IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON RADIO FREQUENCY IDENTIFICATION AND SECURITY TECHNIQUES

Received February 26, 2018, accepted April 12, 2018, date of publication May 1, 2018, date of current version June 5, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2832119

Towards Extended Bit Tracking for Scalable and Robust RFID Tag Identification Systems

ABDULRAHMAN FAHIM[1,2], TAMER ELBATT[3,4], (Senior Member, IEEE), AMR MOHAMED[2], AND ABDULLA AL-ALI[2]

[1]Wireless Intelligent Networks Center, Nile University, Giza 12585, Egypt
[2]Department of Computer science and Engineering, Qatar University, Doha 2713, Qatar
[3]Computer Science and Engineering Department, The American University in Cairo, New Cairo 11835, Egypt
[4]Electronics and Communications Engineering Department, Faculty of Engineering, Cairo University, Giza 12613, Egypt

Corresponding author: Abdulrahman Fahim (abdulrahman.fahim@gmail.com)

This work was supported by the Qatar National Research Fund (a member of Qatar Foundation) through NPRP under Grant 7-684-1-127. The work of A. Fahim and T. ElBatt was supported by the Vodafone Egypt Foundation.

ABSTRACT The surge in demand for Internet of Things (IoT) systems and applications has motivated a paradigm shift in the development of viable radio frequency identification technology (RFID)-based solutions for ubiquitous real-time monitoring and tracking. Bit tracking-based anti-collision algorithms have attracted considerable attention, recently, due to its positive impact on decreasing the identification time. We aim to extend bit tracking to work effectively over erroneous channels and scalable multi RFID readers systems. Towards this objective, we extend the bit tracking technique along two dimensions. First, we introduce and evaluate a type of bit errors that appears only in bit tracking-based anti-collision algorithms called *false collided bit error* in single reader RFID systems. A false collided bit error occurs when a reader perceives a bit sent by tag as an erroneous bit due to channel imperfection and not because of a physical collision. This phenomenon results in a significant increase in the identification delay. We introduce a novel, zero overhead algorithm called *false collided bit error selective recovery* tackling the error. There is a repetition gain in bit tracking-based anti-collision algorithms due to their nature, which can be utilized to detect and correct false collided bit errors without adding extra coding bits. Second, we extend bit tracking to ''error-free'' scalable mutli-reader systems, while leaving the study of multi-readers tag identification over imperfect channels for future work. We propose the multi-reader RFID tag identification using bit tracking (MRTI-BT) algorithm which allows concurrent tag identification, by neighboring RFID readers, as opposed to time-consuming scheduling. MRTI-BT identifies tags exclusive to different RFIDs, concurrently. The concept of bit tracking and the proposed parallel identification property are leveraged to reduce the identification time compared to the state-of-the-art.

INDEX TERMS Mutli-RFID reader systems, bit tracking, tag identification, reader-reader collision, tag collision, false collided bit errors.

I. INTRODUCTION

The radio frequency identification technology (RFID) is widely deployed in various applications, most prominently to automate scalable inventory and warehouse systems. The RFID technology is meant to extend the bar-code systems, due to its low cost, low identification time and scalability merits. However, RFID is not meant only for inventory management as it shows strong potential for a wide variety of applications ranging from asset tracking, healthcare to smart homes and cities [1]–[4]. RFID systems consist of RFID readers and tags. The tags carry out the same functionality as the bar-code sticker. Each tag has its unique pre-stored ID and a radio transceiver to transmit the ID to the reader, when interrogated. Generally, RFID tags operate in three different frequencies; low, high and ultra high frequencies and store different types of data, e.g., its stored ID or other data sensed or computed such as temperature, pressure, etc. There are two types of RFID tags. The first type is battery powered which is known as *active tags* and the second type is battery free which is known as *passive tags*. Active tags are able to sense the spectrum status and other tags transmitting to the reader to avoid collisions. On the contrary, passive tags are

powered up by the readers' power signal and they can scale to thousands of tags in one interrogation area, which makes them cost-efficient and effective solution. However, unlike active tags, they cannot detect concurrent transmitting tags, which increases the probability of collisions.

The main challenge in RFID systems with passive tags is signal collision. There are three types of collisions in multi RFID systems. First, if two or more tags transmit in the same slot, collision occurs at the reader, which is known as tag collision as shown in Figure 1(a). The second type occurs when a tag residing in a shared interrogation area between two or more readers receives signals from different readers simultaneously. These signals lead to a collision at the tag, which is known as reader-reader collision as shown in Figure 1(b). The third type is reader-tag collision and it occurs when a transmitting reader interferes with another receiving reader as shown in Figure 1(c). In this work, tags residing in a shared interrogation area between two or more readers are called *common tags*, whereas tags covered by one reader are called *exclusive tags*.

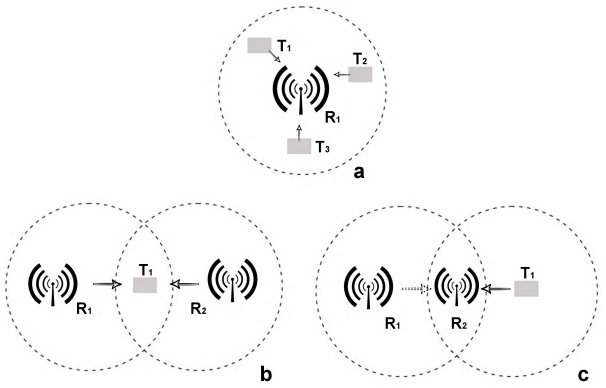

FIGURE 1. Collision events in RFID systems a) Tag collision at the reader. b) Reader-reader collision at the tag. c) Reader-tag collision at the reader.

Many algorithms have been proposed to tackle the problem of tag collision. They can be classified into two main categories; sequential and non-sequential algorithms [5], [6]. Examples on the latter are PIP [7], [8], FDMA [9], Buzz code [10] and CDMA [6], [11]–[15]. Sequential algorithms like ALOHA [16], [17], binary tree (BT) [18], and query tree (QT) [19]–[21] are usually easier to implement and can be applied to off-the-shelf RFID products.

In this work, we mainly focus on sequential algorithms especially binary tree algorithms. The algorithm splits the collided tags into smaller subsets until all tags are identified. The idea behind binary tree algorithms is that each tag has a counter in its storage memory. In the beginning of the identification process, each tag picks a random number within a range of numbers specified by the reader. During the identification, a reader keeps transmitting feedback signals based on the slot type. The feedback has three types; idle, collision and readable (success). Each tag updates its own counter based on the type of the feedback. By using this technique, each tag is assigned a unique slot which prevents collision from occurring.

Bit tracking has been adopted by many researchers due to its promising role in decreasing the identification time. Tags IDs' bits are perfectly synchronized, so when a group of tags send in the same slot, a reader receives a sequence of bits called *pattern*. A reader has the ability to differentiate between the state of each bit in the pattern received. Thus, a reader can locate the position of colliding bits in a collision slot. Consequently, it receives some readable bits (0s and 1s) in a collision slot. In other words, a collision slot is not a complete waste as a reader can receive some bits which can be used to reduce the identification delay.

Previous work tackled reader-reader collision by optimizing the reading schedule, yet with two major shortcomings. First, neighbor readers that are sharing a coverage area cannot transmit signals simultaneously because of collisions. This constraint causes a significant increase in the identification delay [22]. Second, in some proposed schemes, common tags are triggered multiple times by distinct readers unnecessarily.

To overcome reader-tag collision, Electronic Product Code (EPC) Class 1 Generation 2 [23] has specified two different frequencies for the uplink and downlink. In other words, readers commands and tags responses have different frequencies. As a result, our focus in this work is on tag collision and reader-reader collision.

This paper builds upon our earlier work [24]. Our main contribution is multi-fold. First, we introduce and evaluate false collided bit errors. Second, we propose a novel zero-overhead algorithm called *false collided bit error selective recovery (FSR)* that tackles false collided bit errors. Third, to the best of our knowledge, we are the first to utilize bit tracking [20] in multi-reader RFID systems, which reduces the identification delay compared to the Aloha-based scheme employed in [25]. Fourth, we propose a novel technique coined *parallel identification property*. In essence, the incorporated method enables readers that have already identified their exclusive tags to identify common tags (i.e., covered by more than one reader), while their neighbouring readers are identifying their exclusive tags.

This work is organized as follows. The related work is surveyed in Section II. A background on the bit-tracking concept is given in Section III. The system model, underlying assumptions and necessary notation are introduced in Section IV. The study and analysis of communication bit errors in single reader RFID systems, along with an algorithm to tackle them, are presented in Section V. Our novel algorithm for multi-reader RFID systems is presented in section VI. The performance evaluation of the proposed algorithms and simulation results are presented in Section VII. Finally, our work is concluded in section VIII.

II. RELATED WORK

In this section, we briefly review some of the well-known algorithms proposed to handle the two types of collisions; tag and reader-reader collisions. Tag collision occurs when

multiple tags reply in the same slot, causing collision at the reader. Reader-reader collision occurs at common tags residing in the interrogation area between multiple readers. When two or more readers transmit in the same time, common tags in the shared interrogation area cannot resolve commands from any of the readers.

A. FALSE COLLIDED BIT ERROR

Many approaches are proposed to tackle the problem of detecting and recovering flipped bit errors. For example, Cyclic Redundancy Check (CRC-16), which is utilized in Gen2 protocol [23], is able to detect flipped bit errors and request tag ID re-transmission. Moreover, in [26], the authors utilized Reed-Solomon and Hamming codes to detect and correct flipped bit errors. Any of the above mentioned codes are applied by adding extra bits to the original tag ID. In bit tracking based protocols, a new type of bit errors, which is called false collided bit error, occurs when a reader mistakenly declares collided bit while it is originally received successfully. It makes the reader send wrong feedback signal which affects tags' counter updating. This results in more wasted slots; idle and collision slots. False collided bit error is different from bit flip error, and the above mentioned codes cannot correct or detect this type of bit errors.

