

Article

Channel-Quality Aware RFID Tag Identification Algorithm to Accommodate the Varying Channel Quality of IoT Environment

Yen-Hung Chen ^{1,*}, Rui-Ze Hung ¹, Lin-Kung Chen ², Pi-Tzong Jan ³ and Yin-Rung Su ⁴

¹ Department of Information Management, National Taipei University of Nursing and Health Sciences, Taipei 11214, Taiwan; tfreemanhung@gmail.com

² Mathematics and Applied Mathematics (Finance and Statistics), School of Information Engineering, Sanming University, Sanming 365004, Fujian, China; coln01@gmail.com

³ Department of Applied Informatics, Fo Guang University, Yilan County 26247, Taiwan; ptjan@mail.fgu.edu.tw

⁴ National Taiwan University of Science and Technology, Taipei 10607, Taiwan; M10009112@mail.ntust.edu.tw

* Correspondence: pplong@gmail.com; Tel.: +886-2-2822-7101 (ext. 2720)

Received: 22 December 2018; Accepted: 11 January 2019; Published: 17 January 2019



Abstract: Radio Frequency Identification (RFID) technique is broadly adopted as the automated identification system for the Internet of Things (IoT). Many RFID anti-collision algorithms were proposed to accelerate the tag identification process. However, they misjudged some unreadable slots which were due to collision instead of the bad channel condition, causing low bandwidth usage. This study proposes the Channel-quality Aware Query Tree algorithm (CAQT) to improve the identification performance in an error-prone channel environment. CAQT has three novel features: (1) it estimates the channel quality continuously and statistically in the rapidly changing channel quality environment; (2) it asks the tag for retransmission or to split the collide tags based on the channel quality; (3) the number of the groups which it splits tags is based on the estimated number of tags collide in current slot. The simulation results show that CAQT uses less than 31% slots compared with the conventional algorithms. The simulation results also demonstrate that CAQT provides enhanced performance when the channel quality is varying especially in outdoor environment, for example, ticket checking for railway or subway system.

Keywords: RFID; tag identification; anti-collision; channel error; interference

1. Introduction

The Radio Frequency Identification (RFID) technique [1–3] is broadly applied as the automated identification system for Internet of Things (IoT) environment because of its feasibility, convenience, fast and contactless object identification method, as compared with the conventional bar-code system. An RFID system consists of a reader and several tags. Each tag has a unique identification (ID), and the reader identifies objects which attached tag through radio signal. During the tag identification procedure, tags may transmit signals to the reader simultaneously and collide each other, meaning that the bandwidth resource is wasted and the duration of the identification process is prolonged. A well-designed tag anti-collision protocol is thus required for RFID tag identification.

Many RFID anti-collision algorithms were proposed and can be classified into two categories, which are Aloha-based [4] and Tree-based [5] algorithms. The Aloha-based algorithms attempt to adjust the frame size to minimize the duration of the identification process by observing the probabilities of collision and successful transmission in current frame. On the other hand, the Tree-based algorithms repeatedly subdivide collided tags into groups until each group contains, at most, one tag.

The Tree-based algorithms can be further classified into the binary tree protocol (BT) and query tree protocol (QT). For BT, the collided tags are separated by randomly choosing a binary number. With regards to QT, the collided tags are split by the reader's query.

However, the channel quality changes rapidly in the wireless environment; an RFID reader cannot distinguish between a collision and a frame error. When there are frame errors, the conventional algorithms would misjudge the unreadable slots as collisions instead of bad channel condition, resulting in that the measured possibility of collision gets high. Therefore, the number of tags based on the possibility of collision would be overestimated, causing more unnecessary idle slots and lower bandwidth usage. This phenomenon is also called the channel error problem in RFID system [6].

The error resilient estimation and adaptive binary selection algorithm (ERE-ABS) and its extension was proposed to address this channel error problem [6,7]. ERE-ABS applies the packet error rate (PER) and number of tags to judge whether the received unreadable message is caused by collision or it is the result of channel error. However, ERE-ABS does not dynamically adjust PER and the number of tags, when the channel condition changes. This results in that the PER and the number of tags may be incorrectly estimated and therefore cannot reflect the actual condition with regards to the current RFID system, meaning that the idle slots increase, and the inferior bandwidth usage is inescapably caused.

This study therefore develops a novel RFID anti-collision algorithm, called the Channel-quality Aware Query Tree algorithm (CAQT), to improve the identification performance under error-prone channel. The CAQT is based on query tree protocol and owns three novel features:

- (1) it estimates the channel quality continuously and statistically in the rapidly changing channel quality environment to improve the tag estimation;
- (2) it asks the tag for retransmission or to split the collide tags based on the channel quality in order to reduce the number of idle slots;
- (3) the number of the groups which it splits tags is based on the estimated number of tags collide in current slot in order to accelerate the tag identification process.

The structure of this study is as follows. Section 2 presents the RFID system overview, error-prone channel, and previous studies. Section 3 introduces the concept and pseudo codes of the proposed CAQT. Section 4 demonstrates simulation results including the comparisons with CAQT and previous anti-collision algorithms. Section 5 concludes this study.

2. Background

This section provides an overview of RFID system, the RFID tag identification performance problem in error-prone channel, and the previous studies.

2.1. RFID System Overview

RFID is the system containing tags and readers, and it is applied to automatically identify and track tags attached to objects. Figure 1 shows some terminologies applied during the identification process. A frame is the duration from the moment a reader begins recognizing all tags within its reading range to the time it finishes all recognitions. Let f_i denote the i -th frame. A frame contains a set of slots. A slot is the duration when a reader transmits a triggering (feedback) signal to the tags and then the tags respond with their IDs to the reader. Let $S_{i,j}$ denote the j -th slot in the i -th frame. Notably, the status of a slot can be classified as idle, success, and collision. An idle slot indicates that the reader does not receive any signal from tags. The success slot means that the reader can successfully decode one or more tag IDs from the signal. A collision slot is the slot where the reader receives the signals but cannot decode any information from them. As shown in Figure 2, a collision is generally caused by that two or more tags choose the same slot to send message to the reader.

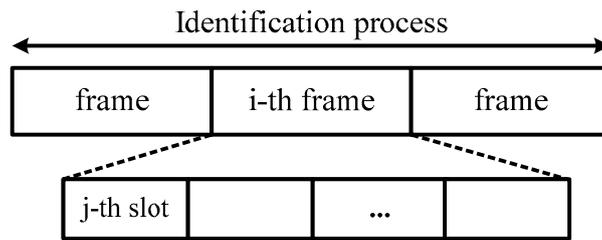


Figure 1. Relationship between slots and frame.

