Fairness Analysis of STAC Protocol for 13.56MHz Passive RFID Systems

Intaek Lim

1Department of Embedded IS, Busan University of Foreign Studies

Abstract—In RFID System, when multiple tags respond simultaneously, a collision can occur. A method that solves this collision is referred to anti-collision algorithm. In 13.56MHz RFID system, STAC protocol is defined as an anti-collision algorithm for multiple tag reading. In STAC protocol, there is no differentiation between the collided tags and others in the identification process. Therefore, tags may never be successfully identified because its responses may always collide with others. This situation may cause the tag starvation problem. This paper analyzes the fairness performance for STAC protocol by the computer simulations.

Keywords—13.56MHz RFID, Anti-collision algorithm, STAC protocol, Fairness problem

I. INTRODUCTION

Unlike conventional barcode systems, radio frequency identification (RFID) is a wireless technology that uses electromagnetic or magnetic response exchange to identify objects at a distance without direct line of sight [1]. Recently, the RFID (Radio Frequency Identification) technique attracts a lot of attention due to its automatic identification capability for the identity information of an object [2]. The RFID systems provide an efficient and inexpensive mechanism for automatically collecting the identity information. RFID is an automatic identification system that consists of reader and tags [3]. A tag has an identification (ID) stored in its memory that is represented by a bit string. A reader is able to read the IDs of tags within its identification range by a simple anti-collision algorithm over the wireless channel. RFID systems have many advantages over barcode systems. RFID networks use radio frequency as a method of data transmission. Thus, unlike a barcode label, a tag does not need to be placed in a line of sight position from the reader in order to be successfully identified.

In the RFID system, tag identification process is performed by the reader’s query to tags and then the tag’s backscattering its identifier as its response. But if there are multiple tags within the identification range of reader, some of them might respond simultaneously and leads to collisions which decrease the performance. When multiple tags respond to the reader simultaneously, tag collisions occur and no tag can be identified by the reader successful. Therefore, the system requires a multiple-access scheme that allows the reader to read data from the individual tags. A technical scheme that handles multiple-access is called an anti-collision algorithm [4].

There are two types of anti-collision algorithms: deterministic and probabilistic algorithm [5][6][7]. The deterministic algorithm resolves collisions by muting subsets of tags that are involved in a collision. By successively muting larger subsets, only one tag will be left and finally led to successful transmission. Binary tree and query tree algorithms are the two main methods of the deterministic algorithm. The probabilistic algorithms are based on ALOHA-like protocol that provides slots for the tags to send their data. Almost all the probabilistic algorithms use framed slot ALOHA (FSA), which has been advanced in function by adding slotting and framing on ALOHA. The tags send their identifiers at a randomly selected slot. When collisions occur, the tags that are involved in collisions retransmit their identifiers in the next query round. The probabilistic algorithms may have limitations on the completeness of tag identification because there is still a probability of failing to be identified in a limited time period.
In almost all the 13.56 MHz RFID systems, slotted terminating adaptive collection (STAC) protocol is used for anti-collision algorithm for multiple tag reading [6]. The basic operations of STAC protocol are similar with the framed slot ALOHA (FSA) algorithm. FSA algorithm is based on the slotted ALOHA scheme with a fixed frame size. Therefore, the performance of FSA algorithm is dependent on the frame size and the number tags in the reader’s identification range. In case of small frame size, when the number of tags in the reader’s identification range is large, the identification time will increase because of the frequent collisions. On the other hand, when the number of tags is large, the number of wasted slots increases if the frame size is large. The tag identification time and system efficiency depend mainly on the frame size and the number of tags.

In STAC protocol, a query round begins with a BeginRound command that contains the number of slots in the forthcoming round. Tags that are not identified randomly select a new slot-count value after receiving the BeginRound command and involve in the identification process. STAC algorithm is simple. However, it has the tag starvation problem that a tag may never be successfully identified because its responses always collide with others. It is really hard to provide a fair identification delay. This will give a starvation to some tag. Therefore, this paper make a performance analysis for the STAC protocol in terms of fairness index.

II. STAC PROTOCOL DESCRIPTION

In STAC protocol, tags respond at randomly selected slots whose beginning and end are controlled by the reader [6]. A reader command signals both the end of the current and the beginning of the next slot. Figure 1 illustrates a reply round. A number of slots form a reply round.

![Figure 1. Structure of reply round.](image)

The reader’s operations of STAC protocol is depicted in Figure 2. As shown in the figure, the reader broadcasts a BeginRound command, which will cause a subset of tags to participate in the identification process. It contains several parameters. Some of these define the number of slots in the next round. Others define a selection from among the number of tags that will participate in the round. Another parameter is a hash value that is used to generate reply slot positions within a round. Tags that receive a BeginRound select randomly a reply slot position and load it into their slot-counter. When their slot-counter reaches the reply slot position, tags reply with their EPC code. The reader starts the identification process slot-by-slot basis. If the reader detects that two or more tag replies present or no tag reply present, it issues a CloseSlot command. On the other hand, if there is only one tag reply, the reader sends a FixSlot command. This process will be repeated until the end of reply round. The reader estimates the number of collided tags during a reply round. If there still exist unidentified tags in spite of the completion of reply round, the reader determines the number of slots based on the estimated number of tags and begins a new reply round.