B. TAG COLLISION

Sequential algorithms family has three main categories; binary tree (BT), query tree (QT) and Aloha [27]. Aloha based protocols are probabilistic, while on the other hand tree protocols are deterministic.

In BT algorithms [28], in the beginning of the identification process, a reader sends a trigger signal to tags. Each tag picks a random number within a given range sent in the triggering command. The chosen random number is saved in a counter, and each tag keeps changing it during the identification process. The updating procedure is based on feedback commands that are sent by a reader and it differs from one algorithm to another. However, all of them have the same theory of operation. We introduce here arbitrary binary splitting (ABS) [18], but we note that other algorithms may differ. In the beginning of each slot, tags with counters equal to zero send their IDs, and other tags stay idle. If only one tag sends its ID, the reader receives it successfully and sends readable feedback. The identified tag does not participate in the remaining slots, and other tags update their counters and decrease them by one. If no tag sends its ID, the reader sends an idle feedback, and tags decrease their counters by one. If two or more tags send in the same slot, collision occurs and the reader sends a collision feedback. Tags participating in the collision slot pick zero or one randomly and the remaining tags increase their counters by one. The identification process is terminated when all tags are identified.

It is hard to summarize ALOHA based algorithms because they have been continuously modified along the previous decade and thus, underwent multiple enhancement iterations [29], [30]. We mention here the most related versions to our work that are called framed ALOHA systems [16], [17]. The identification process consists of multiple frames. In each frame, the reader sends a broadcast message to all tags asking them to choose a random number between 0 and an upper limit, L. In each slot, the reader interrogates the tags that pick number 0. If only one tag selected this number, the reader successfully reads the tag's ID and sends readable feedback signal. If no tag answers, the reader sends idle slot feedback. The tags decrease their random numbers by 1 if the feedback signal is readable or idle. If two or more tags pick the same number, the reader can't receive the data successfully and sends a collision feedback. The tags that participated in this slot will participate in the next frame. The readers continue in the same manner until it identifies all tags. In summary, the basic difference between Aloha and BT algorithms is the ways of handling tags participating in a collision slot. In Aloha, tags participating in a collision slot ignore the currently used frame and take part in the next frame. On the other hand, tags participating in a collision are resolved in the consequent slots in binary tree based algorithms.

In QT algorithms [31]–[33], tags do not have physical storage, hence they do not save a random number during the identification as opposed to the aforementioned techniques. Therefore, tags supporting QT based algorithms are the cheapest. The theory of operation behind QT is that a reader transmits a query to all tags and when the query matches a tag's ID, the tag responds. A reader transmits a prefix and the tags check if their IDs contain this prefix or not. If the prefix matches a tag's ID, the tag responds, on the other hand, it ignores the command if the ID does not match. If collision occurs, a reader appends to the original prefix zero or one or even a pattern of zeros and ones. Thus, the prefix becomes longer and the number of IDs that match the new prefix decreases. A reader keeps updating the prefix until all the tags are uniquely matched with one of the prefixes and transmit its ID with no collision. The updating scheme of the prefix differs from one algorithm to the other. In addition, other schemes control the number of bits that are sent by the tag [33], [34].

C. READER-READER COLLISION

The fundamental constraint in multi-reader systems is reader-reader collision. Most of previous work avoid collision by ensuring that interfered readers do not operate simultaneously, or use different frequencies [22], [35]–[37]. Researchers proposed many algorithms tackling reader-reader collision through optimizing the reading schedule. In other words, the algorithms aim to decrease the number of rounds needed to identify all the tags in the system. The most commonly used algorithm that used the pre-mentioned approach is Colorwave [22]. It prevents any neighbor readers from sending in the same time. The algorithm allows a reader to transmit and prevent its neighbor readers from transmitting. Thus, the reader-reader collision is alleviated, however, this approach induces delay.

In [25], the proposed algorithm, Season, allows neighbor readers to work cooperatively over two phases, which reduces the identification time significantly. In the first phase, readers identify their exclusive tags concurrently using ALOHA algorithm. Common tags cannot resolve concurrent commands, so they ignore them. In the second phase, the readers cooperate to identify common tags in identification rounds. In each round, the system selects a different group of readers to identify common tags. However, this approach has drawbacks that cause a significant delay. That is when a reader identifies its own exclusive tags, it has to wait for the reader with the maximum number of exclusive tags to finish identifying its exclusive tags, afterwards, the system cooperate to identify common tags. Furthermore, common tags are identified sequentially in multiple identification rounds. In unbalanced tags distribution, this algorithm induces significant delay, however, our approach makes use of the non-uniform distribution of tags residing in an interrogation area to identify them faster using parallel identification property.

III. BACKGROUND

In this section, we review *bit tracking* and *Optimal Binary Tracking Tree (OBTT)*, which is a sequential algorithm tackling tag collision problem using bit tracking. We also analyze the algorithm and emphasize the implied features that are utilized in our algorithm. In other words, we look into OBTT from a different angle that helps in developing our algorithms.

A. BIT TRACKING

Bit tracking is a technique based on Manchester code. A bit "1" is expressed by a high to low transition and a bit "0" is expressed by low to high transition. RFID tags send their IDs (0's and 1's) encoded as sequential transition using Manchester code. Manchester code exhibit a unique property that facilitates tag collision detection. If the reader fails to detect a transition in the received sequence, it means that a collision occurred in a certain bit. The reader leverages the aforementioned property to detect tag collisions (two or more tags send in the same slot). Figure 2 shows an example of two synchronized tags A and B with ID's: 01100001 and 01110100, respectively. As shown, the reader receives $011x0x0x$, where x represents bits that the reader could not resolve due to having different values.

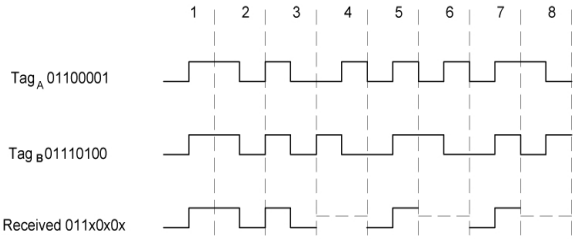

FIGURE 2. Bit tracking.

B. OPTIMAL BINARY TRACKING TREE PROTOCOL (OBTT) [20]

In OBTT [20], the authors target the tag collision problem in single reader systems. The algorithm works through two main phases; random number assignment followed by identification process. In the first phase, a reader interrogates the tags to pick a random number in a given range. Each tag has a counter called TC, which carries the random number it chooses. In the second phase (identification), the reader interrogates tags with TC equal 0 to send their IDs. If one tag sends its ID, the reader successfully decodes it and sends readable feedback (positive ACK). If no tag answers, the reader sends idle feedback. In the case of idle and readable feedbacks, the tags decrease their TCs by 1. However, if a reader does not decode an ID successfully, it means that two or more tags have picked the same number, and a collision occurs. Consequently, the reader sends collision feedback. Tags with $TC > 0$ increase their TCs by 1 and the colliding tags update their TCs according to the following equation:

$$TC = TC + ID(j) \tag{1}$$

where j is the position of the first collided bit. Following with the same example, the two colliding tags receive a feedback from the reader informing them with the position of the first collided bit, ($j = 3$) Consequently, tag_A picks number zero, $0 + 0 = 0$, and tag_B updates its counter to be one, $0 + 1 = 1$. Thus, the two tags will be identified sequentially in the next two slots.

IV. SYSTEM MODEL

In our algorithm, passive tags send their IDs perfectly synchronized on the bit level to one or more readers by Manchester code as we can use *bit tracking* features. In this paper, we consider two scenarios. First, we consider a single reader system where tags experience MAC collision (tag collision only) and false collided bit errors. Second, we extend the system to multiple readers in error free environment without communication errors (other than MAC layer collisions). We only consider false collided bit error while leaving handling mixed types of bit errors (coexistence of bit flip and false collided bit errors together) for future work. We define p as the probability of occurrence of a false collided bit error. Collision might occur due to physical collision or errors, so we identify each one separately to alleviate confusion. Collided bit occurs when two or more tags send different bits, and it is denoted as *corrupted bit*. On the other hand, we denote the bit that is altered from success to collision (0 or 1 to x) due to false collided bit error as *infected bit*. The position of an infected bit has a significant effect on the identification time. In some cases, as it will be shown later, false collided bit error might occur without affecting the identification time. We denote false collided bit errors that affect the identification time as *sensitive bit errors* as opposed to *tolerable bit errors* that don't have an effect on the identification time. When a sensitive false collided

bit error takes place, *false collision* occurs which consumes more time. However, for *real collision*, it means that there are two or more physical tags sending in the same slot. We denote j as the position of a particular received bit.

When we extend the system to two or more readers, we consider N RFID readers form a ring as shown in Figure 3. Readers are connected via a separate network, and they have sufficient storage and computation resources. Each reader is sharing two interrogation areas with two surrounding readers such that the system has N shared interrogation areas. Let R_i denote the i^{th} reader index in the ring, where $i \in 1, 2, \ldots, N$, and r_i represents the number of the exclusive tags of the reader R_i. Let r_{ij}, which is equal to r_{ji}, represent the number of common tags between R_i and R_j. We also denote n as the total number of tags. Key notations are summarized in Table 1.