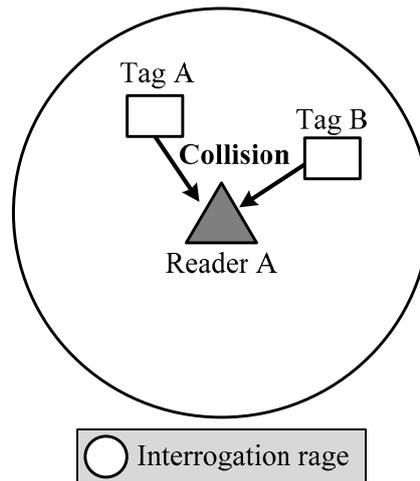


Figure 2. Collision in RFID (Radio Frequency Identification) system.

During the identification process, the reader periodically broadcasts a query or some information with regarding the frame in the first slot of the frame. The tags within radio coverage range of the reader will select one slot and response their own ID in this slot if their ID matches the query. Ideally, the number of slots within a frame should equal to the number of the tags within the coverage of a reader in order to maximize the bandwidth usage. Otherwise, some unused slots will exist if the number of slots is larger than the number of the tags, or the tags will suffer serious collisions if the number of slots is fewer than the number of the tags. Therefore, the RFID reader has to estimate the number of the tags to decide an appropriate frame size.

2.2. Error-Prone Channel

The RFID reader and tags frequently exchange messages within a frame. However, the signal strength of tags is typically very weak and significantly attenuated by distance comparing with the signal power of RFID reader. Therefore, the tags may be interfered from one or more readers [8,9], and this phenomenon is reflected in the form of packet error rate (PER). For example, as shown in Figure 3, the signal from Tag A to Reader B may be interfered by the signal transmitted by Reader A. Reader B cannot identify this phenomenon is caused by the interference from Reader A. Therefore, Reader B would wrongly regard it as that two or more tags simultaneously send message in the same slot, even though there is only one tag within its interrogation range.

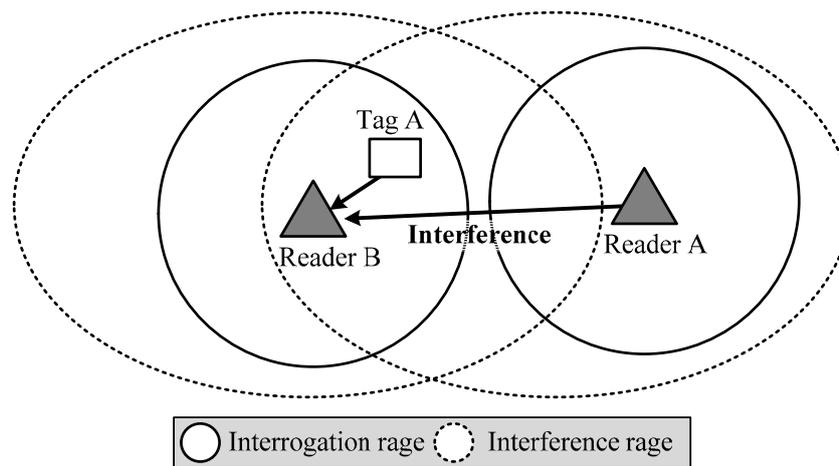


Figure 3. Example of the interference.

The inability to distinguish the difference between collision and channel error would cause that the reader inaccurately estimates the number of tags within in its interrogation range. This is because that most tag estimation methods are based on the probability of collision and the reader cannot distinguish the difference between collision and interference. This means that the reader will overestimate the number of collision slots and underestimate number of readable slots, as shown in Figure 4. According to the overestimated collisions, the tag estimation algorithm would wrongly regard that too many tags transmit message in current frame which causes high collision probability, and then it increases the number of slots in the next frame in order to decrease the collisions. Therefore, the slots in the next frame are overestimated and the bandwidth usage inescapably decreases.

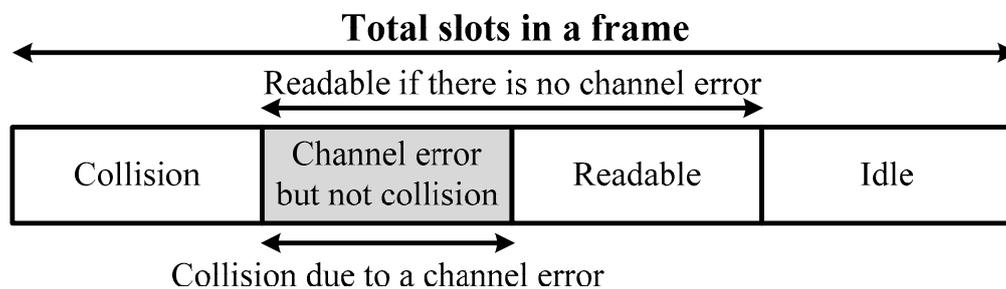


Figure 4. Relationship among collision, readable, idle and signal interference slot.

2.3. Previous Studies

Many RFID anti-collision algorithms were proposed to identify all tags by using the minimal bandwidth resource. These algorithms can be divided into two categories, which are the probabilistic methods and deterministic methods.

The probabilistic methods, which are also called the Aloha-based algorithms [10–18], attempt to adjust the frame size to maximize the bandwidth usage by observing the probabilities of collision and idle in current frame. That is, each tag randomly chooses one slot in current frame to send its message to the reader, and then the reader observes the number of collisions, readable, and idle slots in this frame in order to estimation the number of tags within its interrogation range. Ideally, in order to minimize the collision and idle slots, the number of slots in a frame should equal to the number of tags within a reader’s interrogation range. Therefore, the reader then modifies the next frame size to the number of the tags.

The deterministic methods, which are also called the Tree-based algorithms [19–31], resolve a collision among tags by repeatedly subdivided into groups until each group contains at most one tag. The Tree-based algorithms can be further classified into the binary tree protocol (BT) and query tree

protocol (QT). For BT, the collided tags are separated by randomly choosing a binary number and sending it back to the reader. This procedure repeats until the reader confirms each group contains at most one tag. With regards to QT, the collided tags are split by the reader’s query. If the tag’s ID prefix matches this query, the tag will respond with its ID to the reader. This query procedure continues until no collided response occurs for each query. However, these conventional algorithms did not consider the condition of channel error, meaning that the reader would overestimate the number of collided tags.

