Computer Communications 32 (2009) 1862–1870

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Adaptive splitting and pre-signaling for RFID tag anti-collision

Ming-Kuei Yeh, Jehn-Ruey Jiang *, Shing-Tsaan Huang

Department of Computer Science and Information Engineering, National Central University, Taiwan, ROC

ARTICLE INFO

Article history:
Received 15 January 2009
Received in revised form 7 July 2009
Accepted 9 July 2009
Available online 19 July 2009

Keywords:
Collision resolution
RFID
Anti-collision
Tag identification

ABSTRACT

In an RFID system a reader requests tags to send their IDs by RF signal backscattering for the purpose of identification. When multiple tags respond to the request simultaneously, tag collisions occur and the tag identification performance is degraded. There are several tag anti-collision protocols proposed for reducing tag collisions. The protocols can be categorized into two classes: ALOHA-based and tree-based protocols that include deterministic tree-based and probabilistic counter-based subclasses of protocols. ALOHA-based protocols have the tag starvation problem; deterministic tree-based protocols have the problem that their performances are influenced by the length and/or the distribution of tag IDs. On the contrary, probabilistic counter-based protocols do not have such problems. In this paper, we propose a probabilistic counter-based tag anti-collision protocol, called ASPS, to reduce tag collisions by adaptively splitting tags encountering collisions into several groups according to the estimated number of tags to be split, and to reduce the number of messages sent between the reader and tags by utilizing a pre-signaling bit. We simulate and analyze ASPS and compare it with related ones to show its advantages.

1. Introduction

The RFID (Radio Frequency IDentification) technique attracts a lot of attention recently due to its automatic identification capability through RF communication [1]. An RFID system consists of a reader and one or more tags. Tags store unique IDs and are attached to objects; a reader recognizes an object by issuing RF signals to interrogate the ID of the attached tag. According to the source of power supply, tags are classified into two types: active tags, which contain a battery and can transmit signals autonomously, and passive tags, which contain no battery and derive energy from the RF field generated by the reader to transmit signals passively. Most RFID tags are passive; they have the advantage over other electronic products that are energized by batteries or other power sources. Furthermore, tags are usually of tiny sizes and low costs. The RFID system is thus suitable for many applications, such as logistic control, supply chain management, and asset tracking, etc.

When a tag and a reader are close enough, they can communicate with each other. For such a situation, we say that the tag is in the *interrogation zone* of the reader. To figure out which tags are within the interrogation zone, a reader initiates an identification procedure (or interrogation procedure) to request tags to send back their IDs. When multiple tags respond to the reader simultaneously, tag collisions occur and no tag can be identified by the reader successful. How to reduce tag collisions to speed up the identification procedure is thus important. There are several tag anti-collision protocols proposed for reducing tag collision. According to [2], they can be categorized into two classes: ALOHA-based protocols and tree-based protocols that include deterministic tree-based and probabilistic counter-based subclasses of protocols.

In ALOHA-based protocols [3–7], tags respond to the reader by transmitting IDs in a probabilistic manner. For example, in slotted ALOHA protocol [4], the whole interrogation procedure period is divided into several time slots, and each tag randomly chooses a time slot for transmitting its ID to the reader. ALOHA-based protocols are simple; however, they have the *tag starvation problem* that a tag may never be successfully identified because its responses always collide with others'.

The basic idea of the tree-based protocol is to repeatedly split the tags encountering collisions into subgroups until there is only one tag in a subgroup to be identified. The tree protocols do not have the tag starvation problem. In order to emphasize the different mechanisms for performing the tag-splitting based on either static tag IDs or dynamic counters, we classify the tree-based protocols into deterministic tree-based [8–11] and probabilistic counter-based [12–14] subclasses of protocols. The deterministic tree-based protocol relies on tag IDs and thus has the problem that its performance is influenced by the tag ID length and/or distribution, while the probabilistic counter-based protocol has not. We hence focus on counter-based protocols in this paper.

This paper presents a novel counter-based tag anti-collision protocol, called ASPS, using two schemes, adaptive splitting and

* Corresponding author. Tel.: +886 3 4227151; fax: +886 3 4222681.
E-mail address: jrjiang@csie.ncu.edu.tw (J.-R. Jiang).

0140-3664/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2009.07.011

pre-signaling, to reduce tag collision. By predicting the number *k* of tags to be split, ASPS adaptively splits tags into *k* groups. It is likely that each group has only one tag to be identified successfully. In this way, collision is reduced significantly. Furthermore, ASPS utilizes a pre-signaling bit to reduce the number of messages sent between the reader and tags. The tag identification delay is thus reduced. We simulate and analyze ASPS and compare it with related ones to show its advantages.

The rest of this paper is organized as follows. Some related work is introduced in Section 2. In Section 3, we describe ASPS protocol by elaborating the concepts of adaptive splitting and pre-signaling. In Section 4, we simulate and analyze ASPS and compare it with related protocols. And finally, conclusion is drawn in Section 5.

2. Related work

In this section, we introduce some representative ALOHA-based, deterministic tree-based and probabilistic counter-based tag anti-collision protocols.

2.1. ALOHA-based protocols

ALOHA-based protocols try to stagger tag response times in a probabilistic manner to reduce collisions. Below, we introduce some ALOHA-based protocols: ALOHA [3], slotted ALOHA [4], frame slotted ALOHA [5], and dynamic frame slotted ALOHA [6,7] protocols.

