frames and each frame having N slots.

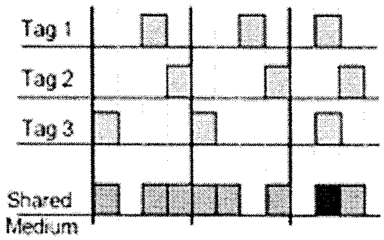

Fig.6. Frame-slotted Aloha

Frame-slotted Aloha is the extension of a Slotted-Aloha. No significant change is made to the definition of the slot architecture, but the tags are required to transmit exactly once every frame in a randomly selected slot within each frame N, the number of slots in a frame. Tag repetitiveness is strongly bounded in a way similar to a system where the reader would repetitively and constantly communicate. The extra synchronization overhead required by Frame-slotted Aloha is of the same order of magnitude as for Slotted-Aloha. Using the previous methods, a tag exhibiting a too high response frequency is pointlessly colliding with potentially valid response from other tags in the field. Frames regrouping several slots implicitly bound this repetitive behavior by setting a lower-limit to the number of messages a tags transmits to one per frame and at the same time establishing an upper limit by preventing more than one message being sent per frame.

D. Summary

Aloha		Slotted Aloha		Frame slotted Aloha	
Advantages	Disadvantages	Advantages	Disadvantages	Advantages	Disadvantages
1. Quickly adapts to varying number of tags 2. simplest reader design: "listen"	1. Worst case: never finishes 2. theoretically proven maximum channel utilization 18.4%	1.less of a free "free for all" 2. double the channel utilization of Aloha	1. 36.8% channel utilization 2. require synchronization 3. tags need to count slots	1. automatically diminishes each tag's repeat rate to once per frame	1.requires synchronization 1. frame size needs to be known/transmitted 2. tags need to count frames/slots

Table 1: comparison of probabilistic anti-collision protocol

E. Basic Framed Slotted ALOHA (BFSA) Algorithm

BFSA algorithm uses a fixed frame size and does not change the size during the process of tag identification. In BFSA, the reader offers information to the tags about the frame size and the random number which is used to select a slot in the frame. Each tag selects a slot number for access using the random number and responds to the slot number in the frame.

F. Dynamic Framed Slotted ALOHA (DFSA) Algorithm

In this protocol the reader is able to temporarily expand or contract the number of slots in a frame for the upcoming request round. The number of slots can then follow approximately the number of tags in the field, either reducing the number of collisions in a frame by increasing the number of slots, or decreasing them if there are too many empty slots. The three factors number of collisions, number of successful replies and number of free slots are combined into a ratio which should determine the most adequate frame-size for the next round of reader listening. So DFSA algorithm can solve partially the problem of BFSA that is inefficient to identify the tag. DFSA algorithm has several versions depending on the methods changing the frame size.

DFSA algorithm can identify the tag efficiently because the reader regulates the frame size according to the number of tags. But, the frame size change alone can not reduce sufficiently the tag collision when there are a number of tags because it can not increase the frame size indefinitely. Also, when the number of tags is small, then it can identify all the tag without too much collision. However, if the number of tags is large, it needs exponentially increasing number of slots to identify the tags because it always starts with the initial minimum frame size after identifying a tag, regardless how many tags are unread.

Ⅳ. Performance evaluation and analysis

Tree based protocols use binary search tree as their searching method to identify tags. The performance of anti-collision protocols is represented by average slot delay. The average slot delay of binary tree protocol approximates to 2.885*m (where m is the no. of tags) and that of the optimal frame slotted ALOHA protocol approximates e*m. The aloha based protocols reduce the possibility of the occurrence of tag collisions how tags transmit at the distinct time. In aloha based protocol the collisions cannot be perfectly prevented. Furthermore, they have the serious problem known as tag starvation. On the other hand, the tree based anti-collision protocol do not cause the tag starvation problem but they have relatively long identification time delay. Due to the fact that the tree based anti-collision protocol is free from tag starvation so this method is the fast identification of tag.