The error resilient estimation and adaptive binary selection algorithm (ERE-ABS) was proposed to handle the issue of channel error [6]. ERE-ABS applies the PER and number of tags to judge which event, i.e., collision or channel error, treated by the reader would acquire the highest throughput and then adopts the corresponding strategy, when the reader receives an unreadable message. That is, for example, if the throughput of regarding the unreadable message as the result caused by channel error is higher than the throughput of treating this message as a collision, the reader would ask the tag who transmitted this message to retransmit it immediately. Otherwise, the reader would treat it as a collision and adopt the Tree-based anti-collision procedure so as to identify the collided tags.

As shown in Figure 5, the ERE-ABS used two phases for accurate estimate channel status and improves identification efficiency. In the error resilient estimation (ERE) phase, the reader would attempt to split the number of tags into j groups until the probability of idle slots is smaller than the threshold. And then, the reader will operate the estimate process to calculate number of tags and channel state. In the adaptive binary selection (ABS) phase, the reader would apply the estimate result to judge retransmission or binary selection when it detects a collision slot.

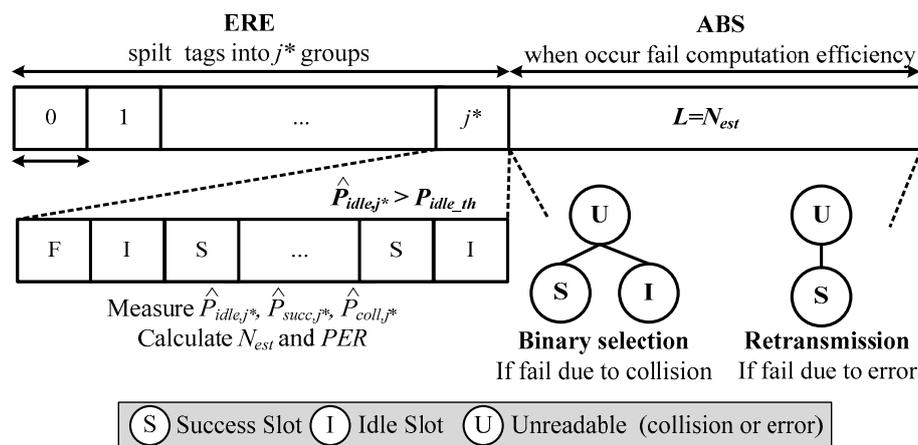


Figure 5. The mechanism of ERE-ABS.

However, the channel status changes rapidly and ERE-ABS does not dynamically adjust PER and the number of tags during the collision resolving procedure. This results in that the PER and the number of tags may be incorrectly estimated and therefore cannot reflect the actual condition with regards to the current RFID system, meaning that the inferior throughput will be inescapably caused.

3. Proposed Channel-Quality Aware Query Tree Algorithm

This study proposes the Channel-quality Aware Query Tree algorithm (CAQT) to minimize the effect of the rapidly changing channel quality on estimating number of the tags. The CAQT is based on query tree protocol and can be characterized by three features:

- (1) it estimates PER in each slot including the collision resolution process in order to improve the accuracy of the tag estimation in the rapidly changing channel quality environment;
- (2) when a reader receives an unreadable message from tags, it then adopts the most efficient strategy, which is to ask the tag to retransmit it or to split the collide tags, after evaluating the efficiency of regarding this event caused by channel error and that of treating it as a collision event;

- (3) the number of the group which it splits is based on the estimated number of tags collide in current slot in order to accelerate the tag identification process.

3.1. Channel Error Rate Estimation

Since PER is required to evaluate the efficiency of asking the unreadable tag to retransmit the message or to split the collided tags, CAQT estimates PER through sampling the slot in which only one tag transmits. The following shows the details:

First, the tree-based RFID anti-collision operation is utilized to distinguish whether the unreadable slot is due to collision or channel error. That is, if the reader only identifies one tag in the split slots after detecting an unreadable slot, this unreadable slot can be deducted as the slot interfered by noise rather than the collided slot, as shown in Figure 6.

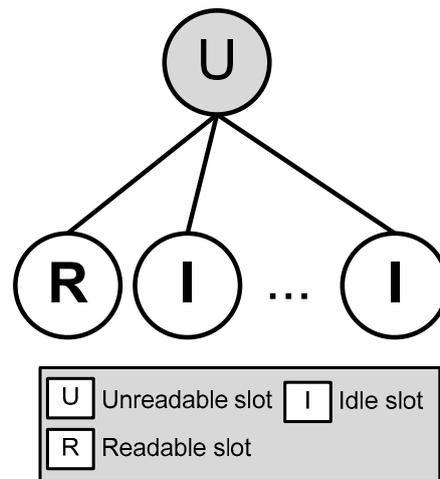


Figure 6. An example of unreadable slots.

Second, the reader estimates the PER by calculating the number of the readable slots between the two adjacent slots in which only one tag transmits and suffers channel error, named U_s herein. That is, as shown in Figure 7, the slots between two adjacent U_s , which means the slots starts after U_s^{i-1} and ends to (includes) U_s^i , can be categorized into three groups including readable slots, idle slots, and unreadable slots. The readable slots mean that only one tag transmits in these slots including U_s^i ; the unreadable slot indicates there are more tags transmitted in one slot, and these tags therefore collide; the idle slot is the slot that no tag transmits in it. The PER for the readable slots between U_s^{i-1} and U_s^i can be calculated by $\frac{1}{N_r^i}$, where N_r^i is the number of readable slots between U_s^{i-1} and U_s^i . Thus, to assume that the channel error is uniform distributed in each slot, the PER for the slots between U_s^{i-1} and U_s^i , named \widetilde{PER}^i , can be estimated by

$$\widetilde{PER}^i \approx \frac{1}{N_r^i} \tag{1}$$

Finally, the exponential smoothing method is applied to forecast the PER in the following slots, which is,

$$\widetilde{PER} = \alpha \times \widetilde{PER}^i + (1 - \alpha) \times \widetilde{PER}_{old} \tag{2}$$

where \widetilde{PER} is the estimated PER in the following slots and \widetilde{PER}_{old} is the previous estimated PER. In addition, the coefficient α is used to revise \widetilde{PER} .