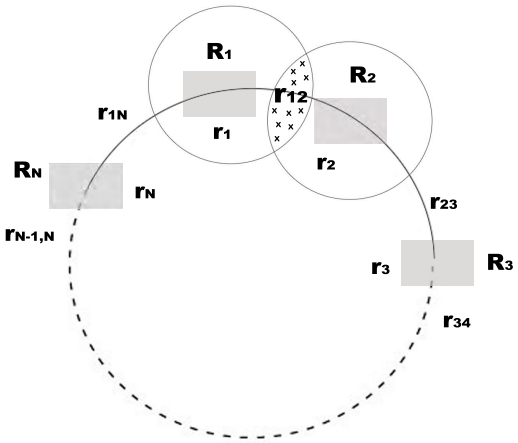

FIGURE 3. System model.

A *slot* is the time taken by a tag to reply back either with its full ID, b, or a fragment of it to a reader, which replies back with an ACK. The slot has three states; idle, collision or readable. An idle slot occurs when a reader receives no response from tags. A collision slot occurs when a reader receives noisy signal which implies multiple replies. A readable slot occurs when a reader decodes one or more tags within the same slot. We note that the definition of readable slot is different from the literature. This is because readers can successfully decode two tags in the same slot by *parallel identification property* proposed in this paper, however, other literature algorithms identify one tag in each readable slot.

V. SINGLE-READER TAG IDENTIFICATION IN THE PRESENCE OF ERRORS

In this section, we analyze the performance of bit-tracking based algorithms under false collided bit errors. It occurs when a successful (readable) bit is perceived as collision due to channel imperfection. Towards this objective, we need first to emphasize the implied features of OBTT and mapping collision resolving procedure to a *splitting tree*. Moreover, we introduce a zero overhead novel algorithm called *false collided bit error selective recovery* (FSR) tackling this type

TABLE 1. Key notation.

Symbol	Description
N	Total number of readers
n	Total number of tags
R_i	i^{th} reader in the ring
r_i	Number of exclusive tags of reader R_i
r_{ij}	Number of common tags between readers R_i and R_j
b	Length of tag's ID
j	The position of a particular bit in a received pattern
p	Probability of error

TABLE 2. Tags ID.

Tag Name	Tag IDs
W	00010110
X	00101111
Y	00111010
Z	01110110

TABLE 3. Example pattern received in each slot.

Slot Number	Slot Name	Participated Tags	Pattern Received
1	S	W, X, Y, Z	$0xxxxx1x$
2	L	W, X, Y	$00xxxx1x$
3	LL	W	00010110
4	LR	X, Y	$001x1x1x$
5	LRL	X	00101111
6	LRR	Y	00111010
7	R	Z	01110110

of error. We can garner the repetition fact in anti-collision algorithms' nature to detect and correct false collided bit errors without adding extra coding bits. We note that FSR can recover some bit errors, but not all of them.

1) ANALYZING OBTT OVER AN ERROR-FREE CHANNEL

The algorithm splits collided tags into smaller sub groups until each tag sends alone with no collision. This scheme can be represented by a splitting tree. Due to space limitations, we refer the reader to [38] for more information about tree algorithms and splitting trees. To illustrate the formation of the splitting tree, we consider a simple example shown in Table 2, where a set of four tags send their IDs in the same slot. The pattern received is $0xxxxx1x$, and they follow (1) to be split into different slots. Consequently, tag_W, tag_X and tag_Y pick number zero, $0 + 0 = 0$, and tag_Z updates its counter to be one, $0 + 1 = 1$. Following the same procedure, the algorithm keeps splitting the tags involved in the collision into smaller subsets until tags are identified. The pattern received in each slot is shown in Table 3, and its corresponding position in in the splitting tree is shown in Figure 4. We highlight two important characteristics of OBTT. First, tags involved in a collision are divided into **two** slots. Second, OBTT algorithm identifies tags involved in a collision without idle slots, so the slots in splitting tree are always readable or collision slots. This is because OBTT

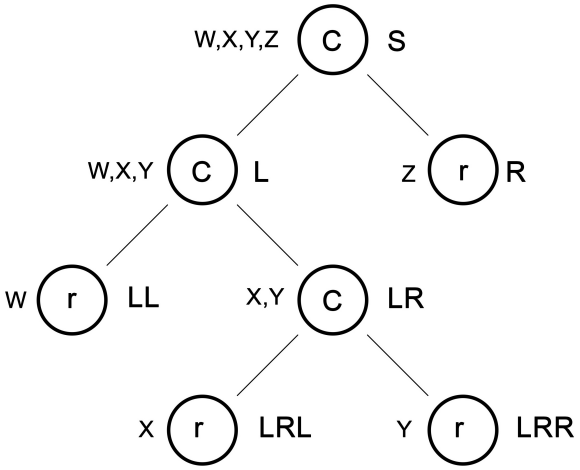

FIGURE 4. An example shows how OBTT algorithm splits tags involved in a collision into smaller subsets until all tags are identified. Each circle represents a slot, and its type, readable (r) or collision (c), is written inside. The name of each slot and its participating tags are shown next to it.

assigns slots to collided tags on deterministic base in contrast to ALOHA that follows random slot assignment procedure.

2) THE EFFECT OF FALSE COLLIDED BIT ERROR

As explained in Section III, when collision occurs at a reader, it sends a collision feedback signal informing the tags with the position of the first collided bit. Thus, the tags follow the algorithm and split the collided tags into two subsets using (1). Since the false collided bit is originally 0 or 1 and it is changed due to channel error, the collided tags will re-send their IDs in one of the sub slots and the other slot is idle. To make the problem clear, we use the previous example shown in Figure 4, and expose it to false collided bit errors. Assume a false collided bit error occurs in the third bit in *LRL* slot, where tag_X is sent. The received pattern in this case is $00\tilde{x}01111$, where $\tilde{x}$ is the bit in error. The reader perceives it as a collision slot while it is originally readable slot, and consequently informs the tags with the position of the first collided bit ($j = 3$). Tag_X updates its counter according to (1) to send its ID in *LRLR* slot as shown in Figure 5. Consequently, no tag sends in *LRLL*, which makes it an idle slot. In summary, *LRL* and *LRLL* are false collision and idle slots, respectively. These two slots are considered wasted, which can be prevented using smart algorithms to achieve the performance of OBTT in error free environment. We define *wasted slots* as the unnecessary slots that are consumed due to false collided bit error. The aforementioned example can be generalized to any number of colliding tags if the reader informs tags with the position of a false collided bit error. Thus, there are always two wasted slots if a *sensitive false collided bit error* takes place.

3) SENSITIVE AND TOLERABLE FALSE COLLIDED BITS

The false collided bit error's effect depends on its position in the pattern received at the reader. If the false collided bit

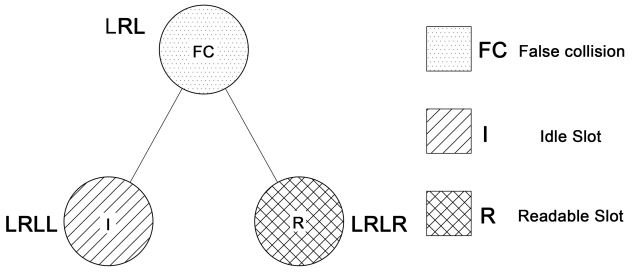

FIGURE 5. An example shows the effect of false collided bit error on OBTT algorithm.

occurs before the real collided bit, $j_{false} < j_{real}$, it becomes dominant and the reader informs the tags with the position of the false collided bit. In this case, sensitive false collided bit error occurs and affects the identification time. On the other hand, if $j_{false} > j_{real}$, the reader informs tags with the position of the real collided bit, consequently, the bit error becomes tolerable. These two cases can be verified from the given examples. In case of one transmitting tag, readable slot occurs, however, the reader perceives any false collided bit error wherever its position as a collided bit and consequently sensitive bit error always occurs.

The key observations in false collided bit error and OBTT that motivate our algorithm can be summarized in the following:

- Sensitive and tolerable collision errors depend on the position of the reported bit error. The latter does not affect the identification time.
- As part of the nature of the OBTT, if one slot is a false collision, it results in an idle and a collision slot. As a result, the numbers of wasted collision and idle slots due to false collision are equal.
- For slots belonging to the same root (same branch) such as S, L, LR, LRL or S, L, LR, LRR, the number of collided bits in a pattern is less than or equal the number of collided bits in a previous pattern in the branch. This is because there are zero or more bits decoded in each new slot. This observation is intuitive and can also be verified from table 3.

A. FALSE COLLIDED BIT ERROR SELECTIVE RECOVERY (FSR)

FSR does not only detect false collided bit errors but it recovers some of them. The unique feature in FSR is zero-overhead i.e, it does not add error detecting/correcting coding bits (overhead) as CRC or Reed Solomon codes. FSR depends only on the repetition gain of the tags that sent their IDs many times to recover collision. The system keeps tracking tag responses to build the splitting tree, which is used to determine the roots of a transmitting tag (tag's branch). Tag's roots (tag's branch) are the slots that a tag has participated in. In the example shown in Figure 4, tag_X sends its ID four times in the following slots; S, L, LR and LRL (tag's branch slots). In each slot, some new bits from the tag's ID are decoded and become available to the system. In the example

shown in Figure 5, a false collided bit error occurs in the third bit ($j = 3$) which results in two wasted slots. In the pattern received in the previous node, *LR*, bit number three ($j = 3$) was received successfully, which means that the false collided bit error is originally bit 1. This example shows that the bits received in the previous slots can help in correcting false collided bit error. Thus, by utilizing this information, the system can differentiate between false and real collisions.

Our algorithm modifies OBTT to minimize wasted time slots due to channel error. When a collision occurs, the reader builds a splitting tree with S equal to the currently received pattern and creates two leafs (L and R). It keeps tracking tags responses and fulfill the tree's leafs. For any sub collision, the reader extract the leaf's roots (tags roots) and correct the bit errors if any and sends the appropriate feedback. If it is still collision slot, the reader records the pattern and creates two blank sub leafs that will be fulfilled with future replies. The reader terminates the splitting tree when all leafs are fulfilled because it means that all tags involved in this splitting tree are received successfully. Algorithm 1 summarizes our scheme.