In a time slot of the binary tree protocol and ABS, tags transmit first and then the reader responses. The reader of the binary tree protocol transmits only one bit in order to indicate whether collision occurs or not. Similarly, the reader of ABS transmits two bits whose values match the empty, readable, or collisional time slot respectively. On the other hand, the reader, in the query tree protocol, transmits first and then tags response it with their IDs. The reader's signal contains one prefix (the maximum length of the prefix is equal to the length of tag's ID). Therefore, the time slot period of the query tree protocol may be, relatively, longer than the period of other protocols.

The performance of the query tree and the binary tree show better performance than the probabilistic protocols. ABS and AQS experience the least amount of delay among the anti-collision protocols. This is because they already have knowledge of the population of tags. Among probabilistic protocols, the performance of DFSA is the best where it can estimate the number of tags more precisely. ABS and AQS achieve good performance compared with other protocols because they can take advantage of the identification information of the previous stage. ABS and AQS achieve good performance than other protocols and they can recognize tag faster whether the tag is reader's range or not.

Ⅴ. Conclusion and Future works

In this paper, various anti-collision protocols are introduced. The positive and negative aspects of each anti-collision protocol are explained and their performance are analyzed and compared with each other. ABS and AQS experience the least amount of delay and achieve good performance among the different anti-collision protocols. In addition, among the probabilistic protocols, the performance of Dynamic Frame slotted aloha is the best. Here in this paper, we have compared the performance analysis of various anti-collision protocols in RFID system and in near future we will focused in the simulations of different anti-collision protocol to so that ABS and AQS outperform the other anti-collisions protocols.

References

1. Jihoon Myung and Wonjun Lee "Adaptive Binary Splitting: A RFID Tag Collision Arbitration Protocol for Tag Identification" IEEE 2005, pp 175-182

2. V.Daniel Hunt, Albert Puglia, Mike Puglia "RFID A guide to radio frequency identification" (2007)-Wiley, pp 1-23

3. Kashif Ali, Hossam Hassanein "RFID Anti-collision Protocol for Dense Passive Tag Environments" IEEE 2007, pp 819-821, 32nd IEEE Conference on Local Computer Networks

4. Naval Bhandari, Anirudha Sahoo and Sridhar Iyer "Intelligent Query Tree (IQT) Protocol to Improve RFID Tag Read Efficiency" IEEE 2006, 9th International Conference on Information Technology (ICIT'06)

5. Ji Hwan Choi, Dongwook Lee, Youngwoo Youn, Hyoungsuk Jeon, and Hyuckjae Lee "Scanning-Based Pre-Processing for Enhanced Tag Anti-Collision Protocols" IEEE 2006, pp 1207, pp 1210

6. Nowshad Amin, Puah Weng Lin "Anti-collision Protocol Development for Passive RFID Tags", Proceedings of the 7th WSEAS International Conference on Applied Informatics and Communications, Athens, Greece, August 24-26, 2007, pp 393-394

7. Syed Ahson, Mohammad Ilyas "Rfid handbook Applications, Technology, Security and Privacy" CRC Press, pp 184-179

8. Xu Huang and Dat Tran "Adaptive Binary Splitting for a RFID Tag Collision Arbitration Via Multi-agent Systems" Springer-Verlag Berlin Heidelberg 2007, pp. 926 - 933.

9. Jihoon Myung, Wonjun Lee, Jaideep Srivastava and Timothy K. Shih "Tag-Splitting: Adaptive Collision Arbitration Protocols for RFID Tag Identification" IEEE VOL. 18, NO. 6, JUNE 2007 pp 763-775

10. Luc André Burdet "RFID Multiple Access Methods" ETH Zurich, Summer semester 2004, pp 1-8, pp 11, Seminar "Smart Environments"

11. Su-Ryun Lee, Sung-Don Joo, Chae-Woo Lee "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification "IEEE 2005, Proceedings of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services

※This work was supported in part by MIC & IITA
<08-Infra-13, IT Research Infrastructure Program>