Abstract—In RFID systems, the reader is unable to discriminate concurrently reported IDs of tags from the overlapped signals, and a collision happens. Many algorithms for anti-collision are proposed to improve the throughput and reduce the latency for tag identification. Existing anti-collision algorithms mainly employ CRC based collision detection functions for determining whether the collision happens. Generating CRC codes, however, requires complicated computations for both RF tags and readers, and hence incurs non-trivial time consumption, becoming the bottleneck. In this study, we design a Quick Collision Detection (QCD) scheme based on the bitwise complement function plus collision preamble, which significantly reduces the number of gates for computation and facilitates to simplify the IC design of RFID tags. The QCD scheme does not require any modification on upper-level air protocols, so it can be seamlessly adopted by current anti-collision algorithms. Through comprehensive analysis and simulations, we show that QCD improves the identification efficiency by 40%.

Keywords—RFID, Collision Detection, CRC, Bitwise Complement

I. INTRODUCTION

Radio Frequency Identification (RFID) has gained substantial attentions recently due to the adoption in many applications such as logistics, retails, asset management, access control [23], and health care. These RFID based systems typically comprise of a number of readers and tags. Attaching to objects or persons, tags can report their IDs to readers via RF signals. The RF based communication enables RFID based systems to identify or localize objects without keeping the tags in sight or touch, hereby facilitating the automatic identification and localization [1] [24] [28].

Identifying tags is like a challenge-response procedure. Within its detecting range, a reader broadcasts a frame, which comprises of a number of slots. Upon receiving the frame, each tag randomly selects a slot, and transmits its ID to the reader in that slot. If only one tag responds in a given slot, the reader can successfully receive the tag’s ID. Accordingly, such a slot is termed as a single slot. If none of tags responds in a slot, the slot is termed as idle slot. If there are more than one tag respond in a slot, the overlapped signals will cause a collision on the RF communication channel, which is called Tag Collision problem. Correspondingly, a slot with a collision is termed as collided slot. If a collision happens, each collided tag has to reselect a slot to transmit its ID in the next frame, which may cause delay on the identification procedure.

One attempt for solving Tag Collision is to extend the RF bandwidth. A larger RF bandwidth can provide more non-collided channels to tags. But physically extending RF bandwidths is not adopted by popular RFID standards due to the scarcity of available RF Spectrums [2] [3]. Another solution is to employ anti-collision algorithms. Anti-collision algorithms can be classified into two categories, viz. Framed Slotted ALOHA based (FSA) and Binary Tree based (BT). The key issue of anti-collision solutions is the identification efficiency, i.e., identifying all tags with minimum time duration. The identification efficiency of these algorithms depends on the efficiency of their collision detection functions, which are used for determining the types of slots. However, current detection functions used by RFID systems are inefficient due to the adoption of CRC code.

There are two major ways to establish a detection collision function. One is to utilize special hardware for sensing collisions in wireless channels. Designing such hardware, however, is costly and especially unaffordable to low-cost RFID tags [22]. The other approach is to utilize CRC as the collision detection. In a given slot, a tag emits its ID together with the tag’s ID’s CRC code. The reader can validate the ID and the code

![Figure 1. CRC-CD Scheme](image-url)
match, the slot is identified as a single slot. Otherwise, a collision happens in this slot. We term this method as CRC-CD, as illustrated in Figure 1. In the following, we denote crc(.) as the CRC operation. We also denote \( \lor \) as the signal overlapping, which can be abstracted as the bitwise Boolean sum [5] [6]. For example, the overlapping result of two tags’ ID signals is

\[
(011001) \lor (010010) = (011011)
\]

In Figure 1, the reader first computes

\[
crc(id_1 \lor id_2 \lor \cdots \lor id_m)
\]

and then determines the slot is a collided slot if the result is equivalent with

\[
crc(id_1) \lor crc(id_2) \lor \cdots \lor crc(id_m)
\]

If the slot is collided, the reader needs to launch another frame to identifying those collided tags, until all tags are successfully identified. Thus, none of the idle or collided slot is helpful to identify tags. In fact, the reader identifies tags only in single slots. Unfortunately, the throughput of the single slots, which is defined as the ratio of the number of single slots to the total number of slots in the identification procedure, is very low in most FSAs. For example, we will show that the throughput of FSAs cannot exceed 0.37 later in Lemma 1, indicating that existing anti-collision algorithms cannot identify any tag in around 63% slots.

The observation motivates us to revisit the solutions of tag collision problem. Clearly, if we are able to reduce the time consumed in determining those idle slots, the identification process will be remarkably improved. Adopting CRC-CD as the collision detection function, however, leads several shortcomings to current detection functions. First, CRC-CD is based on the cyclic redundancy check algorithm. The computing complexity of CRC is \( \mathcal{O}(l) \), where \( l \) is the length of ID. Second, CRC-CD requires more than 100 instructions for generating a CRC code, which is non-trivial for RFID tags and uses random numbers for splitting the tag sets. Third, the length of CRC codes is relatively long so that the communication overhead is high. For example, ISO 18000-6 (also compatible with EPC Gen 2) employs 32 bits CRC function. Indeed, CRC-CD becomes the barrier of collision detection functions.

Instead of using CRC-CD, we propose a Quick Collision Detection (QCD) scheme, to accelerate the collision detection process. Adopting bitwise complement as the collision detection function, QCD can significantly reduce the complexity of IC design as well as the number of gates required by RFID tags. We also leverage a collision preamble to further speed up the collision detection process. Our theoretical analysis and experimental results show that QCD can save more than 40% time for both FSA and BS anti-algorithms during the tag identification.