Figure 3 illustrates the state transition diagram for tag in the STAC protocol. Tags which enter the energizing field, when they have sufficient power for operation, wait before replying in a READY state for the reception of one of several commands. The reader sends a BeginRound command with the number of slots in the round. The issuing by the reader of a BeginRound command causes the definition, within each of the selected tags, of a round size parameter, a selected or not selected flag, a proposed reply slot position, and sets to zero a counter of reply slot positions. Tags that are energized by the reader select a random slot number as the proposed reply slot and set their states to the SLOTTED READ state and slot counters to zero. In this state, tags calculate a proposed reply slot and wait until their slot counter reaches the proposed reply slot position, whereupon the tag will reply during the slot. This counter advances each time the reader sends the end of a slot.
A tag sends its response to the reader when its counter reaches the proposed reply slot position. The reply conditions within that slot can be separated into three categories: no tag reply present; one tag reply present; and two or more tag replies present. If the reader detects that no tag reply is present, the reader send a CloseSlot command, which signals to all tags in the SLOTTED READ state to increment their slot counter. If the reader receives a response correctly, it closes the slot by issuing a FixSlot command, which makes all tags to increment their counters and prompts the tag that was correctly heard to go into the FIXED SLOT state. In the FIXED SLOT state, the tag continues to reply in each round. If however, the reader detects a collision, it sends a CloseSlot command forcing all tags to increment their counters, while those tags that had responded in this slot,
realize that there was a collision since they did not receive the FixSlot command, and thus they select another slot for transmission.

III. PERFORMANCE ANALYSIS

In STAC protocol, the identification process will be continued by the reader until the end of query round even though a lot of slots in a round collide. A new query round begins after identifying a frame. In the new query round, tags that come into a collision in the previous round contend with other tags in selecting a slot counter value. This may cause a series of collision. Thus, it is anticipated that the identification delay for each tag will be various and it is hard to guarantee tags the fair identification delay.

In this section, for the purpose of analyzing the fairness of STAC protocol, we analyze the distribution of identification delay for each tag. The performance analysis was done by the computer simulations. It is assumed that the frame structure and slot length for simulation are same with the 13.56MHz RFID system proposed by Auto-ID center [6]. Also, the initial frame size and the minimum frame size are set to 16 slots. Figure 4 and Table 1 show the frame structure and the read cycle timing of 13.56MHz RFID system, respectively. The EPC code length assumes to be 64 bits for the timing values in the Table 1.

<table>
<thead>
<tr>
<th>Items</th>
<th>Value (µsec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Begin Round</td>
<td>1,623.68</td>
</tr>
<tr>
<td>Slot F</td>
<td>188.79</td>
</tr>
<tr>
<td>Success</td>
<td>2,756.48</td>
</tr>
<tr>
<td>Collision</td>
<td>2,114.56</td>
</tr>
<tr>
<td>Empty</td>
<td>226.54</td>
</tr>
<tr>
<td>Success</td>
<td>2,945.27</td>
</tr>
<tr>
<td>Collision</td>
<td>2,303.35</td>
</tr>
<tr>
<td>Empty</td>
<td>490.85</td>
</tr>
</tbody>
</table>

In this paper, we evaluate the performance for the proposed scheme through the computer simulations. The simulation system was developed with SMPL libraries[8] and MS Visual C++ 6.0. The system parameters for simulations are same as Table 1 in Section III. We compare the proposed scheme with STAC protocol. All the results of simulation were averaged after iterating 100 times. The performance measures of interest are the identification delay, identification speed, slot efficiency, and fairness index. The identification delay means the elapsed time to identify the tag. The identification speed is defined as the number of tags identified in a second. The slot efficiency is defined as the average number of tags identified in a slot. The fairness index rates the fairness of a set of values where there are $N$ tags. The delay fairness index is defined as follows [9]:

$$\text{Delay Fairness Index} = \frac{\max_{i=1}^{N}(d_i)}{\min_{i=1}^{N}(d_i)}$$
\[
\text{Fairness} = \frac{\left( \sum_{i=1}^{N} Y_i \right)^2}{N \sum_{i=1}^{N} Y_i^2}
\]  

where \(Y_i\) is the measured delay for the tag \(i\), and \(N\) is the total number of tags.

Figure 5 and Figure 6 illustrate the identification delay of each tag where the number of tags is 100 and 500, respectively. When the number of tags is 100, the average identification delay for each tag is 340msec and the standard deviation is 57.7msec. On the other hand, when there are 500 tags in the identification range of reader, the average identification delay for each tag and standard deviation are 1,857msec and 367.4msec, respectively. As shown in the figures, the more tags are in the identification range of reader, the more various the distribution of identification delay is. Therefore, it seems that a scheme needs to guarantee a fair identification delay.

**IV. CONCLUSIONS**

This paper made the performance evaluations for the STAC protocol of 13.56MHz RFID systems in terms of fairness index. In STAC protocol, a query round begins with a BeginRound command that contains the number of slots in the forthcoming round. Tags that are not identified randomly select a new slot-count value after receiving the BeginRound command and involve in the identification process. STAC algorithm is simple. However, it has the tag starvation problem that a tag may never be successfully identified because its responses always collide with others. It is really hard to provide a fair identification delay.
REFERENCES