Algorithm 1 FSR Algorithm

```
Receive a pattern
if Splitting Tree exists then
  if collision then
    Recover bit errors if any
    Correct the pattern
    if the corrected pattern has no collided bits then
      Transmit readable feedback
    else
      Transmit collision feedback
    end if
  else
    Record the pattern
    Proceed in normal OBTT operations
  end if
  if All tree leafs are fulfilled then
    Terminate the current splitting tree
  end if
else
  Build a new splitting tree with S = the received pattern
end if
```

B. FSR LIMITATIONS

In this part, we discuss the limitations of our algorithm. In addition, we briefly summarize the cases where the aforementioned features can be utilized. Our algorithm does not recover all bit errors, but it decreases the number of sensitive false collided bit errors with zero cost coding. In Figure 6, the number of sensitive bit errors over the total number of bit errors is shown. FSR limits the taken place of sensitive bit errors from 60% in OBTT to only 20%. The parameters used this experiment are the same used in Section VII.

The main idea is to record a history of a branch's received patterns from all nodes and utilize this information to fill gaps

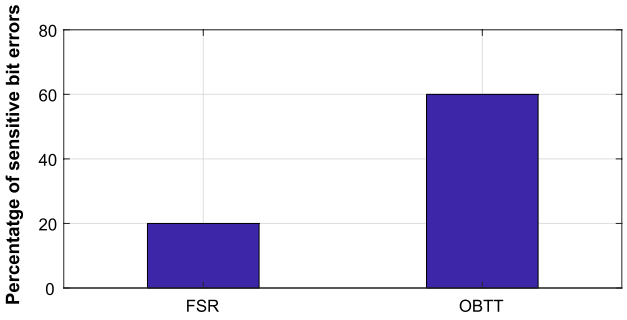

FIGURE 6. An example shows the effect of false collided bit error on OBTT algorithm.

further down the branch and correct future false collided bit errors if possible. A tag must participate in a previous slot, thus the system saves a fragment or some bits of its ID. We can conclude that the features cannot be utilized unless a collision occurs. However, this does not guarantee bit error recovery, because the retrieval process depends also on the position of the bit error; lacking bit information in the history of the branch leads to error occurrence. To sum up, the cases that our algorithm cannot retrieve false collided bit errors are:

- If a false collided bit error occurs in a pattern when it was first received (i.e the tags do not participate in a collision slot before).
- If the location of a false collided bit error in the received pattern is corresponding to a real collided bit in the previous slots.

VI. MULTI-READER RFID TAG IDENTIFICATION USING BIT TRACKING (MRTI-BT)

In this section, we introduce our novel MRTI-BT algorithm. We have two main contributions in this part. First, we introduce MRTI-BT for single reader, which is a modified OBTT, that addresses tag collision problem without scarifying the identification time. Each tag sends a short length ID, so that the reader can resolve the random number contention. Then, the reader identifies the tags in a collision free manner. Second, the incorporated *parallel identification property* tackle the *reader-reader collision* problem. It enables readers that have already identified their exclusive tags to identify common tags while their surrounding readers are still identifying exclusive tags. The proposed scheme (MRTI-BT) accelerates the identification time significantly compared to the state-of-the-art. In the beginning, each reader identifies its exclusive tags. As soon as a reader finishes, it cooperates with a neighbouring reader to identify common tags by *parallel identification property*. It follows the same procedure until all tags are identified. We note that our algorithm identifies both common tags and exclusive tags concurrently as opposed to Season that identifies them sequentially. We notice two main deficiencies in previous work that motivate ours, and they are summarized in the following points.

- The number of tags residing within the exclusive interrogation area of each reader varies. In Season [25], a reader

that identifies its own exclusive tags waits for the reader with the maximum number of exclusive tags before it starts identifying common tags. This induces delay and inefficient use of the system resources.
- In OBTT, tags send their complete ID in each identification slot, although the first collided bit most of time occurs at high significance bits. As illustrated earlier, the reader needs only to know the first collided bit to resolve collision, so waiting for complete ID is an unnecessary waste of time.

As shown in Figure 3, readers are distributed such that the number of common tags groups is equal to the number of exclusive tags groups. The numbers of tags residing in each group are assumed to be known before the identification process. There are variety of counting protocols proposed in literature [39]–[41] that can be utilized for that purpose. The MRTI-BT identification algorithm passes through three main phases; 1) Random number generation, followed by 2) Collision handling and 3) Parallel identification.

We begin by explaining *parallel identification property*, which is a key feature in this paper. By utilizing this property, the system can identify two tags in one slot. Figure 7 shows an example, with two perfectly synchronized tags; t_i and t_j with IDs 10011000 and 11001110, respectively. t_i is an exclusive tag and t_j is a common tag such that reader R_2 hears only t_j whereas reader R_1 hears both tags. The received ID at $R_1 = 1x0x1xx0$ and the received ID at $R_2 = 11001110$. Note that R_2 received the correct ID identifying t_j. Since direct communication is allowed between readers, then the ID of t_j would be known to R_1, which can leverage it to decode the corrupted bits in the ID of t_i. Simply, a corrupted bit in t_i would be the complement of the corresponding bit in the colliding t_j ID. This can be easily verified from the example given in Figure 7. *Parallel identification property* is feasible only in case of two tags in a two-reader system, where one of the tags is exclusive to one of the readers whereas the other is common to both.

The basic idea behind our algorithm is that we use short length identification to resolve random number contention and assign unique random number to each tag. Second, we utilize the property to identify a common tag and an exclusive tag in each slot (if one of the neighbour readers is free) or each reader identifies its own exclusive tags in contention free manner. This procedure has two main advantages. First, it allows neighbour readers to co-operate and identify their tags simultaneously. Second, even in single reader identification, our algorithm outperforms OBTT because it identifies the tags over two phase; resolve collision and identifying.

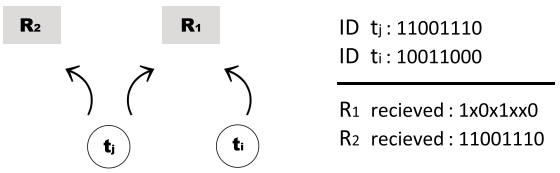

FIGURE 7. Parallel identification property.

In order to achieve that, our algorithm work on three main phases. First, it assigns a unique range of random numbers to each group of tags. Second, the system resolves collision and assigns a unique number to each tag. Finally, the system identifies tags sequentially with no collision and applies parallel identification property to identify common tags and exclusive tags simultaneously.

A. RANDOM NUMBER GENERATION

In this stage, readers assign unique range of random numbers to each group of tags. We explain how to extent the assignment procedure to multi-reader RFID systems. Tags are interrogated by readers in order to be assigned a random number within a predefined range. Consequently, tags are able to transmit their stored IDs according to the assigned slots. In [20], it is proven that the range of numbers assigned to a group of tags strongly impact the identification time. It is also proven that the optimal range is $[0, \lceil 0.6\hat{d} \rceil]$, where $\hat{d}$ is the estimated number of tags belonging to a group. In our model, the coverage area of a reader includes three different groups of tags; two sets of common tags and a set of exclusive tags. Since the estimated number of tags belonging to each set, $\hat{d}$, is different, readers assign different ranges of random numbers to each group. The three sets in an interrogation area hear any signal from the reader, and hence unique ranges cannot be assigned. To get over this challenge, unique name (ID) is assigned to each group. Thus, a reader can communicate with each group with the assigned ID individually while the other two groups in the interrogation area ignore its command unless they are triggered with their IDs. The assigned ID is called *group ID* (GID). We note that GID is different from the prestored IDs in the tags. It is just intitiated during the identification process so that the reader can trigger a specific group of tags while the others keep silent.

In this phase, each group of tags is assigned a group ID (GID) and a unique range of random numbers. We first define *assignment command* and then proceed in our illustration.

Assignment Command: It is a command sent by a reader such that tags perform differently based on their conditions. Upon hearing the command, if a tag has not been assigned a random number, it chooses a number within the given range. On the other hand, if a tag has already chosen a number in a previous assignment, it ignores the command.

Our algorithm performs this phase over two stages. In the first stage, readers assign GID and unique range of random numbers to their exclusive tags. They transmit triggering commands concurrently and hence readers' signals collide at common tags, so the tags keep silent. Only exclusive tags are able to hear the triggering commands without collision. In the second stage, common tags are assigned GIDs and random numbers over multiple iterations. The main idea is to allow a reader to transmit an *assignment command* while a neighbour reader transmits a dummy command. As a result, one group of common tags hears the command while the other

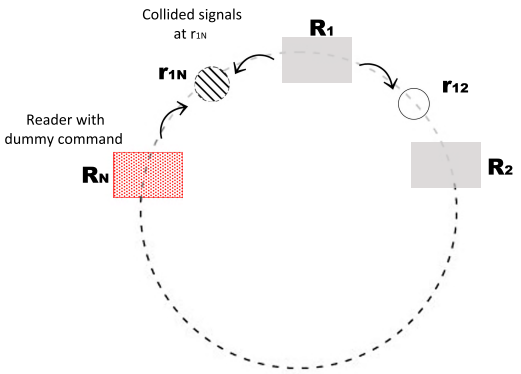

FIGURE 8. Assigning random numbers to common tags.

group hears colliding signals that cannot be resolved. We note that exclusive tags keep silent because they have already been assigned random numbers and GIDs in the the first stage. To demonstrate the idea, we consider an example in Figure 8. R_1 transmits an *assignment command* while R_N transmits a dummy command. As shown, r_{12} tags hear the command with no collision while r_{1N} cannot resolve the command. Inspired by the shown example, the system assigns GID and random numbers in N iterations. In each iteration, R_i is chosen to transmit an *assignment command* while its neighbor, R_{i-1} transmits a dummy command and the remaining readers in the system keep silent. Thus, after N iterations, all the groups in the system are assigned distinct GIDs, and each tag has picked a slot number.