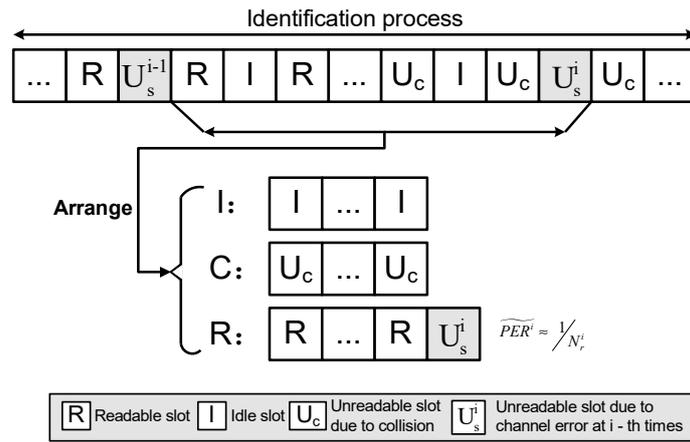


Figure 7. An example of PER (Packet Error Rate) between U_s^{i-1} and U_s^i .

3.2. The CAQT Algorithm

For the conventional tree-based collision resolution protocols, when a reader encounters an unreadable slot, this reader would regard collision occurs in this slot and then split the tags into several groups in order to resolve the collision. However, this strategy causes inferior performance because the unreadable slot may result from the channel error. To be more specific, when there is only one tag transmitting in one slot and encountering channel error, the reader would treat this slot as a collision slot and attempt to use more slots to resolve this collision. Nevertheless, only one slot will be used, and remaining slots will be wasted.

This study therefore provides a CAQT to overcome above mentioned issue. When a reader encounters an unreadable slot, CAQT will statistically judge whether it is simply due to channel error or it is caused by collision. If this unreadable slot is due to channel error, CAQT will ask the tag to retransmit its message in the next slot. On the other hand, CAQT will treat this unreadable slot as a collision and split the collided tags into several group to resolve the collision.

Figure 8 shows the flowchart of CAQT. The basic idea of CAQT is that, for example, when the reader detect a slot of which the status is unreadable, the reader would calculate E_{rtx} and E_{spl} to determine whether to ask the tag for retransmission or to split the collided tags.

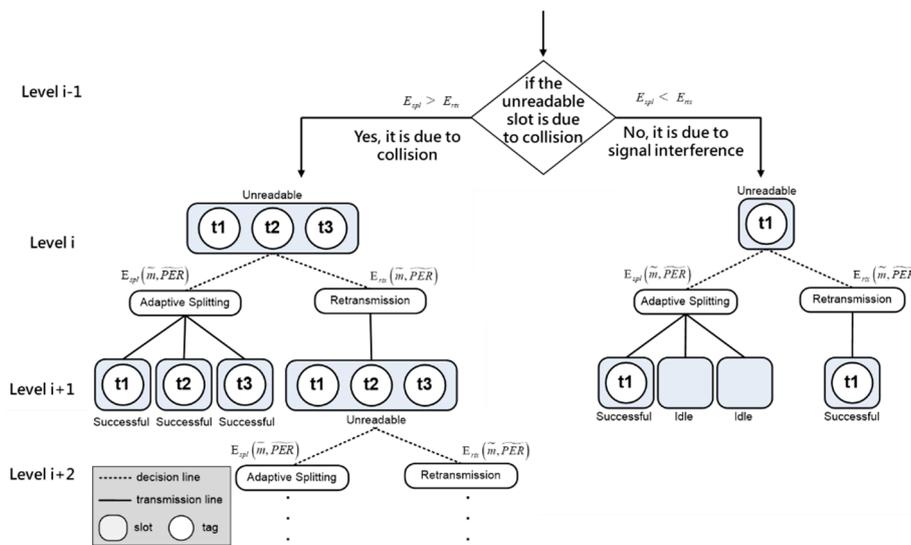


Figure 8. Flowchart of CAQT (Channel-quality Aware Query Tree algorithm).

If E_{spl} is lower than E_{rtx} . When the reader detects an unreadable slot, it then statistically calculates the efficiency. In this case, the slots consumed by the splitting procedure (3 slots) are more than that

consumed by retransmission (1 slot). Therefore, the reader then asks the tag to retransmit its message immediately. On the other hand, if E_{spl} is higher than E_{rtx} , the reader would treat this event as the collision and then adopt the Query tree-based algorithm to split the tags.

Figure 9 demonstrates the pseudo code of CAQT. Line 3~5 in the Main function and Line 8~10 in the query function as the same as the conventional query tree based algorithm for the initial query. Once an unreadable slot is detected in Line 7, CAQT has to estimate the number of tags (\tilde{m}) which transmit their message in this slot. To assume \tilde{m} is the margin between the current estimated number of tags within the reader radio coverage ($\widetilde{N_{curr}}$) and the previously estimated one ($\widetilde{N_{pre}}$) and is therefore the reason resulting in this collision, that is:

$$\tilde{m} = \left| \widetilde{N_{curr}} - \widetilde{N_{pre}} \right|, \tag{3}$$

The number of the tags within the reader radio coverage (\tilde{N}) can be estimated by

$$\tilde{N} = \frac{N_s}{\text{Pr}(q)} \tag{4}$$

where N_s is the number of identified tags and $\text{Pr}(q)$ is the probability to identify the N_s tags through the query string q . Since the procedure to resolve the collided tags is based on Query Tree algorithm, and thus the $\text{Pr}(q)$ can be estimated by

$$\text{Pr}(q) = \left(\sum_{i=1}^{\text{length}(q)} \frac{q_i}{2^i} \right) + \frac{1}{2^{\text{length}(q)}} \tag{5}$$

where q_i is the i -th character of the query string q and $\text{length}(q)$ is the length of the query string q [32]. It should be noted that the query string q is a binary string, which is $q = q_1q_2\dots q_j\dots q_k$ where $q_j \in \{0, 1\}$.

Then the reader has to compare the efficiency between asking the tags to retransmit and splitting the collided tags, which are denoted by $E_{rtx}(\tilde{m}, \widetilde{PER})$ and $E_{spl}(\tilde{m}, \widetilde{PER})$, respectively. The efficiency means the expected number of tags can be transmit in one slot. If the efficiency of retransmission is higher than that of splitting, as shown in Line 13–19, the reader will ask the tag to retransmit immediately. Otherwise, as shown in Line 20–29, the reader will adopt the query tree algorithm to split the tag for collision resolution.