The rest of this paper is organized as follows. We discuss the related works in Section II and revisit the efficiency of existing anti-collision algorithms in Section III. Then formally define the collision detection problem and present the design of QCD in Section IV. We theoretically analyze the performance of QCD and compare QCD with CRC-CD in details in Section V. Last, we extensively evaluate QCD in Section VI and conclude the paper in Section VII.

II. RELATED WORKS

In the literature, the works related to the collision detection comprise of two categories, Framed Slotted ALOHA based and Binary Tree based algorithms.

**Framed Slotted ALOHA (FSA) based algorithms:** Roberts [17] first proposes an ALOHA-based anti-collision scheme for RFID identification. Lee [8] finds that the reader obtains a maximum identification throughput within its scanning field when the size of frame equals to the number of tags. Lee also leverages his observation to propose a dynamic FSA, which improves the throughput by adaptively tuning the length of current frame based on the number of collided slots reported from the previous frame. Similarly, EPC Gen2 [2] adopts a variation of FSA, ‘Q-Adaptive’, which also adaptively adjusts the frame length according to the type of last slot. If the last slot is idle or collided, the reader will end the current frame immediately and launch a new detecting frame. The length of new frame will be shorter than the current frame if the last slot is an idle one, otherwise it will be longer if the last slot is collided.

**Binary Tree (BT) based algorithms:** The binary tree (BT) based RFID identification protocols has been adopted by another well-known RFID air protocol, ISO 18000-6 [3]. Hush and Wood [12] analyze the throughput of BT based algorithms by using the conclusion from [11]. Myung and Lee [9] [10] propose an Adaptive Binary Splitting (ABS) protocol to reduce collisions and identify tags efficiently. ABS starts the tag identification only from readable cycles and uses random numbers for splitting the tag sets. ABS achieves a quick identification by eliminating unnecessary cycles. In particular, researchers develop Query Tree (QT) based anti-collision algorithms to resolve the ‘starvation problem’ that may occur in both FSAs and BTs [18] [19]. QT based protocols distributes tags in a binary tree according to their IDs. The reader broadcasts a query with a bit string prefix \((q_1 q_2 \cdots q_x)\). The tag which has a match prefix in its ID responds the query with it’s ID. If the responses collide, the reader appends one bit to prefix, \((q_1 q_2 \cdots q_x 0)\) or \((q_1 q_2 \cdots q_x 1)\) in the next slot. Then the tags with the prefix \((q_1 q_2 \cdots q_x)\) are further spitted into two sets. The process continues until only one tag responds. In this way, each tag can be recursively distinguished. Myung and Lee [9] [10] presents an Adaptive Query Splitting (AQS), which is an advanced version of the QT protocol. Since each tag can be deterministically identified, QT resolves the starvation problem. The reader first computes

\[
crc(id_1 \lor id_2 \lor \cdots \lor id_m)
\]

and then determines the slot is a collided slot if the result is equivalent with

\[
crc(id_1) \lor crc(id_2) \lor \cdots \lor crc(id_m)
\]

If the slot is collided, the reader needs to launch another frame to identifying those collided tags, until all tags are successfully identified. Thus, none of the idle or collided slot is helpful to identify tags. In fact, the reader identifies tags only in single slots. Unfortunately, the throughput of the single slots, which is defined as the ratio of the number of single slots to the total number of slots in the identification procedure, is very low in most FSAs. For example, we will show that the throughput of FSAs cannot exceed 0.37 later in Lemma 1, indicating that existing anti-collision algorithms cannot identify any tag in around 63% slots.

The observation motivates us to revisit the solutions of tag collision problem. Clearly, if we are able to reduce the time consumed in determining those idle slots, the identification process will be remarkably improved. Adopting CRC-CD as the collision detection function, however, leads several shortcomings to current detection functions. First, CRC-CD is based on the cyclic redundancy check algorithm. The computing complexity of CRC is \( \mathcal{O}(l) \), where \( l \) is the length of ID. Second, CRC-CD requires more than 100 instructions for generating a CRC code, which is non-trivial for RFID tags and uses random numbers for splitting the tag sets. Third, the length of CRC codes is relatively long so that the communication overhead is high. For example, ISO 18000-6 (also compatible with EPC Gen 2) employs 32 bits CRC function. Indeed, CRC-CD becomes the barrier of collision detection functions.

Instead of using CRC-CD, we propose a Quick Collision Detection (QCD) scheme, to accelerate the collision detection process. Adopting bitwise complement as the collision detection function, QCD can significantly reduce the complexity of IC design as well as the number of gates required by RFID tags. We also leverage a collision preamble to further speed up the collision detection process. Our theoretical analysis and experimental results show that QCD can save more than 40% time for both FSA and BS anti-algorithms during the tag identification.

The rest of this paper is organized as follows. We discuss the related works in Section II and revisit the efficiency of existing anti-collision algorithms in Section III. Then formally define the collision detection problem and present the design of QCD in Section IV. We theoretically analyze the performance of QCD and compare QCD with CRC-CD in details in Section V. Last, we extensively evaluate QCD in Section VI and conclude the paper in Section VII.

II. RELATED WORKS

In the literature, the works related to the collision detection comprise of two categories, *Framed Slotted ALOHA* based and *Binary Tree* based algorithms.

**Framed Slotted ALOHA (FSA) based algorithms:** Roberts [17] first proposes an ALOHA-based anti-collision scheme for RFID identification. Lee [8] finds that the reader obtains a maximum identification throughput within its scanning field when the size of frame equals to the number of tags. Lee also leverages his observation to propose a dynamic FSA, which improves the throughput by adaptively tuning the length of current frame based on the number of collided slots reported from the previous frame. Similarly, EPC Gen2 [2] adopts a variation of FSA, ‘Q-Adaptive’, which also adaptively adjusts the frame length according to the type of last slot. If the last slot is idle or collided, the reader will end the current frame immediately and launch a new detecting frame. The length of new frame will be shorter than the current frame if the last slot is an idle one, otherwise it will be longer if the last slot is collided.