B. COLLISION HANDLING

So far, all tags in the system have already chosen a random number. However, within the same group, some tags may have picked the same random number. The purpose of this phase, *collision handling*, is to resolve this "random number contention" and assign a unique random number to each tag to avoid collisions. Therefore, each individual tag can transmit its stored ID in a unique slot in a contention-free manner in the next phase.

We utilize the idea of short length identification, and we emphasize it by the following example Let t_i and t_j are two colliding tags, and each send only x bits (not the complete ID, $x < b$). Since each ID consists of 0s and 1s, the probability of detecting a collided bit in the received sequence is as follows:

$$p(\textit{detecting collision}) = 1 - (\frac{1}{2})^x = \epsilon, \qquad (2)$$

where ϵ becomes larger as the number of sent bits in the first phase increases, which agrees with the intuition. Thus, motivated by this notation, we propose MTRI-BT algorithm that identifies tags in two phases; *collision handling* and *parallel identification* as opposed to single phase identification in OBTT. Only x bits are transmitted for collision resolution in the first phase, while the remaining bits, $b - x$, are sent in the next phase. *Collision handling* phase is conducted through

Algorithm 2 Tag Operation in Intra-Group Slot Assignment

```
TC2 = TC1
while TC1 ≥ 0 do
  if TC1 = 0 then
    send first x bits
    receive feedback from the reader
    if feedback = readable then
      TC1 = TC1 − 1
      TC2 remains as is
    else {feedback = collision}
      TC1 = TC1 + ID(i)
      TC2 = TC2 + ID(i)
    end if
  else
    if feedback = readable then
      TC1 = TC1 − 1
      TC2 remains as is
    else if feedback = collision then
      TC1 = TC1 + 1
      TC2 = TC2 + 1
    else
      TC1 = TC1 − 1
      TC2 = TC2 − 1
    end if
  end if
end while
```

two main stages; *intra-group slot assignment* and *inter-group slot assignment*.

Intra-group Slot Assignment

Each group of tags has a unique identifier, however within the same group, two or more tags may pick the same slot number. In this stage, each tag is assigned a unique slot. Our proposed scheme, as shown in Algorithm 2, operates in a similar manner to OBTT with two main differences.

- We use two phase identification technique. Thus, only x bits are sent in this phase for collision resolution, and the remaining bits are sent in *parallel identification* phase.
- In OBTT, each tag has only one internal counter that is updated based on reader's feedback as illustrated in Section III. In our procedure, tags have two internal counters; TC_1 and TC_2. TC_1 is updated in a similar manner to OBTT's counter, TC, (collision resolution). However, TC_2 is used to save the tag's unique slot, so each tag sends the remaining bits of each ID in a unique slot in the next phase, *parallel identification*.

Each tag updates its two counters based on the reader's feedback signal. We show in this paragraph, how the tag updates its counters to resolve collision and reserve a unique slot for the next phase. Initially, TC_2 and TC_1 have the same value, that is picked by the tag in the previous phase. TC_1 is updated in a similar manner to OBTT's counter. TC_2 updating procedure is as follows. In case of idle feedback, tags decrease the counter's value by 1. In case of collision feedback, tags resolve collision using (1). Both updates are similar to TC_1

TABLE 4. Intra-group slot assignment.

TC_1	TC_2	Slot State	Slot Number
0 , 2 , 2 , 3	0 , 2 , 2 , 3	Initial State	0
(-1) , 1 , 1 , 2	0 , 2 , 2 , 3	Readable Slot	1
(-1) , 0 , 0 , 1	0 , 1 , 1 , 2	Idle Slot	2
(-1) , 0 , 1 , 2	0 , 1 , 2 , 3	Collision Slot	3

and TC in OBTT. As opposed to TC_1, in case of readable feedback, a tag does not change the value of TC_2. An example of our updating counters scheme is shown in Table 4. There are four tags with initial counter values, and they are updated based on the reader's feedback signal. A readable feedback is received in slot 1. Thus, tags keep their TC_2s as they are and decrease TC_1 by one. In slot 2, the values of both counters in all tags are decreased by 1. Finally, in slot 3, tags update their counter according to (1). As shown, all tags have different TC_2s and, hence, they can send in different slots in the next phase. In the next three slots (not shown in the table), the reader identifies them sequentially with no collision because they have different TC_1s. We also note that readers save the first x bits of each ID, and hence, tags send $b - x$ only in the next phase.

Inter-group Slot Assignment

We illustrated how the collision is resolved within the same group. In this stage, the algorithm manages the reader cooperation during *collision handling* phase such that each reader can resolve the collision of the tags that are residing in its interrogation area without interfering the neighbour reader. A reader interrogates a group of tags by broadcasting a command with the assigned GID. The group's tags reply and other tags even if they hear the command, they ignore it. Readers leverage the aforementioned property to assign a unique slot to each tag in the system.

First, readers resolve collision of their exclusive tags simultaneously. Second, the system arranges readers coordination for common tags collision resolution. This can be usually done over two iterations. In the first iteration, even readers, $R_2, R_4 \ldots .., R_N$, identifies odd groups of common tags, $r_{1,2}, r_{3,4}, \ldots .., r_{N-1,N}$. In the second iteration, even readers identifies even groups of common tags.

C. PARALLEL IDENTIFICATION

So far, phases 1 and 2 of the proposed MRTI-BT algorithm have resolved random number contention. In this phase, the system identifies the remaining bits of each tag, $b - x$, because the first x bits are already sent in the previous phase. The system also applies *parallel identification property* whenever it is possible.

First, each reader identifies its exclusive tags in collision free manner. As mentioned, the distribution of tags is not usually uniform. The number of tags is different from one interrogation area to another. Thus, it is expected that some readers finish identifying their own exclusive tags before others. Those readers co-operate with their surrounding readers to identify common tags by applying *parallel identification property*. We design a heuristic procedure that manages the readers cooperation. When a reader completes identifying its exclusive tags, it co-operates with the neighboring reader that has the most number of exclusive tags. Since the property cannot be applied on more than two readers, a busy flag (bf) is set by the two readers during applying the property to indicate that they are not ready to cooperate with any other readers. Algorithm 3 shows our heuristic approach. In short, the property is utilized to make use of the system resources while the reader with the maximum number of tags is still identifying its exclusive tags. When the reader completes identifying its exclusive tags, the system proceed with a behaviour similar to the state-of-the-art.

Algorithm 3 Reader R_i Performs *Parallel Identification Property* With Surrounding Readers

```
TC_2 = TC_1
while TC_1 ≥ 0 do
  if R_{i-1}(bf) = 0 & R_{i+1}(bf) = 0 then
    if r_{i-1} ≥ r_{i+1} then
      R_i(bf) = 1
      R_{i-1}(bf) = 1
      cooperate with R_{i-1} to identify r_{i,i-1}
    else {r_{i-1} < r_{i+1}}
      R_i(bf) = 1
      R_{i+1}(bf) = 1
      cooperate with R_{i+1} to identify r_{i,i+1}
    end if
  else if R_{i-1}(bf) = 0 & R_{i+1}(bf) = 1 then
    R_i(bf) = 1
    R_{i-1}(bf) = 1
    cooperate with R_{i-1} to identify r_{i,i-1}
  else if R_{i-1}(bf) = 1 & R_{i+1}(bf) = 0 then
    R_i(bf) = 1
    R_{i+1}(bf) = 1
    cooperate with R_{i+1} to identify r_{i,i+1}
  else
    stay idle
  end if
end while
```

VII. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed algorithms. We first investigate the performance of MRTI-BT algorithm for single reader and FSR algorithm in error free and noisy environments, respectively. Afterwards, we test our algorithm (MRTI-BT) in multi-reader under error free environment. We conduct simulations using Matlab to quantify the performance of each algorithm. The results are obtained by computing the average from 10^4 times of simulations.

The performance metric of interest in this paper is the total tag identification time defined as the total number of slots needed to identify n tags in the system. For an accurate measure of the identification time, assume that each bit is transmitted within a fixed time period. According to the

parameters set in [20], the reader's feedback on collision, readable or idle slot is 3 bits. In Season, all slots have fixed number of occupied bits, which is $b + 3$ bits for reader's feedback, where b is the length of the tag's ID [25]. In MRTI-BT, we run the identification process over three main phases. The number of occupied bits in *random number generation* phase is ignored, because the readers broadcast commands which are negligible compared to the other two phases; *collision handling* and *parallel identification* phases. The number of occupied bits in *collision handling* phase is as follows. An idle and a readable slot occupies x bits + 3 bits for reader's feedback, and a collision slot occupies x bits + 3 bits for feedback + $\lceil log_2\ x \rceil$ bits to inform the tag with the position of the first collided bit, where x is the number of bits sent by a tag in the *collision handling* phase. In *parallel identification* phase, readers may decode two tags in the same slot using *parallel identification property*. Thus, the number of occupied bits in a *parallel identification* slot is $(b - x + 3)$, where $(b - x)$ is the length of the ID fragment sent by the tags and the 3 bits are for reader's feedback. In OBTT, it identifies tags in one phase, so the length of all slots in $b + 3$ bits for feedback.