Regarding the efficiency of retransmission, $E_{rtx}(\tilde{m}, \widetilde{PER})$ is calculated by:

$$E_{rtx}(\widetilde{PER}, \tilde{m}) = \sum_{r=1}^{\tilde{m}} \left(1 / \left(\sum_{k=1}^{\infty} k \times (1 - P_{rtx}) \times (P_{rtx})^{k-1} \right) \right) \tag{6}$$

where P_{rtx} is the probability of successful retransmission in current slot and k is the slots used in the k -th retransmission. Therefore, $k \times (1 - P_{rtx}) \times (P_{rtx})^{k-1}$ is the expected used slot given that the tag is successfully identified in k -th retransmission, and $\sum_{k=1}^{\infty} k \times (1 - P_{rtx}) \times (P_{rtx})^{k-1}$ is the expected number of slots to successfully retransmit a tag. Furthermore, P_{rtx} is the probability that has r tags successfully transmit in current slot and remaining $(r - 1)$ tags suffer channel error, which is deduced by:

$$P_{rtx} = \binom{r}{1} \times (1 - \widetilde{PER}) \times \widetilde{PER}^{r-1} \tag{7}$$

It should be noted that, if the reader still detects the unreadable slot after asking the tag to transmit, the reader will regard the unreadable slot is more likely due to collision rather than channel error. Then the reader will split the tag follows the procedure of the query tree-based algorithm to resolve the collision, as shown in Line 16–19.

The efficiency of splitting the tags by query tree-based algorithm, $E_{spl}(\tilde{m}, \widetilde{PER})$, is the expected slots to resolve the \tilde{m} collided tags and is calculated by:

$$E_{spl}(\widetilde{PER}, \tilde{m}) = \frac{\tilde{m}}{S(\widetilde{PER}, \tilde{m})}, \tag{8}$$

where $S(\widetilde{PER}, \tilde{m})$ is the total number of slots which use to recognize \tilde{m} tags and can be recursively estimated by:

$$S(\widetilde{PER}, \tilde{m}) = \begin{cases} 1 & , \tilde{m} = 1 \\ \sum_{i=1}^{\tilde{m}} \sum_{j=0}^{\tilde{m}} (1 + S(\widetilde{PER}, j)) \times \binom{\tilde{m}}{j} \times \left(\frac{1}{\tilde{m}}\right)^j \times \left(1 - \frac{1}{\tilde{m}}\right)^{\tilde{m}-j} & , \tilde{m} > 1 \end{cases} \tag{9}$$

where $\binom{\tilde{m}}{j} \times \left(\frac{1}{\tilde{m}}\right)^j \times \left(1 - \frac{1}{\tilde{m}}\right)^{\tilde{m}-j}$ the probability of j tags transmits in i -th slot. It should be noted that the number of slots applied to split the tags should be equal to the number of estimated collided tags ideally [33,34]. Therefore, there are \tilde{m} slots applied in current tag-splitting procedure.

In addition, as shown in Line 23–24 of Figure 8, when only one slot is readable and remaining slots are idle, this means the previous unreadable slot is due to channel error rather than collision. Therefore, the \widetilde{PER} should be updated by the procedure described in Section 3.1.

However, it should be noted that the time complexity of CAQT in the worst case scenario is $n \times (k + 2 - \ln n)$ [35] if all unreadable slots are due to collision.

```

1. // Reader procedure:
2. Main() {
3.   q = NULL,  $\tilde{m} = \widetilde{PER} = 0$ ;
4.   query(q,0);
5.   query(q,1);
6. }
7. query(char [] q, char c) {
8.   q += c;
9.   broadcast (q);
10.  receiveResponse();
11.  if (unreadable slot is detected) {
12.    calculate  $\tilde{m}$  by Eq. (3)–(5)
13.    if ( $E_{rx}(\widetilde{PER}, \tilde{m}) > E_{spl}(\widetilde{PER}, \tilde{m})$ ) {
14.      broadcast (q, 1);
15.      receiveResponse();
16.      if (unreadable slot is still detected) {
17.        query(q,0);
18.        query(q,1);
19.      }
20.    } else {
21.      broadcast (q,  $\tilde{m}$ );
22.      receiveResponse();
23.      if (only one slot is readable and remaining slots are idle) {
24.        update  $\widetilde{PER}$  by the procedure described in Section 3.2;
25.      } else if (unreadable slot is detected) {
26.        query(q,0);
27.        query(q,1);
28.      }
29.    }
30.  }
31. }

```

Figure 9. Pseudo codes of CAQT.

4. Simulation Results

Simulations were conducted to compare the performance of CAQT with the query-based (QT) algorithm [21], the typical aloha-based algorithm called dynamic frame-slotted Aloha (DFSA) [14,18],

and ERE-ABS [6] in terms of number of slots and number of idle slots under varying number of tags and PERs. The number of slots means the total number of slots to identify all tags, and the number of idle slots is the total number of idle slots during the tag identification procedure. The estimated number of tags during the progress of tag identification was also investigated to ensure DFSA can adjust the estimated number of tags under error-prone channel.

The simulation environment setting was as the same as that in [6] for fair comparison. That is, for the DFSA algorithm, the minimum frame size was set to the minimum frame size in order to estimate n through the tag estimation algorithm; regarding the ERE-ABS, L_{est} was set to 128, the threshold for probability of idle slots $P_{idle_th} = 0.2$, and the factor $f_d = 4$ used to reduce the number of tags; for CAQT, the α was set to 0.75 to estimate channel condition.

As shown in Figure 10, the effect of various PER ranging between 0 and 0.5 was explored when number of tags is 500. Figure 10a demonstrates that CAQT used the fewest slots to identify all tags, and Figure 10b also presents that CAQT had the fewest idle slots during the tag identification procedure. Therefore, CAQT outperformed QT, DFSA, and ERE-ABS under all PER cases. This was because QT and DFSA could not distinguish the unreadable slot was due to collision or channel error, meaning that QT and DFSA inescapably overestimated the number of tags within the coverage of the reader. Therefore, the overestimated number of tags caused that QT used too many slots to split the tags and results in lots of unnecessary idle slots for resolving the unidentified tags. On the other hand, the overestimated number of tags made DFSA difficult to accurately determine the optimal initial frame size and therefore suffer many idle slots.