In ALOHA protocol [3], on receiving the reader's interrogation request, each tag in the interrogation zone independently chooses a random back-off time and responds its tag ID to the reader at that time. If an ID is received by the reader without collision, it can be identified properly and acknowledged by the reader. A tag with acknowledged ID will stop responding to the reader. On the other hand, an unacknowledged tag will repeatedly select a random back-off time to send its ID until it is identified and acknowledged by the reader. In slotted ALOHA protocol [4], the random back-off time must be a multiple of a pre-specified slot time. If collisions occur in a slot, the reader will notify the colliding tags to re-select a response time randomly. As shown in [15], the performance of slotted ALOHA protocol is twice that of ALOHA protocol since there is no partial collision of tag ID responses in the former protocol.

Frame slotted ALOHA protocol [5] is similar to slotted ALOHA protocol. However, to limit the response time, frame slotted ALOHA protocol divides the whole interrogation procedure into a set of frames. Each frame has a fixed number of time slots, and a tag sends its ID to the reader in only one randomly chosen slot during a frame period. One drawback of frame slotted ALOHA protocol is that its performance will degrade when the number of slots in the frame does not properly match with the number of tags in the interrogation zone. Dynamic frame slotted ALOHA protocols [6,7] try to eliminate the drawback by dynamically adjusting the frame size according to the estimated number of tags. They are therefore have better performance slotted ALOHA protocol. But they need many communication rounds to optimize the frame size before the identification process [6]. Under the assumption that tag IDs are with the same series in production (i.e., tags have the continuous tag ID numbers), paper [16] proposed LoF (Lottery Frame) protocol to reduce the number of communication rounds from $O(n)$ to $O(\log n)$ with the help of the geometric distribution hash function, where n is the total number of tags in the interrogation zone.

In general, ALOHA-based protocols are simple and have fair performance. However, some ALOHA-based protocols have the tag starvation problem that a tag may never be identified when its responses always collide with others'.

2.2. Deterministic tree-based protocols

Deterministic tree-based protocols rely on tag IDs to repeatedly split colliding tags into subgroups until there is only one tag in a subgroup to be identified successfully. Below, we introduce two representative tree-based protocols: query tree [8] and bit-by-bit binary tree [9] protocols.

In query tree protocol (QT) [8], a reader first broadcasts a bit string *S* of a specified length. The tag with an ID whose prefix matches with *S* will respond its whole ID to the reader. If only one tag responds at a time, the tag is identified successfully. But if multiple tags respond simultaneously, the responses collide. In such a case, the reader appends string *S* with bit 0 or 1 and broadcasts again the longer bit string (i.e., *S*0 or *S*1). In this manner, the colliding tags are divided into two subgroups. If there is only one tag in a subgroup, it can be identified successfully. The reader keeps track of the request strings needed to broadcast with the help of a stack and perform tag identification procedure until all tags are identified. QT protocol is a memory-less protocol because it does not require tags to be equipped with additional writable on-chip memory. QT protocol does not have the tag starvation problem and its identification delay is affected by the distribution and the length of tag IDs. Specifically, if the tags have long and continuous IDs, the request bit string will grow very quickly for identifying all tags. The delay time of the identification procedure will then increase significantly.

In bit-by-bit binary tree (BBT) protocol [9], on receiving a reader's interrogation request, each tag responds with the first bit of its tag ID. The reader then records and broadcasts 1 (resp., 0) if the received bit is 1 (resp., 0 or a colliding signal). Only the tags with the first bit being 1 (resp., 0) will respond with its next ID bit; other tags will go into a sleep mode. The above procedure will repeat bit-by-bit until the last ID bit is reached. The reader can then identify and mute one tag, and reset tags in the sleep mode to go through the interrogation procedure from some ID bit position. The bit-by-bit procedure is performed recursively and all tags can be identified. BBT protocol requires tags to be equipped with writable on-tag memory so that tags can keep track of the inquiring bit position. Like QT protocol, BBT protocol has no tag starvation problem and its performance is dependent on tag ID distribution and/or length.

2.3. Probabilistic counter-based protocols

Probabilistic counter-based protocols rely on dynamically changing counters to split colliding tags. Below, we introduce two probabilistic counter-based protocols, ISO/IEC 18000-6B tag anti-collision protocol [13] and ABS (Adaptive Binary Splitting) protocol [12].

The well known ISO/IEC 18000-6B standard [13] proposes a probabilistic counter-based tag anti-collision protocol (later we just name it *ISO/IEC 18000-6B protocol* for short). In the protocol, each tag maintains a counter which is initially 0. Every tag with counter value 0 can transmit its tag ID to respond to the reader's interrogation request. When a collision occurs, the reader will notify all tags of this. And the tags with counter values larger than 0 will increase their counters by 1, while the tags with counter value 0 will randomly add 0 or 1 to their counters. In this way, the colliding tags (i.e., the tags with counters value 0) are split into two subgroups. The splitting procedure will be repeated until there is only one or no tag with counter value 0. In the former case, the tag with counter value 0 can be identified successfully. And in both cases, the reader sends a command to inform all unidentified tags to decrease their counters by 1. In this way, every tag will be the unique one to have counter value 0 and be identified successfully.