**Binary Tree (BT) based algorithms:** The binary tree (BT) based RFID identification protocols has been adopted by another well-known RFID air protocol, ISO 18000-6 [3]. Hush and Wood [12] analyze the throughput of BT based algorithms by using the conclusion from [11]. Myung and Lee [9] [10] propose an *Adaptive Binary Splitting (ABS)* protocol to reduce collisions and identify tags efficiently. ABS starts the tag identification only from readable cycles and uses random numbers for splitting the tag sets. ABS achieves a quick identification by eliminating unnecessary cycles. In particular, researchers develop Query Tree (QT) based anti-collision algorithms to resolve the ‘starvation problem’ that may occur in both FSAs and BTs [18] [19]. QT based protocols distributes tags in a binary tree according to their IDs. The reader broadcasts a query with a bit string prefix \((q_1 q_2 \cdots q_x)\). The tag which has a match prefix in its ID responds the query with it’s ID. If the responses collide, the reader appends one bit to prefix, \((q_1 q_2 \cdots q_x 0)\) or \((q_1 q_2 \cdots q_x 1)\) in the next slot. Then the tags with the prefix \((q_1 q_2 \cdots q_x)\) are further spitted into two sets. The process continues until only one tag responds. In this way, each tag can be recursively distinguished. Myung and Lee [9] [10] presents an *Adaptive Query Splitting (AQS)*, which is an advanced version of the QT protocol. Since each tag can be deterministically identified, QT resolves the starvation problem.
problem, in which a specific tag may not be identified for a long time. Unfortunately, QT based approaches suffer from malicious interfering. When a ‘malicious’ tag keeps responding, QT fails to identify any tag. However, a bad thing can be turned into a good one. In [20], the authors develop such a kind of ‘malicious’ tags, called ‘blocker tags’, to selectively protect consumer’s privacy.

**Reader-Tag and Reader-Reader collisions:** Besides the tag-tag collisions, there are other two types of collisions in multi-reader environments: Reader-Tag collision and Reader-Reader collision. In a recent work [21] [25], the authors analyze these two types of collisions. When a reader A is within another reader B’s scanning field, the response from tags targeted at A will be ‘drown’ by B’s signals. This collision is defined as Reader-Tag collision, which can be assigned by addressing different channels to adjacent readers, or scheduling their interrogations into different slots. If a region is overlapped by two readers’ scanning signals, the tags within this region cannot differentiate the signals simultaneously emitted from two readers. Such a collision is called Reader-Reader collision. The effective way to address the Reader-Reader collision is to avoid activating two readers at the same time. In our work, the term of collision only refers to tag-tag collisions. We assume that there are no collisions of other two types.

**Bitwise boolean sum model:** Recently, the bitwise Boolean sum model is widely used in the design of RFID security protocols. Choi and Roh [6] observe that the strength of signals through the forward channel, i.e., from the reader to tags, should be stronger than that of the backward channel, i.e., from tags to the reader. In QT based algorithms, since the reader utilized queries with increasing prefix via the forward channel, if some eavesdroppers within the channel hear the query, they can retrieve the tag’s ID. Therefore, the authors in [5] propose a backward channel based protection method. The main idea is to allow the reader to send a randomly generated pseudo-ID. The pseudo-ID, mixed with the tag’s real ID through a bitwise Boolean sum operation, is sent to the reader. The reader utilizes the pseudo-ID to resolve the real ID from the overlapped signals. Since the eavesdropper lacks the knowledge of pseudo-IDs, it cannot know the real ID. Lim et al [5] also focus on the backward channel based protection. They propose a randomized bit-encoding scheme to strength the privacy for RFID tags and alleviate the ‘same-bit’ problem, in which some bits of the ID could be disclosed. They also define an entropy-based metric to effectively measure the performance of privacy protection.

### III. Efficiency of Existing Anti-collision Algorithms

To elaborate the shortcoming of CRC-CD based collision detection functions, we revisit the principles and efficiency of FSA and BT based anti-algorithms. For ease of exploration, we summarize the main notions in Table I.

#### Table I

<table>
<thead>
<tr>
<th>Notion</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{F}$</td>
<td>The frame length</td>
</tr>
<tr>
<td>$n$</td>
<td>The number of tags within the reader’s detecting range</td>
</tr>
<tr>
<td>$m$</td>
<td>The number of tags transmitting IDs simultaneously in one slot</td>
</tr>
<tr>
<td>$\vee$</td>
<td>Bitwise boolean sum operation</td>
</tr>
<tr>
<td>$s$</td>
<td>Signal sent by the $i$-th tag</td>
</tr>
<tr>
<td>$s_i$</td>
<td>Signal received by reader</td>
</tr>
<tr>
<td>$c$</td>
<td>Checksum</td>
</tr>
<tr>
<td>$\oplus$</td>
<td>Concatenation operation</td>
</tr>
<tr>
<td>$c_{\text{crc}}$</td>
<td>CRC operation</td>
</tr>
<tr>
<td>$r$</td>
<td>Random number (a positive integer)</td>
</tr>
<tr>
<td>$\bar{s}$</td>
<td>Bitwise complement operation</td>
</tr>
<tr>
<td>$l_{\text{id}}$, $l_{\text{prm}}$, $l_{\text{crc}}$</td>
<td>The length of tag’s ID, preamble, and CRC code</td>
</tr>
<tr>
<td>$N_0, N_1, N_c$</td>
<td>The number of idle slots, single slots, and collided slots</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>The throughput of anti-collision algorithm</td>
</tr>
</tbody>
</table>

#### A. Framed Slotted ALOHA based algorithms

Framed Slotted ALOHA (FSA) based algorithms [2] [3] [7] [8] employ a randomized method to reduce the collision probability. In a FSA algorithm, the reader divides a detecting frame into $\mathcal{F}$ slots. Each tag randomly selects a slot in the frame for transmitting its ID. In a given slot, multiple tags may transmit their ID simultaneously and thereby yield a collision. In this case, each collided tag will transmit its ID in a randomly chosen slot in the next frame. This procedure continues until all tags have been successfully identified.