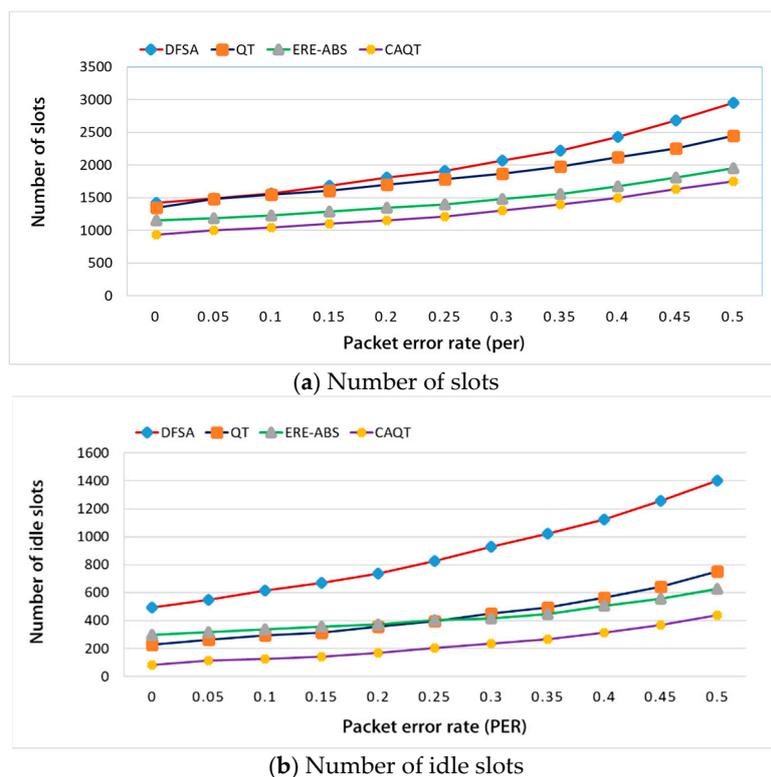


Figure 10. The effect of Packet Error Rate (PER).

With regards to ERE-ABS, its inferior performance was due to that ERE-BAS merely estimated the PER before the tag resolution procedure. However, the channel status changed rapidly, the estimated PER might not accurately reflect the current channel condition since ERE-ABS did not dynamically adjust the PER during the collision resolving procedure. This resulted in the number of the idle slots still being large.

CAQT owned the superior performance compared with other methods, and the higher PER was, the larger improvement CAQT had. As shown in Figure 10a, CAQT only used 69.0%, 57.8%, and 87.2% slots consumed by QT, DFSA, and ERE-ABS, respectively, when the PER is 0.5. The two reasons were

- (1) it estimated PER in each slot including the collision resolution process in order to improve the accuracy of the tag estimation in the rapidly changing channel quality environment, and
- (2) when a reader received an unreadable message from tags, it then statistically determined the most efficient strategy, which was to ask the tag to retransmit it or to split the collide tags.

Therefore, the number of unnecessary idle slots was reduced, as shown in Figure 10b, and the performance was improved.

It should be noted that the higher PER was, the more slots were interfered by poor channel condition, meaning that more slots were required to identify all tags by all algorithm.

The CAQT also had better performance comparing with others when PER was equal to 0. This is because, QT and ERE-ABS simply divided collided tags into two groups while detecting a collision event, resulting in that more slots were used to resolve the collide tags and poor performance was caused. On the other hand, CAQT can estimate the number of collided tags of a collision event and then separate tags into groups of which the number equals the estimated collided tags. This strategy can therefore minimize the slots used for resolving the collided tags and provide a better performance than other algorithms.

Figure 11 demonstrates the effect of various numbers of tags ranging between 500 and 4000 when PER was set to 0.3. Figure 11a shows that CAQT used the fewest slots to identify all tags and also had the fewest idle slots show in Figure 11b during tag identification procedure. Thus, the CAQT had superior performance compared with QT, DFSA, and ERE-ABS. The reasons were as the same as discussed in Figure 10. In addition, as shown in Figure 11a, the more tags existed in the reader’s coverage, the more slots were required to identify all tags by all algorithms.

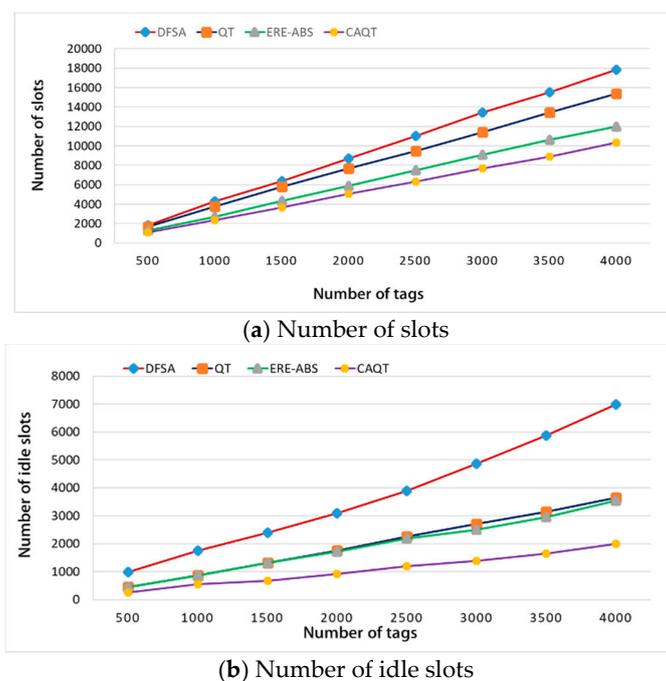
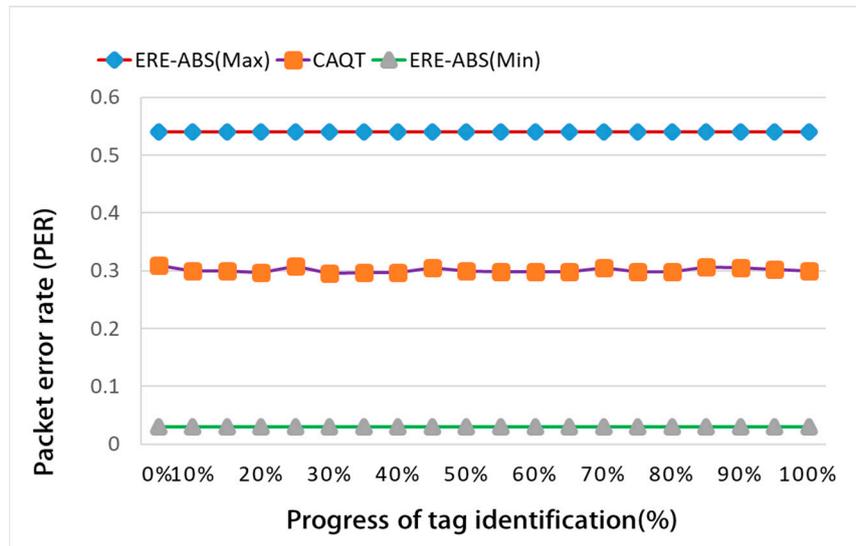