A. SIMULATION SETUP

We generated random unique IDs throughout the evaluation to ensure that all algorithms utilize the same input. In order to fairly evaluate our proposed schemes, we consider different simulation environments. In single reader error free environment, the problem of interest here is *tag collision* only. We compare the performance of our proposed MRTI-BT algorithm for single reader with that of the OBTT algorithm. Second, we consider noisy channel in single reader RFID system, and we compare between our proposed FSR and OBTT. We note that both MRTI-BT for single reader and FSR are modified versions of OBTT. The first aims to accelerate the identification time in error free environment. FSR also aims to speed up the identification time but in the existence of errors by recovering them. We also note that if probability of error is equal to zero, FSR and OBTT are identical, hence, they have the same performance. We use a simple error model, because our main purpose here is to evaluate the performance in the existence of false collided bit errors. For any received pattern, there is a p percent chance to be infected. We only consider single bit error and it is uniformly distributed over the pattern (i.e all bits belong to a pattern have equal chance). If the position of an infected bit coincidentally lies on the top of a collided bit, the error doesn't take place, and it is ignored. Future extension of our work will incorporate multi-bit errors. Third, we extend our system to multi-RFID readers system. Readers are equally distant forming a ring in a circular area as shown in Figure 3. Two groups of common tags and one group of exclusive tags residing within the interrogation area of each reader. Tags are distributed according to homogeneous two point Poisson distribution with intensity equal to λ over a circular deployment area. We restrict the number of common tags to be 10% of the total number of tags to convey a real life scenario. The number of common tags is usually small in comparison with the total number of tags. We choose an even number of readers, $N = 10$, in a ring topology for the following reason. In this case, Season [25] allows even or odd readers to identify common tags concurrently. As a result, Season needs one round only to identify common tags, which rarely occurs, instead of multiple rounds as it often does. Thus, we are comparing between MRTI-BT and Season's best scenario. The simulation parameters are summarized in Table 5.

TABLE 5. Simulation parameters.

System parameter	Value
N	10
n	$50 * 10^3, 51 * 10^3, ..., 60 * 10^3$ (5000)
λ	$10^4, 2 * 10^4, ..., 10^5$
x	10
b	128
p	0.2

B. SIMULATION RESULTS

1) SINGLE READER (ERROR FREE CHANNEL)

In Figure 9, we show the gains of identifying tags over two phases, *collision handling* and *parallel identification*, compared to single phase identification in OBTT. In addition, we show the gains of identifying tags by utilizing bit tracking compared to ALOHA that is used in Season. Moreover, we measure the effect of different x sent in *collision handling* phase on the total tag identification time. First, we note that the identification time monotonically increases with the number of tags, which agrees with intuition. Second, MRTI-BT considerably outperforms OBTT, with respect to the tag identification time. This gain increases with the number of tags and is attributed to the key feature of MRTI-BT, *collision handling* phase (illustrated earlier in Section IV), which reduces the number of occupied bits in a collision slot from 138 bits in OBTT [20] to only 17 bits in MRTI-BT ($x = 10$). Third, when $x = b$, the tags transmit their complete ID in a single phase, which is the same technique used in OBTT. Finally, MRTI-BT ($x = 10$) considerably outperforms Season by 100% on the average, with respect to the tag identification time.

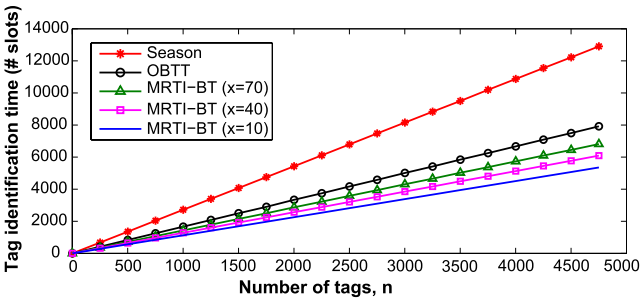

FIGURE 9. Single reader identification with different x.

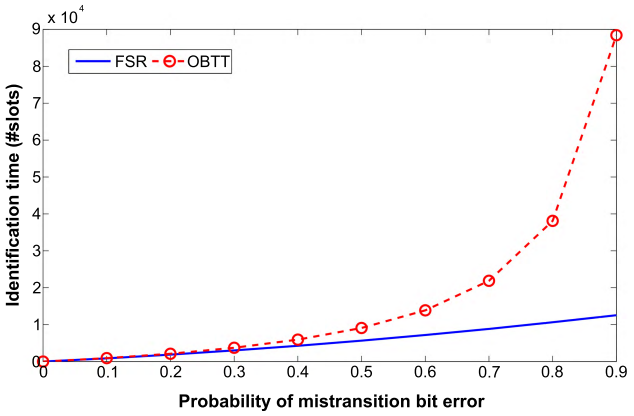

FIGURE 10. Comparison between FSR and OBTT with varying *p*.

2) SINGLE READER (NOISY CHANNEL)

In Figure 10, we show the gains of utilizing repetition gain to recover false collided bit errors in comparison with OBTT. First, we note that identification time is approximately monotonically increasing with p in FSR algorithm. On the contrary, identification time is exponentially increasing as p grows in OBTT. Thus, the gain increases as p increases. This is because the number of wasted slots is expanding due to false collided bit error, however, FSR utilizes the repetition gain and recovers a portion of these wasted slots. When FSR prevents a false collision, it saves two slots, collision and idle slots, so the exponential growth is not surprising. Finally we note that the identification time goes to infinity when $p = 1$ in OBTT curve, because the system considers every false collided bit error as collision. On the other hand, FSR recovers these false collided bits from the repetition that is occurring due to the nature of the identifying algorithm. In Figure 11, we show the gain of utilizing FSR. We measure the average number of slots per tag required for identification with varying n. As shown, OBTT approximately occupies 1.6 slot per tag on average. Recall that FSR in error free in environment ($p = 0$) has an identical performance to OBTT. In noise environment, FSR occupies 1.85 slot per tag on average, and OBTT with error occupies 2.15 slot per tag. FSR considerably outperforms OBTT with error as it decreases the identification time delay from 34% to only 16%. Recall that the gain increases as the probability of error increases as shown previously in Figure 10.

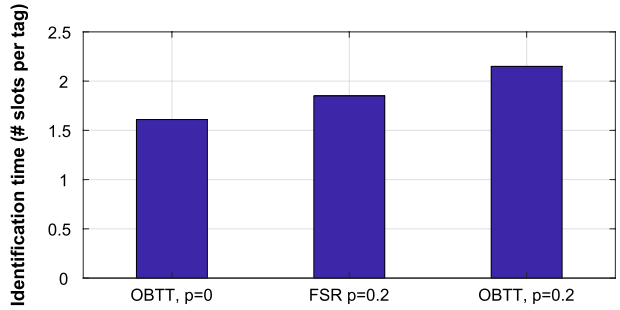

FIGURE 11. Comparison between FSR and OBTT with varying *n*.

As explained earlier, the numbers of wasted idle and collision slots due to false collided bit error are equal. This implies that the total numbers of saved idle and collision slots due to utilizing FSR features are the same. However, this does not appear clearly in Figure 12 and Figure 13. The gap between OBTT and OBTT with error in Figure 12 is smaller than the corresponding gap in 13. This is because the number of collision slots is larger than idle slots in OBTT (i.e when $p = 0$) [20]. Collision slots are not completely useless because FSR decodes some bits and they are used to correct future patterns, however, idle slots are profitless. We note that splitting tree optimization techniques can further decrease wasted idle slots [38]. We left this investigation for future work.

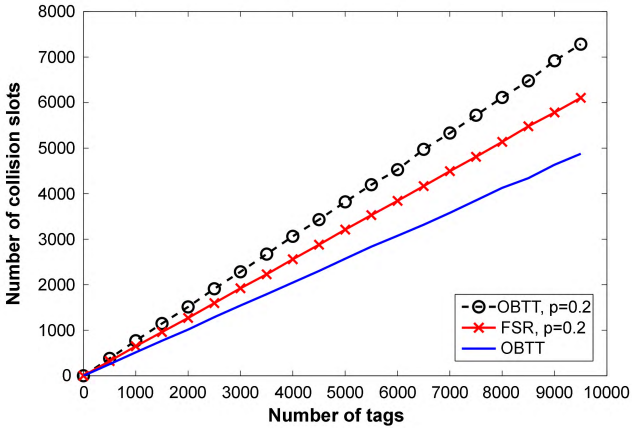

FIGURE 12. Collision slots in FSR and OBTT algorithms with varying n.

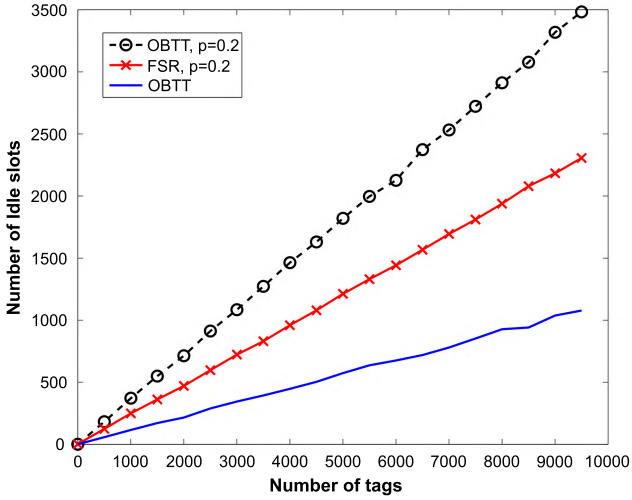

FIGURE 13. Idle slots in FSR and OBTT algorithms with varying n.

3) MULTI-READERS (ERROR FREE CHANNEL)

As shown in Figure 14, we compare between our MRTI-BT and Season. First, we note that the benefits increase

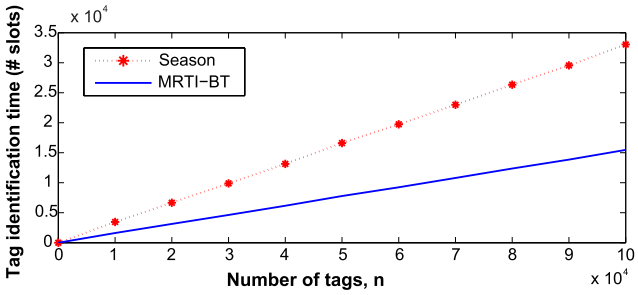

FIGURE 14. Comparison between MRTI-BT and Season in homogeneous two point Poisson distribution multi-reader RFID systems($x = 10$).