Before showing the efficiency of FSAs, we introduce several necessary definitions. In the $i$-th slot of a detecting frame of FSA, we denote the random variable $X_i = 1$ as the event that $\text{NONE}$ of tags responds, $Y_i = 1$ as the event that only one tag transmits its ID, and $Z_i = 1$ as the event that a collision happens. Note that $X_i + Y_i + Z_i = 1$ for any slot $i$ in the detecting frame. Let $N_0 = \sum_{i=1}^{\mathcal{F}} X_i$ to denote the total number of idle slots, $N_1 = \sum_{i=1}^{\mathcal{F}} Y_i$ to denote the total number of single slots, and $N_c = \mathcal{F} - N_0 - N_1$ to denote the total number of collision slots. We define the throughput $\lambda$ of FSA as

$$\lambda = \frac{N_1}{N_0 + N_1 + N_c}$$

**Lemma 1:** In a detecting frame with $\mathcal{F}$ slots, if $n \approx \mathcal{F}$, the maximum throughput of FSA is given by $\lambda_{\text{max}} \approx 0.37$, where $n$ denotes the total number of tags.

**Proof:** Since

$$N_1 = \sum_{i=1}^{\mathcal{F}} Y_i = \mathcal{F} \binom{n}{1} \left( \frac{1}{2} \right) \left( 1 - \frac{1}{2} \right)^{n-1}$$

$$\approx ne^{-n/\mathcal{F}}$$
, we have
\[
\lambda = \frac{N_1}{N_0 + N_1 + N_c} = \left(\frac{n}{\mathcal{F}}\right) e^{-n/\mathcal{F}}
\]
We can achieve the maximum throughput by computing the partial derivative of \( \lambda \) with respect to \( \mathcal{F} \).
\[
\frac{\partial \lambda}{\partial \mathcal{F}} = -\frac{n}{\mathcal{F}^2} e^{-n/\mathcal{F}} + \frac{n^2}{\mathcal{F}^3} e^{-n/\mathcal{F}} = 0
\]
Therefore, the optimal length of frame is \( \mathcal{F} = n \), and the maximum throughput is 0.37.
\[
\lambda_{max} = \frac{1}{e} \approx 0.37
\]

From Lemma 1, we observe that only 37% slots are used by FSAs to successfully identify tags, while more than 63% slots are underutilized in the entire identification procedure. In addition, FSAs suffer from a potential flaw, ‘tag starvation problem’ [9], where some specific tags are unable to complete the identification for a long time if they always collide with others.

B. Binary Tree based algorithms

The Binary Tree (BT) based anti-collision algorithms [3] [10] employ a virtual binary tree to organize the IDs of tags. For identifying a tag, the reader launches a slotted identification procedure and recursively probe the tree from the root to leaves. Every tag owns a counter, in which the value is initialized as 0. In each slot, a tag transmits its ID if and only if the value of its counter equals to 0. At the very beginning, all tags transmit their ID concurrently. After each slot, the reader claims the type of this slot, i.e., the slot is idle, single, or collided. According to the reader’s report, each tag changes its counter. If a collision happens in the previous slot, the tags which are involved in the collision randomly select 0 or 1, and add the number to the counter.

The other tags which are not involved in collision directly increase their counter by 1. Consequently, the entire tag set is split into two subsets. In one subset, each tag’s counter is 0. In the other subset, each tag’s counter is equals or greater than 1. In a non-collided slot, all tags decrease their counter by 1. The tags that have been identified keep silent until the identification process terminates. A BT based anti-collision algorithm is illustrated in Figure 2. We examine the average throughput of BT via Lemma 2.

**Lemma 2:** For identifying \( n \) tags using BT based algorithms, the average total number of needed slots is 2.885\( n \), including 1.443\( n \) collided slots, 0.442\( n \) idle slots, and \( n \) single slots. The average throughput \( \lambda_{avg} = 0.35 \).

**Proof:** Borrowing the conclusion from [11] [12], the average number of collided slots is 1.443\( n \), and the average number of idle slots is 0.442\( n \) in the whole process. Hence, we obtain the average throughput of BT as
\[
\lambda_{avg} = \frac{n}{1.443n + 0.442n + n} = 0.35
\]

Base on Lemma 1 and 2, we can find that both FSAs and BTs can only use 35% - 37% slots to successfully identify tags, while more than 60% slots are not utilized. If we can shorten the time consumed for identifying the types of unused slots, we can improve the efficiency of anti-collision algorithms significantly.

IV. QUICK COLLISION DETECTION

In this section, we formulate the collision detection problem. We then present our collision detection methodology, Quick Collision Detection (QCD) scheme.