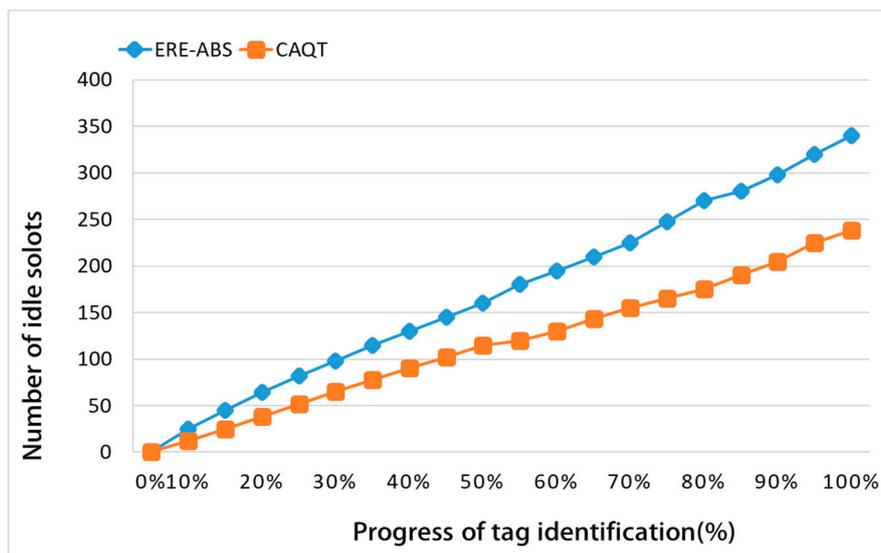
Figure 11. The effect of number of tags.

Figure 12a demonstrates the estimated PER during the tag identification progress while the number of tags was set to 500 and PER was set to 0.3. Since ERE-ABS only calculated the PER before the tag resolution process, the estimate PER would keep a value during the whole tag identification

procedure. Therefore, the max and min estimated PER during the 100 times identification process is used to compare the deviation. However, the CAQT could dynamically adjust PER according to the current channel condition, and then thus the estimated PER would slightly oscillate and approached the real PER, which was 0.3. Figure 12b also presents that the number of idle slots of CAQT was fewer than that of ERE-ABS during the whole tag identification procedure because CAQT could accurately estimate the PER while ERE-ABS could not.



(a) Estimated PER



(b) Number of idle slots

Figure 12. Estimated number of tags during the tag identification progress.

5. Conclusions and Future Work

This study proposes the CAQT algorithm to cope with the overestimated number of tags problem in the error-prone channel environment. The CAQT not only estimates current channel condition in order to estimate more accurate number of tags in the rapidly changing channel quality environment, but it also statistically determines the most efficient strategy, which is to ask the tag to retransmit it or to split the collide tags, in order to minimize the number of idle slots. The simulation results show that CAQT significantly outperforms the existing algorithms, which are ERE-ABS, QT, and DFSA. CAQT only used 69.0%, 57.8%, and 87.2% slots consumed by QT, DFSA, and ERE-ABS, respectively, when the

PER was 0.5. Therefore, the CAQT provides realistic description and a useful guideline of how things or people can be identified in varying channel quality of IoT environment, which contributes to the development of consumer electronics industry and literature. The simulation results also demonstrate that CAQT provides enhanced performance when the channel quality varies, especially in outdoor environments, for example, ticket checking for railway or subway system. We will extend our work to identify lots of tags in a short period of time duration in RFID systems [36].

Author Contributions: Conceptualization, Y.-H.C.; Data curation, L.-K.C.; Formal analysis, Y.-H.C.; Investigation, Y.-H.C.; Methodology, Y.-H.C.; Project administration, Y.-H.C.; Software, Y.-H.C. and R.-Z.H.; Supervision, Y.-H.C.; Writing—original draft, Y.-H.C. and Y.-R.S.; Writing—review and editing, Y.-H.C. and P.-T.J.

Acknowledgments: This work is supported by the Ministry of Science and Technology (MOST) (MOST 106-2218-E-227-001-MY2).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. ISO. *Information technology—Radio Frequency Identification for Item Management—Part 6: Parameters for Air Interface Communications at 860 MHz to 960 MHz, Amendment 1: Extension with Type C and Update of Types A and B. ISO/IEC 18000-6:2004/Amd. 1:(E)*; ISO: Geneva, Switzerland, 2006.
2. EPCglobal IncTM. *EPCTM Radio-Frequency Identity Protocols Class 1 Generation-2 UHF RFID Protocol for Communications at 860-960MHz Version 1.2.0*; EPCglobal Inc.: Brussels, Belgium, 2008.
3. Azambuja, M.; Marcon, C.A.M.; Hessel, F.P. Survey of standardized ISO 18000-6 RFID anti-collision protocols. In Proceedings of the Second International Conference on Sensor Technologies and Applications, Cap Esterel, France, 25–31 August 2008; pp. 468–473.
4. Cha, J.R.; Kim, J.H. Dynamic framed slotted ALOHA algorithms using fast tag estimation method for RFID system. In Proceedings of the IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 8–10 January 2006; pp. 768–772.
5. Vinod, N.; Lixin, G. Energy-aware tag anti-collision protocols for RFID systems. In Proceedings of the IEEE International Conference on Pervasive Computing and Communications, White Plains, NY, USA, 19–23 March 2007; pp. 33–36.
6. Park, J.; Lee, T.J. Error resilient estimation and adaptive binary selection for fast and reliable identification of RFID tags in error-prone channel. *IEEE Trans. Mob. Comput.* **2012**, *11*, 959–969. [[CrossRef](#)]
7. Shakiba, M.; Singh, M.J.; Islam, M.T.; Sundararajan, E. Applying cubic spline method to estimate the number of RFID tags in error-prone communication channels. *Wirel. Pers. Commun.* **2015**, *83*, 361–382. [[CrossRef](#)]
8. Zhang, L.; Gandino, F.; Ferrero, R.; Rebaudengo, M. Evaluation of the additive interference model for RFID reader collision problem. In Proceedings of the 2012 Fourth International EURASIP Workshop on RFID Technology, Torino, Italy, 27–28 September 2012.
9. Zhang, L.; Ferrero, R.; Gandino, F.; Rebaudengo, M. Evaluation of single and additive interference models for RFID collisions. *Math. Comput. Model.* **2013**, *58*, 1236–1248. [[CrossRef](#)]
10. Vogt, H. Efficient object identification with passive RFID tags. In Proceedings of the First International Conference on Pervasive Computing, Zürich, Switzerland, 26–28 August 2002; LNCS 2414, pp. 98–113.
11. Lei, Z.; Yum, T.S.P. The optimal reading strategy for EPC Gen-2 RFID anti-collision systems. *IEEE Trans. Commun.* **2010**, *58*, 2725–2733.
12. Maguire, Y.; Pappu, R. An optimal Q-Algorithm for the ISO 18000-6C RFID protocol. *IEEE Trans. Autom. Sci. Eng.* **2009**, *6*, 16–24. [[CrossRef](#)]
13. Lei, Z.; Yum, T.S.P. Optimal framed aloha based anti-collision algorithms for RFID systems. *IEEE Trans. Commun.* **2010**, *58*, 3583–3592.
14. Eom, J.B.; Lee, T.J.; Rietman, R.; Yener, A. An efficient framed-slotted aloha algorithm with pilot frame and binary selection for anti-collision of RFID tags. *IEEE Commun. Lett.* **2008**, *12*, 861–863. [[CrossRef](#)]
15. Chen, W.T. An accurate tag estimate method for improving the performance of an RFID anti-collision algorithm based on dynamic frame length ALOHA. *IEEE Trans. Autom. Sci. Eng.* **2009**, *6*, 9–15. [[CrossRef](#)]
16. Li, B.; Wang, J. Efficient anti-collision algorithm utilizing the capture effect for ISO 18000-6C RFID protocol. *IEEE Commun. Lett.* **2011**, *15*, 352–354. [[CrossRef](#)]