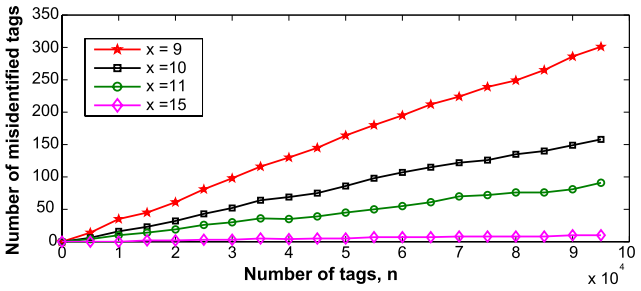

FIGURE 16. Number of misidentified tags with varying x.

as n increases. Second, we calculated the identification time gain. On average, we found that our algorithm outperforms Season by 113%. Since both MRTI-BT for single reader and MRTI-BT for multi-readers are utilized, the added gain from *parallel identification property* is only 13%. The distribution of the tags here does not provide high level of variation between the numbers of exclusive tags, so the property is not fully reaped.

In Figure 15, we study a more realistic scenario where the number of exclusive tags belonging to each reader varies. In order to achieve this, we set a constant number of exclusive tag while varying its distribution across the readers. However, the sum of the exclusive tags of all readers is constant (50,000 in this scenario). We also vary the number of common tags from 0 to 20% of the total number of exclusive tags. Without loss of generality, we make the numbers of exclusive tags residing within the interrogation area of each reader follow a Gaussian distribution with mean $\mu = 50,000/10$ and a changing variance σ^2. This experiment provides a good variation, and hence can measure the performance in more realistic scenarios. The number of common tags is shown on the $x - axis$ and the total identification time required for identifying common tags is on the $y - axis$. We meant to show the identification time of common tags only because the benefits of our proposed property might not appear clearly in the total identification time curve. First, we note that Season has constant performance that does not change with the alteration of σ^2. This is because Season waits for the reader with the maximum number of tags before it identifies common tags sequentially. On the other hand, our approach identifies them in parallel with exclusive tags which incurs lower

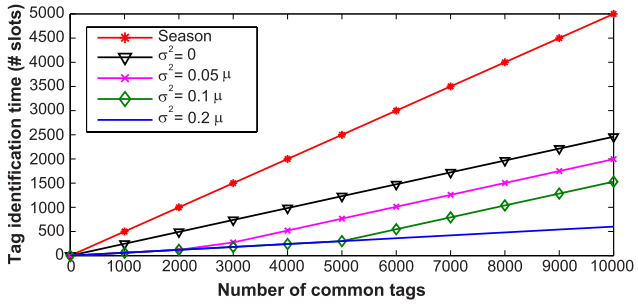

FIGURE 15. Comparison of MRTI-BT and Season with varying σ^2($x = 10$).

identification time. Second, we measure the performance of our algorithm with different σ^2. As shown, when $\sigma^2 = 0$, *parallel identification property* is not reaped. This is because the readers have the same number of tags, and hence they finish identifying their tags at the same time. Thus, the system cannot identify a common tag and an exclusive tag concurrently using *parallel identification property*. Although this is the most challenging scenario for MRTI-BT, it outperforms Season by 100% on average with respect to the identification time, which is equal to the gain of single reader identification. *Parallel identification property* is reaped when $\sigma^2 > 0$. In the beginning, the identification time increases gradually because the common tags are identified in parallel with exclusive tags and hence, the shown identification time is the cost of sending the first x bits only. Afterwards, the system becomes unable to apply the property, because the reader with maximum number of tags finishes and hence the the shown identification time is the cost of sending b bits. It is also clear that the identification time decreases as the variation, σ^2, increases. Finally, the system identifies all common tags with *parallel identification property*, which is the best case, when $\sigma^2 = 0.2\mu$. In other words, all common tags are identified while the reader with maximum number of exclusive tags is identifying its exclusive tags.

In our proposed scheme, the first x bits of each ID are received for collision resolution. According to (2), there is a probability that two or more tags pick the same slot after *collision handling* phase. This leads to identification errors in the *parallel identification* phase. We define misidentified tags as the tags that are assigned the same slot after the *collision handling* phase. In Figure 16, we show the number of misidentified tags in the system with varying x. First, we note that the number of misidentified tags is small in comparison with n. Moreover, the number of misidentified tags decreases as x increases which agrees with intuition. Finally, we note that the number of misidentified tags is almost equal to zero when x is large enough (i.e $x = 15$). Thus, MRTI-BT is able to identify n tags with high accuracy without sacrificing the identification time.

VIII. CONCLUSION

Bit tracking anti-collision algorithms have proven their positive impact on the identification time. In this work, we extended their usage in two avenues. First, we introduced

and evaluated a bit error type called *false collided bit error* which contributes to significant delays in the identification time. We also demonstrated the relationship between the position of the false collided bit error and the identification delay; it does not affect the identification process if its location is after the location of the first real collided bit in a collision slot. In addition, we proposed a zero overhead novel algorithm that tackles false collided bit errors. Second, we designed a novel algorithm that accelerates the identification process in ''error free'' multi-reader RFID systems. We proposed *parallel identification property* that allows identifying common tags in parallel with exclusive tags. If a reader finishes identifying its exclusive tags, it cooperates with a neighbour reader to identify the shared (common) tags while the neighbour is identifying its exclusive tags. We also proposed a short length ID for resolving random number contention and identify the full ID in a collision free manner. The incorporated techniques outperform the state-of-the-art and reduce the identification time significantly.

ACKNOWLEDGMENT

The findings achieved herein are solely the responsibility of the authors.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, ''The Internet of Things: A survey,'' *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] A. Mitrokotsa and C. Douligeris, ''Integrated RFID and sensor networks: Architectures and applications,'' in *RFID Sensor Networks: Architectures, Protocols, Security Integrations*. Boca Raton, FL, USA: CRC Press, 2009, ch. 18, pp. 511–535.

[3] S. López-Soriano and J. Parrón, ''Wearable RFID tag antenna for healthcare applications,'' in *Proc. IEEE-APS Topical Conf. Antennas Propag. Wireless Commun. (APWC)*, Sep. 2015, pp. 287–290.

[4] S. Sambanthan and V. Saravanan, ''RFID based pervasive tracker for physically challenged,'' in *Proc. Int. Conf. Innov. Inf., Embedded Commun. Syst. (ICIIECS)*, 2015, pp. 1–5.

[5] D.-H. Shih, P.-L. Sun, D. C. Yen, and S.-M. Huang, ''Taxonomy and survey of RFID anti-collision protocols,'' *Comput. Commun.*, vol. 29, no. 11, pp. 2150–2166, 2006.

[6] Y. Fukumizu, S. Ohno, M. Nagata, and K. Taki, ''Communication scheme for a highly collision-resistive RFID system,'' *IEICE Trans. Fundamentals Electron., Commun. Comput. Sci.*, vol. E89-A, no. 2, pp. 408–415, 2006.

[7] L. Kong, L. He, Y. Gu, M.-Y. Wu, and T. He, ''A parallel identification protocol for RFID systems,'' in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 154–162.

[8] J. Ou, M. Li, and Y. Zheng, ''Come and be served: Parallel decoding for COTS RFID tags,'' in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 500–511.

[9] H.-C. Liu and J.-P. Ciou, ''Performance analysis of multi-carrier RFID systems,'' in *Proc. Int. Symp. Perform. Eval. Comput. Telecommun. Syst. (SPECTS)*, vol. 41. 2009, pp. 112–116.

[10] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, ''Efficient and reliable low-power backscatter networks,'' in *Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2012, pp. 61–72.

[11] C. Mutti and C. Floerkemeier, ''CDMA-based RFID systems in dense scenarios: Concepts and challenges,'' in *Proc. IEEE Int. Conf. RFID*, Apr. 2008, pp. 215–222.

[12] Z. Zhang, Z. Lu, Q. Chen, and X. Yan, ''Design and optimization of a CDMA-based multi-reader passive UHF RFID system for dense scenarios,'' *IEICE Trans. Commun.*, vol. 95, no. 1, pp. 206–216, 2012.

[13] V. Liu, V. Talla, and S. Gollakota, ''Enabling instantaneous feedback with full-duplex backscatter,'' in *Proc. 20th Annu. Int. Conf. Mobile comput. Netw.*, 2014, pp. 67–78.

[14] E. Vahedi, R. K. Ward, and I. F. Blake, ''Performance analysis of RFID protocols: CDMA versus the standard EPC Gen-2,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1250–1261, Oct. 2014.

[15] Z. Zhang, Z. Lu, Q. Chen, X. Yan, and L.-R. Zheng, ''Code division multiple access/pulse position modulation ultra-wideband radio frequency identification for Internet of Things: Concept and analysis,'' *Int. J. Commun. Syst.*, vol. 25, no. 9, pp. 1103–1121, 2012.

[16] S.-R. Lee, S.-D. Joo, and C.-W. Lee, ''An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification,'' in *Proc. 2nd Annu. Int. Conf. Mobile Ubiquitous Syst., Netw. Services (MobiQuitous)*, 2005, pp. 166–172.

[17] M. A. Bonuccelli, F. Lonetti, and F. Martelli, ''Tree slotted aloha: A new protocol for tag identification in RFID networks,'' in *Proc. Int. Symp. World Wireless, Mobile Multimedia Netw.*, 2006, pp. 603–608.