A. Problem Formulation

When a collision occurs, the physical signals emitted by multiple tags are overlapped, so that they are indistinguishable for identifying the tags. Indeed, the overlapped signals can be considered as a bitwise Boolean sum. Given that there are \( m \) tags selecting a given slot \( t \), the final signal received by the reader is \( s = s_1 \lor s_2 \lor \cdots \lor s_m = \lor_{i=1}^{m} s_i \) and \( |s| = |s_1| = \cdots = |s_m| \), where \( s \) denotes the final signal received by the reader, \( s_i \) denotes the signal sent by the \( i \)-th tag, \( \lor \) represents the bitwise Boolean sum operation, and \( |s| \) denotes the length of the signal.

Existing approaches mainly employ CRC-CD for collision detection. For example, according to the well-known RFID standard, EPC Class-1 Gen-2 [2], a tag transmits its EPC ID (64 bits) as well as a CRC code (32 bits) to the reader in a given slot. The reader then computes a CRC code of the received ID and compares the result with the received CRC code. If they are not match, a collision happens.

Otherwise, there is no collision. As shown in Figure 4, solving the collision detection problem through CRC-CD algorithm is equivalent to deciding whether the value of \( \text{crc}(\lor_{i=1}^{m} id_i) \) is equal to the value of \( \lor_{i=1}^{m} \text{crc} \). According to the analysis in [4], the error of CRC is \( 2^{-r} \), where the \( r \) is the strength of CRC. For example, the error of CRC-32 is \( 1/2^{32} \). Such an error is negligible for practical applications. However, CRC is a sophisticated error-detection and error-correction technique for the capacity-limited RFID tag. CRC requires the tag to allocate relatively large computing
resource, while some functions of CRC, for example the error correction, are not necessary for detecting collisions.

For improving the collision detection, we aim to detect all collisions as quickly as possible with low computation complexity. We take this objective to guide the design of our detection scheme. Different from CRC-CD based approaches, we allow each tag to send a collision preamble before transmitting its ID. The preamble comprises a random positive integer \( r_i \) and an additional checking code \( c_i \). Namely, the collision preamble is \( r_i \oplus c_i \), where the \( \oplus \) denotes the concatenation operation. Therefore, if there are \( m > m_\geq \) positive integers \( r_i \), then the final signal received by the reader is \( s = r \oplus c \), where \( r = \lor i=1 \lor m \) and \( c = \lor i=1 \lor m \). We define the function \( f(r_i) = c_i \) as a collision function and elaborate its definition as follows:

**Definition 1 (Collision function):** Given that a set of positive integers \( R = \{ r_1, r_2, \ldots, r_m \} \), where \( m \geq 1 \), and there are at least two elements are not equal in \( R \) when \( m > 1 \), \( f(x) \) is a collision function if it meets the conditions that \( m > 1 \) if and only if \( f(\lor i=1 \lor m r_i) \neq \lor i=1 \lor m f(r_i) \).

With above definition, collision detection problem can be formalized as follows. Suppose that there are \( m (m \geq 1) \) tags choosing a given slot to transmit their IDs. In this slot, each of these \( m \) tags selects a random positive integer \( r_i \) and transmits \( r_i \oplus f(r_i) \). We assume that if \( m > 1 \), there are at least two tags’ random integers are different. Based on Definition 1, the collision detection problem is equivalent to determining whether \( f(\lor i=1 \lor m r_i) \) is equal to \( \lor i=1 \lor m f(r_i) \). If \( f(\lor i=1 \lor m r_i) = \lor i=1 \lor m f(r_i) \), there is only one tag replying in the slot, i.e., \( m = 1 \). Otherwise, a collision happens and hence \( m > 1 \). Figure 4 illustrates the situation when a collision happens.

Based on the received \( s \), the reader can distinguish the type of given slot and perform operations accordingly. If the slot is idle or collided, the reader moves to next slot. If the slot is ‘single’, the reader commands the tag to report ID. Figure 3 shows the tag identification procedure. Note that in QCD, the length of a single slot is different from that of an idle or collided slot. Moreover, tags only transmit their collision preambles in idle or collided slots in QCD. In contrast, previous approaches require each tag to transmit the ID and CRC code in any types of slots. Therefore, the variable-length mechanism of QCD can reduce the time consumed for the transmission in both idle and collided slots.

B. Quick Collision Detection

To improve the detecting efficiency, we aim to seek a simple and fast collision detection function \( f(r) \) for QCD. There are a large number of collision functions in the literature. Among them, we find that the function \( f(r) = \overline{r} \) fulfills our requirements, where \( \overline{r} \) denotes the bitwise complement operation. We prove the feasibility and correctness of this selection as follows.

**Theorem 1:** if \( f(r) = \overline{r} \) is a collision function, where \( \overline{r} \) is bitwise complement of \( r \).

**Proof:** Given a positive integer set \( R = \{ r_1, r_2, \ldots, r_m \} \) and \( m \geq 1 \). If \( m > 1 \), we assume that there are at least two different elements in \( R \). We need to prove the following two claims:

1) \( m > 1 \Rightarrow f(\lor i=1 \lor m r_i) \neq \lor i=1 \lor m f(r_i) \)

Since \( m > 1 \), we assume that \( r_i \neq r_j \), where \( r_i, r_j \in R \). Hence, there must exist a \( k, 1 \leq k \leq |r_i| \), such that the \( k \)-th bit in \( r_i \) is not equal to that in \( r_j \), namely \( r_k \neq r_j \). Since \( r_k \neq r_j \), there is at least two elements are not equal in \( R \). Therefore, \( f(r_i) \neq f(r_j) \). According to the principle of Boolean sum [13], the Boolean sum operation on each bit is independent with those on other bits. Thus, we have \( (r_i \lor r_j)^k = 1 \Rightarrow (\lor i=1 \lor m r_i)^k = 1 \). The bitwise complement on one bit is also independent with those on other bits, thus

\[
(f(\lor i=1 \lor m r_i))^k = (\lor i=1 \lor m f(r_i))^k = 0
\]
Algorithm 1 Quick Collision Detection