17. Park, J.; Chung, M.Y.; Lee, T.J. Identification of RFID tags in Framed-Slotted Aloha with robust estimation and binary selection. *IEEE Commun. Lett.* **2007**, *11*, 452–454. [[CrossRef](#)]
18. Lin, C.F.; Lin, F.Y.S. Efficient estimation and collision-group-based anti-collision algorithms for dynamic frame-slotted aloha in RFID networks. *IEEE Trans. Autom. Sci. Eng.* **2010**, *7*, 840–848. [[CrossRef](#)]
19. Yeh, M.K.; Jiang, J.R.; Huang, S.T. Parallel response query tree splitting for RFID tag anti-collision. In Proceedings of the 40th International Conference on Parallel Processing Workshops, Taipei City, Taiwan, 13–16 September 2011; pp. 6–15.
20. Myung, J.; Lee, W.; Shih, T.K. An adaptive memoryless protocol for RFID tag collision arbitration. *IEEE Trans. Multimed.* **2006**, *8*, 1096–1101. [[CrossRef](#)]
21. Choi, J.H.; Lee, D.; Lee, H. Query tree-based reservation for efficient RFID tag anti-collision. *IEEE Commun. Lett.* **2007**, *11*, 85–87. [[CrossRef](#)]
22. Gou, H.; Jeong, H.C.; Yoo, Y. A bit collision detection based query tree protocol for anti-collision in RFID system. In Proceedings of the 6th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Niagara Falls, ON, Canada, 11–12 October 2010; pp. 421–428.
23. Zein-Elabdeen, W.; Shaaban, E. An enhanced binary tree anti-collision technique for dynamically added tags in RFID systems. In Proceedings of the 2nd International Conference on Computer Engineering and Technology, Chengdu, China, 16–18 April 2010; pp. 349–353.
24. Lai, Y.C.; Lin, C.C. Two blocking algorithms on adaptive binary splitting: Single and pair resolutions for RFID tag identification. *IEEE/ACM Trans. Netw.* **2009**, *17*, 962–975.
25. Lai, Y.C.; Lin, C.C. Two couple-resolution blocking protocols on adaptive query splitting for RFID tag identification. *IEEE Trans. Mob. Comput.* **2011**, *11*, 1450–1463. [[CrossRef](#)]
26. Namboodiri, V.; Gao, L. Energy-aware tag anticollision protocols for RFID systems. *IEEE Trans. Mob. Comput.* **2010**, *9*, 44–59. [[CrossRef](#)]
27. Lai, Y.C.; Hsiao, L.Y. General binary tree protocol for coping with the capture effect in RFID tag identification. *IEEE Commun. Lett.* **2010**, *14*, 208–210. [[CrossRef](#)]
28. Yang, C.N.; He, J.Y. An effective 16-bit random number aided query tree algorithm for RFID tag anti-collision. *IEEE Commun. Lett.* **2011**, *15*, 539–541. [[CrossRef](#)]
29. Annur, R.; Jig, T.L.; Nakpeerayuth, S.; Wuttitsittikulij, L. Tree-based anti-collision algorithm with single error bit tracking for RFID systems. In Proceedings of the 2017 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 28–29 August 2017; pp. 25–28.
30. Hou, Y.; Zheng, Y. PHY-tree: Physical layer tree-based RFID identification. *IEEE/ACM Trans. Netw.* **2018**, *26*, 711–723. [[CrossRef](#)]
31. Sasiwat, Y. Bi-slotted fast query tree-based anti-collision algorithm for large scale of RFID systems. In Proceedings of the 10th International Conference on Knowledge and Smart Technology (KST), Chiang Mai, Thailand, 31 January–3 February 2018; pp. 221–224.
32. Shiha, D.H.; Suna, P.L.; Yenb, D.C.; Huangc, S.M. Taxonomy and survey of RFID anti-collision protocols. *Comput. Commun.* **2006**, *29*, 2150–2166. [[CrossRef](#)]
33. Yeh, M.K.; Jiang, J.R. Adaptive k-way splitting and pre-signaling for RFID tag anti-collision. In Proceedings of the 33rd Annual Conference of the IEEE Industrial Electronics Society, Taipei, Taiwan, 5–8 November 2007; pp. 40–45.
34. Yeh, M.K.; Jiang, J.R.; Huang, S.T. Adaptive splitting and pre-signaling for RFID tag anti-collision. *Comput. Commun.* **2009**, *32*, 862–1870. [[CrossRef](#)]
35. Su, X.H.; Liu, B.L. An investigation on tree-based tags anti-collision algorithms in RFID. In Proceedings of the International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, 23–25 September 2017; pp. 5–11.
36. Chen, W.-T. A Fast Anticollision algorithm with early adjustment of frame length for the EPCglobal UHF Class-1 Generation-2 RFID Standard. *Proc. Eng. Technol. Innov.* **2016**, *3*, 16–18. Available online: <http://ojs.ijeti.org/index.php/PETI/article/view/234> (accessed on 18 December 2018).