**Input:** final signal \( s \) and slot \( t \)

**Output:** The type of slot \( t \):
- 0 - represents idle slot
- 1 - represents single slot
- 2 - represents collided slot

1: The reader receives final signal \( s \)
2: if \( s = 0 \) then
   3: return 0
4: else
5: The reader retrieves \( r \) and \( c \) from \( s \)
6: if \( c = f(r) \) then
5: return 1
8: else
9: return 2
10: end if
11: end if

On the other hand,

\[
\begin{align*}
    r_i^k &\neq r_j^k \Rightarrow (\bar{r}_i)^k \neq (\bar{r}_j)^k \\
    \Rightarrow (f(r_i))^k &\neq (f(r_j))^k \\
    \Rightarrow (f(r_i))^k \lor (f(r_j))^k &\neq 1
\end{align*}
\]

Thus,

\[
(\vee_{i=1}^m f(r_i))^k = 1
\]

We have \((f(\vee_{i=1}^m r_i))^k = 0\) and \((\vee_{i=1}^m f(r_i))^k = 1\), which indicates that the \( k \)-th bit in \( f(\vee_{i=1}^m r_i) \) is different from the \( k \)-th bit in \( \vee_{i=1}^m f(r_i) \). Therefore,

\[
f(\vee_{i=1}^m r_i) \neq \vee_{i=1}^m f(r_i)
\]

2) \( f(\vee_{i=1}^m r_i) \neq \vee_{i=1}^m f(r_i) \Rightarrow m > 1 \)

Suppose \( m = 1 \), we have \( f(\vee_{i=1}^m r_i) = f(r_i) = \bar{r}_i \) and \( \vee_{i=1}^m f(r_i) = \vee_{i=1}^m \bar{r}_i = \bar{r}_i \). This is a contradiction with our assumption. Hence, \( m > 1 \).

In summary, \( f(r) = \bar{r} \) is a collision function.

V. Efficiency Analysis

QCD significantly improves the efficiency for anticollision algorithms on twofold. First, utilizing bitwise complement operation saves a large amount of time consumed on collision detection. Second, the communication latency of QCD can be dramatically reduced by using the preamble mechanism. In this section, we theoretically analyze the improvement made by QCD on FSA and BT based approaches. We also compare QCD with CRC-CD in terms of complexity and overhead.

For ease of exploration, we assume that the time for transmitting one bit is \( \tau \), the length of ID is \( l_{id} \)-bits, the length of CRC codes is \( l_{crc} \)-bits, and the length of collision preambles is \( l_{prm} \)-bits.

A. Improvement on FSA

According to Lemma 1, the maximum throughput of FSA is 37%. Therefore, the minimum total number of slots for identifying \( n \) tags is \( n/0.37 = 2.7n \). The transmission time is \( t_{crc} = 2.7n\tau(l_{id} + l_{crc}) \) if using CRC-CD. For QCD, the time is \( t_{qcd} = n\tau(l_{prm} + l_{id}) + 1.7n\tau l_{prm} \). Compared to CRC-CD, the minimum efficiency improvement made by QCD, denoted as \( EI \), is defined as follows.

\[
EI = \frac{t_{crc} - t_{qcd}}{t_{crc}} = \frac{0.6293l_{id} + l_{crc} + l_{prm}}{l_{crc} + l_{id}}
\]

Following the specification of EPC [2], we adopt \( l_{id} = 64 \) and \( l_{crc} = 32 \). We theoretically summarize the minimum efficiency improvement on FSA based approaches with different strength of QCD in Table II. For example, when the strength of QCD is 8-bit, QCD improves the efficiency of identification process for FSAs up to 58.64%.

<table>
<thead>
<tr>
<th>Strength of QCD</th>
<th>EI</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-bit</td>
<td>\geq 0.6698</td>
</tr>
<tr>
<td>8-bit</td>
<td>\geq 0.5864</td>
</tr>
<tr>
<td>16-bit</td>
<td>\geq 0.4198</td>
</tr>
</tbody>
</table>

B. Comparison of CRC-CD and QCD

According to [2], the minimum number of slots needed by CRC-CD to identify \( n \) tags is \( n/0.63 = 1.597n \). The overall transmission time is \( t_{crc} = 1.597n\tau(l_{id} + l_{crc}) \). For QCD, the transmission time is \( t_{qcd} = n\tau(l_{prm} + l_{id}) + 1.7n\tau l_{prm} \). Compared to CRC-CD, the minimum efficiency improvement made by QCD, denoted as \( EI \), is defined as follows.

\[
EI = \frac{t_{crc} - t_{qcd}}{t_{crc}} = \frac{0.6293l_{id} + l_{crc} + l_{prm}}{l_{crc} + l_{id}}
\]

We also compare QCD with CRC-CD in terms of complexity and overhead.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Complexity</th>
<th>Memory</th>
<th>Transmission</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRC-CD</td>
<td>( \mathcal{O}(l) )</td>
<td>1KB</td>
<td>96bits</td>
</tr>
<tr>
<td>QCD</td>
<td>( \mathcal{O}(1) )</td>
<td>16bits</td>
<td>16bits</td>
</tr>
</tbody>
</table>

Table IV

<table>
<thead>
<tr>
<th># of instruction</th>
<th>CRC-CD</th>
<th>QCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>More than 100 instructions</td>
<td>( \mathcal{O}(l) )</td>
<td>( \mathcal{O}(1) )</td>
</tr>
<tr>
<td>Memory</td>
<td>1KB</td>
<td>16bits</td>
</tr>
<tr>
<td>Transmission</td>
<td>96bits</td>
<td>16bits</td>
</tr>
</tbody>
</table>
### Table V
SIMULATION SETUP

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation Area</td>
<td>100m × 100m</td>
</tr>
<tr>
<td>Number of readers</td>
<td>100</td>
</tr>
<tr>
<td>Identification range of the reader</td>
<td>3m</td>
</tr>
<tr>
<td>Tag ID</td>
<td>Randomly selected 96-bit ID</td>
</tr>
</tbody>
</table>

### Table VI
SIMULATION CASES

<table>
<thead>
<tr>
<th>Case</th>
<th># of tags</th>
<th># of slots</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>50</td>
<td>30</td>
</tr>
<tr>
<td>II</td>
<td>500</td>
<td>300</td>
</tr>
<tr>
<td>III</td>
<td>5000</td>
<td>3000</td>
</tr>
<tr>
<td>IV</td>
<td>5000</td>
<td>30000</td>
</tr>
</tbody>
</table>

### B. Improvement on BT

According to Lemma 2, the average throughput of BT based approaches is 0.35. Therefore, the average total number of slots for identifying $n$ tags is $n/0.35 = 2.885n$. If using CRC-CD, the communication time is $2.885n(l_{id} + l_{crc})\tau$. In contrast, the communication time is $1.885nl_{prm}\tau + n(l_{prm} + l_{id})\tau$ if using QCD. If $l_{id} = 64$ bits and $l_{crc} = 32$ bits, the average $EI$ is given as follows.

$$EI = \frac{t_{crc} - t_{qcd}}{t_{crc}} = \frac{0.6533l_{id} + l_{crc} - l_{prm}}{l_{crc} + l_{id}}$$

We summarize the improvement on BT based approaches in Table III. Especially, if adopting 8-bit strength, QCD can contribute 60.23% efficiency improvement.

### C. CRC-CD vs. QCD

QCD outperforms CRC-CD due to the following advantages. First, CRC-CD is based on cyclic redundancy check algorithm whose complexity is $O(l)$, where $l$ is the length of ID, while the complexity of QCD’s bitwise complement function is $O(1)$. Second, a CRC-CD operation requires more than 100 CPU instructions while QCD only needs 1 instruction in the checksum computation. Third, CRC-CD based approaches need to transmit 96-bits CRC codes in both idle and collided slots, while QCD only needs to transmit 16-bits codes for detecting collisions. Finally, CRC-CD requires 1KB extra memory for containing the lookup table, but the bitwise complement function only requires 16 bits to store the signal. We also compare QCD with CRC-CD based approaches in terms of complexity and overhead in Table IV.

### VI. PERFORMANCE EVALUATION

In this section, we evaluate our design via comprehensive simulations. Our evaluation focuses on four metrics: accuracy, delay, utilization rate, and efficiency improvements both on FSA and BT.
C. Utilization Rate

The second metric is the Utilization Rate (UR). This parameter is defined as the ratio of the time consumed for transmitting IDs of tags to the total time of identification. This parameter shows 'effective time' we spend to successfully identify tags. We define the UR of QCD as

\[ UR = \frac{N_1 l_{id}\tau}{N_1 (l_{prm} + l_{id}) + (N_c + N_0) l_{prm}\tau} \]

In fact, UR reflects the throughput of anti-collision algorithms. A higher UR contributes a larger throughput of successfully identified tags. A high strength, however, leads to a low throughput of QCD. To elaborate the tradeoff, we check the UR of QCD and show the result in Table IX. From the table, we can observe that when the UR decreases, the strength of QCD increases. In particular, if we employ 16-bit as the strength, the UR of QCD dramatically drops to below 50% in all cases.

Tables VII and VIII show the distribution of slots and throughput when deploying QCD to the FSAs and BTs, respectively. In case I, we employ 50 tags and set the frame size as 30 slots. As a result, a FSA based algorithm may totally need 119 slots in average, including 39 idle slots, 50 single slots, and 110 collided slots. The throughput of FSAs in case I, II, III, and IV are 25%, 22%, 20%, and 20%, respectively. Note that the throughput is below the upper bound, i.e., 37% as we discussed in Section III, because the optimal frame size is not employed in our simulation. In practice, the reader cannot exactly know the number of tags in advance. Therefore, it is difficult to set the frame length as the optimal one. Detail discussion about the optimal frame size can be found in [8], [14]–[16].

Moreover, we find that the throughput of BTs is around 35%, which demonstrates the correctness of Lemma 2. Meanwhile, this observation also validates our assumption that the majority part of the identification process is consumed for dealing with collided or empty slots.

Combining above observations, we suggest taking 8-bits as the strength of QCD in practice, which is able to achieve a good balance between the accuracy and throughput of successfully identified tags.

D. Identification Delay

Fast identification is the most significant factor in the mobile tag environment. The tag may move out of the reader's range before it is identified by the reader if the identification is slow. We define the identification delay of tag \( t_i \) as the interval between the start of identification and the time when the tag is identified. We utilize the average delay to understand the relationship between the delay and collision detection. The average delay is computed as

\[ D_{avg} = \frac{\sum_{t_i \in T} D_{t_i}}{|T|} \]

where \( D_{t_i} \) is the delay of tag \( t_i \) and \( T \) is the set of tags.

Figure 6 presents the average delay of CRC-CD (8-bit strength) and QCD. Evident from the graph, QCD significantly reduces the identification delay more than 80% in four cases. Specially, the \( D_{avg} \) of QCD more sharply concentrate around the mean, which indicates QCD is more stable than CRC-CD.

E. Efficiency Improvements

The last important metric is the efficiency improvement in terms of transmission latency. Leveraging the definition in Section V-A, we examine the EI of FSAs and BTs, respectively.

We adopt QCD to FSAs and BTs to compare the performance of QCD with that of CRC-CD. Figure 7(a) presents the comparison on the time consumption between CRC-CD based FSAs and QCD (8-bit strength) based FSAs. We observe that QCD based FSAs spend less than half of transmission time of CRC-CD based FSAs in all cases. When the number of tags increases, the difference also drastically enlarges.

Figure 7(b) shows the comparison on the time consumption between CRC-CD based BTs and QCD based BTs (with the 8-bits strength). The result also indicates QCD can significantly improve the latency of identification.

We show the EI of QCD based FSA in Figure 8(a). If setting the strength as 8-bits, QCD base FSAs shorten the time cost to 65%, 68%, 69%, and 70% of that used by CRC-CD based FSAs in case I, II, III, and IV, respectively. The values of EI in those four cases are all larger than the theoretical lower bound (41.98%). Especially, when we

<table>
<thead>
<tr>
<th>Case</th>
<th># of frame</th>
<th># of idle slots</th>
<th># of single slots</th>
<th># of collided slots</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>6</td>
<td>39</td>
<td>50</td>
<td>110</td>
<td>0.25</td>
</tr>
<tr>
<td>500</td>
<td>7</td>
<td>1376</td>
<td>500</td>
<td>394</td>
<td>0.22</td>
</tr>
<tr>
<td>5000</td>
<td>8</td>
<td>15217</td>
<td>5000</td>
<td>3962</td>
<td>0.20</td>
</tr>
<tr>
<td>50000</td>
<td>8</td>
<td>164477</td>
<td>50000</td>
<td>39622</td>
<td>0.20</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th># of frame</th>
<th># of idle slots</th>
<th># of single slots</th>
<th># of collided slots</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>137</td>
<td>19</td>
<td>50</td>
<td>68</td>
<td>0.20</td>
</tr>
<tr>
<td>500</td>
<td>1426</td>
<td>214</td>
<td>500</td>
<td>712</td>
<td>0.35</td>
</tr>
<tr>
<td>5000</td>
<td>14374</td>
<td>2187</td>
<td>5000</td>
<td>7187</td>
<td>0.34</td>
</tr>
<tr>
<td>50000</td>
<td>143998</td>
<td>21999</td>
<td>50000</td>
<td>71999</td>
<td>0.34</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th># of frame</th>
<th># of idle slots</th>
<th># of single slots</th>
<th># of collided slots</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>1376</td>
<td>19</td>
<td>50</td>
<td>68</td>
<td>0.20</td>
</tr>
<tr>
<td>500</td>
<td>1426</td>
<td>214</td>
<td>500</td>
<td>712</td>
<td>0.35</td>
</tr>
<tr>
<td>5000</td>
<td>14374</td>
<td>2187</td>
<td>5000</td>
<td>7187</td>
<td>0.34</td>
</tr>
<tr>
<td>50000</td>
<td>143998</td>
<td>21999</td>
<td>50000</td>
<td>71999</td>
<td>0.34</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Case</th>
<th>4-bit</th>
<th>8-bit</th>
<th>16-bit</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>66.78%</td>
<td>50.13%</td>
<td>33.44%</td>
</tr>
<tr>
<td>500</td>
<td>63.80%</td>
<td>46.84%</td>
<td>30.58%</td>
</tr>
<tr>
<td>5000</td>
<td>62.33%</td>
<td>45.27%</td>
<td>29.26%</td>
</tr>
<tr>
<td>50000</td>
<td>61.15%</td>
<td>44.03%</td>
<td>28.24%</td>
</tr>
</tbody>
</table>
enlarge the strength of QCD, for example from 4-bits to 16-bits, the value of $EI$ decreases due to the increased transmission overhead.

Figure 8(b) plots the $EI$ made by QCD against CRC-CD for BTs. In particular, the value of $EI$ tends to be stable around an average value 48%, 60.23%, and 78% under the three strength settings. This result shows that the BT algorithm is more stable, which is also demonstrated by [11].

VII. CONCLUSION

Collision detection is a crucial task in RFID systems. In this paper, we propose QCD, a fast and efficient collision detection scheme that does not require special hardware supports, e.g., the CRC design. Our theoretical analysis and comprehensive simulation results show that QCD can achieve accurate detection, and significantly reduce the transmission latency and communication overhead for existing anti-collision algorithms, compared with CRC-based approaches. In the future, we plan to explore more practical issues in the implementation of this scheme. Indeed, this design can be easily extended to other wireless fields, for example the neighbor discovery [26] and coverage [27] [29] [31] of sensor networks, and ad hoc network [30].

ACKNOWLEDGMENT

This work is supported in part by National Natural Science Foundation of China (NSFC) (No.60933003, No.60736016, No.60873262, and No.60903155), China Postdoctoral Science Foundation funded project (No.20090461298) Hong Kong Innovation and Technology Fund GHP/044/07LP and ITP/037/09LP, National High Technology Research and Development Program of China (863 Program)(2009AA01Z116), the Science and Technology Research and Development Program of Shaanxi Province under Grant No.2008KW-02, and IBM Joint Project.
REFERENCES


[26] Sudarshan Vasudevan, Don Towsley, Dennis Goeckel, Ramin Khalili, "Neighbor Discovery in Wireless Networks and the Coupon Collectors Problem", in Proceedings of ACM MobiCom, 2009

[27] Xiaole Bai, Santosh Kumar, Dong Xuan, Ziqiu Yun and Ten H. Lai, "Deploying Wireless Sensors to Achieve Both Coverage and Connectivity", in Proceedings of ACM MobiHoc, 2009