[18] J. Myung, W. Lee, J. Srivastava, and T. K. Shih, ''Tag-Splitting: Adaptive collision arbitration protocols for RFID tag identification,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 6, pp. 763–775, Jun. 2007.

[19] J. H. Choi, D. Lee, and H. Lee, ''Query tree-based reservation for efficient RFID tag anti-collision,'' *IEEE Commun. Lett.*, vol. 11, no. 1, pp. 85–87, Jan. 2007.

[20] Y. C. Lai, L.-Y. Hsiao, and B.-S. Lin, ''Optimal slot assignment for binary tracking tree protocol in RFID tag identification,'' *IEEE/ACM Trans. Netw.*, vol. 23, no. 1, pp. 255–268, Feb. 2015.

[21] Y.-C. Lai, L.-Y. Hsiao, H.-J. Chen, C.-N. Lai, and J.-W. Lin, ''A novel query tree protocol with bit tracking in RFID tag identification,'' *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 2063–2075, Oct. 2013.

[22] J. Waldrop, D. W. Engels, and S. E. Sarma, ''Colorwave: An anticollision algorithm for the reader collision problem,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, vol. 2. May 2003, pp. 1206–1210.

[23] E. EPCglobal, ''Radio-frequency identity protocols generation-2 UHF RFID specification for RFID air interface protocol for communications at 860 MHz–960 MHz, version 2.0. 1 ratified,'' in *Proc. EPCglobal*, 2013, pp. 22–25, paper. GS1.

[24] A. Fahim and T. ElBatt, ''Multi-reader RFID tag identification using bit tracking (MRTI-BT),'' in *Proc. IEEE Int. Conf. RFID*, May 2016, pp. 1–7.

[25] L. Yang, J. Han, Y. Qi, C. Wang, T. Gu, and Y. Liu, ''Season: Shelving interference and joint identification in large-scale RFID systems,'' in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 3092–3100.

[26] C. A. Albright, S. A. Kaiser, L. W. Oglesbee, and D. W. Engels, ''Forward error correction in passive UHF Gen2 communications,'' in *Proc. IEEE Int. Conf. RFID*, Apr. 2015, pp. 17–24.

[27] C. Abraham *et al.*, ''Inventory management using passive RFID tags: A survey,'' Dept. Comput. Sci., The Univ. Texas at Dallas, Richardson, TX, USA, Tech. Rep., 2002, pp. 1–16.

[28] D. R. Hush and C. Wood, ''Analysis of tree algorithms for RFID arbitration,'' in *Proc. IEEE Int. Symp. Inf. Theory*, Aug. 1998, p. 107.

[29] K. Ali, H. Hassanein, and A.-E. M. Taha, ''RFID anti-collision protocol for dense passive tag environments,'' in *Proc. 32nd IEEE Conf. Local Comput. Netw. (LCN)*, Oct. 2007, pp. 819–824.

[30] J. Yin, Y.-G. He, B. Li, X. Deng, Y.-H. Tan, and Y.-Q. Xiao, ''RFID anti-collision algorithm based on grouping dynamic frame slotted,'' *Comput. Eng.*, vol. 20, no. 1000-3428, p. 097, 2009.

[31] C. Law, K. Lee, and K.-Y. Siu, ''Efficient memoryless protocol for tag identification,'' in *Proc. 4th Int. Workshop Discrete Algorithms Methods Mobile Comput. Commun.*, 2000, pp. 75–84.

[32] Y. Kim, S. Kim, S. Lee, and K. Ahn, ''Improved 4-ary query tree algorithm for anti-collision in RFID system,'' in *Proc. Int. Conf. Adv. Inf. Netw. Appl.*, 2009, pp. 699–704.

[33] H. Landaluce, A. Perallos, E. Onieva, L. Arjona, and L. Bengtsson, ''An energy and identification time decreasing procedure for memoryless RFID tag anticollision protocols,'' *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 4234–4247, Jun. 2016.

[34] H. Landaluce, A. Perallos, and I. J. G. Zuazola, ''A fast RFID identification protocol with low tag complexity,'' *IEEE Commun. Lett.*, vol. 17, no. 9, pp. 1704–1706, Sep. 2013.

[35] S. Zhou, Z. Luo, E. Wong, C. J. Tan, and J. Luo, ''Interconnected RFID reader collision model and its application in reader anti-collision,'' in *Proc. IEEE Int. Conf. RFID*, Mar. 2007, pp. 212–219.

[36] S. M. Birari and S. Iyer, ''Mitigating the reader collision problem in RFID networks with mobile readers,'' in *Proc. 13th IEEE Int. Conf. Netw., IEEE 7th Malaysia Int. Conf. Commun.*, vol. 1. Nov. 2005, p. 6.

[37] A.-H. Mohsenian-Rad, V. Shah-Mansouri, V. W. Wong, and R. Schober, "Distributed channel selection and randomized interrogation algorithms for large-scale and dense RFID systems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 4, pp. 1402–1413, Apr. 2010.
[38] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data Networks*, vol. 2. Englewood Cliffs, NJ, USA: Prentice-Hall, 1992.
[39] Z. Zhou, B. Chen, and H. Yu, "Understanding RFID counting protocols," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 312–327, Feb. 2016.
[40] M. Shahzad and A. X. Liu, "Fast and accurate estimation of RFID tags," *IEEE/ACM Trans. Netw.*, vol. 23, no. 1, pp. 241–254, Feb. 2015.
[41] D.-J. Deng, C.-C. Lin, T.-H. Huang, and H.-C. Yen, "On number of tags estimation in RFID systems," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1395–1402, Sep. 2015.
[42] K. Domdouzis, B. Kumar, and C. Anumba, "Radio-frequency identification (RFID) applications: A brief introduction," *Adv. Eng. Informat.*, vol. 21, no. 4, pp. 350–355, 2007.
[43] S. Basagni, "Finding a maximal weighted independent set in wireless networks," *Telecommun. Syst.*, vol. 18, no. 1, pp. 155–168, 2001.
[44] D. K. Klair, K.-W. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 3, pp. 400–421, 3rd Quart., 2010.

ABDULRAHMAN FAHIM received the B.Sc. degree in electrical engineering from Nile University, Egypt, in 2014, and the M.Sc. degree in wireless communications from the Wireless Intelligent Networks Center, Nile University, Egypt, in 2016. He is currently pursuing the Ph.D. degree in computer science with the University of California at Riverside, Riverside, CA, USA. After completing his B.Sc., he joined the Wireless Intelligent Networks Center, Nile University, in 2014, as a Research Assistant. His research interests are wireless communication, RFID networks, indoor localization, and video processing.

TAMER ELBATT (SM'06) received the B.S. and M.S. degrees in electronics and communications engineering from Cairo University, Egypt, in 1993 and 1996, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Maryland at College Park, College Park, MD, USA, in 2000. From 2000 to 2009, he was with major U.S. industry research and development laboratories, e.g., HRL Laboratories, LLC, Malibu, CA, USA, and Lockheed Martin ATC, Palo Alto, CA, USA, at various positions. In 2009, he joined the Electronics and Communications Department, Faculty of Engineering, Cairo University, Egypt, as an Assistant Professor, where he is currently an Associate Professor. He has been holding a joint appointment with Nile University, Egypt, since 2009 and has been serving as the Director of the Wireless Intelligent Networks Center since 2012. He was a Visiting Professor with the Department of Electronics, Politecnico di Torino, Italy, in 2010, FENS, with Sabanci University, Turkey, in 2013, and with the Department of Information Engineering, University of Padova, Italy, in 2015. He has published over 95 papers in prestigious journals and international conferences. He holds seven issued U.S. patents. His research has been supported by the U.S.DARPA, ITIDA, QNRF, FP7,General Motors, Microsoft, and Google, and is currently being supported by NTRA, H2020, and Vodafone Egypt Foundation. His research interests lie in the broad areas of performance analysis, design, and optimization of wireless and mobile networks. He was a recipient of the 2014 Egypt's State Incentive Award in engineering sciences, the 2012 Cairo University Incentive Award in engineering sciences, and the prestigious Google Faculty Research Award in 2011. He has served on the technical program committees of numerous IEEE and ACM conferences. He served as the Demos Co-Chair of the ACM Mobicom 2013 and the Publications Co-Chair of the IEEE Globecom 2012 and EAIMobiquitous 2014. He currently serves on the Editorial Board of the IEEE TRANSACTIONS ON MOBILE COMPUTING and *International Journal of Satellite Communications and Networking* (Wiley).

AMR MOHAMED received the M.S. and Ph.D. degrees in electrical and computer engineering from The University of British Columbia, Vancouver, Canada, in 2001 and 2006 respectively. He was an Advisory IT Specialist with the IBM Innovation Centre, Vancouver, from 1998 to 2007, taking a leadership role in systems development for vertical industries. He has over 20 years of experience in wireless networking research and industrial systems development. He is currently an Associate Professor with the College of Engineering, Qatar University, and the Director of the Cisco Regional Academy. He has authored or co-authored over 120 refereed journal and conference papers, textbook, and book chapters in reputed international journals, and conferences. His research interests include networking and MAC layer techniques mainly in wireless networks. He holds three awards from IBM Canada for his achievements and leadership, and three best paper awards, latest from the IEEE/IFIP International Conference on New Technologies, Mobility, and Security 2015, Paris. He has served as the technical program committee co-chair for workshops in the IEEE WCNC'16. He has served as the co-chair for technical symposia of international conferences, including Globecom'16, Crowncom'15, AICCSA'14, IEEE WLN'11, and IEEE ICT'10. He has served on the organization committee of many other international conferences as a TPC member, including the IEEE ICC, GLOBECOM, WCNC, LCN, and PIMRC, and a technical reviewer for many international IEEE, ACM, Elsevier, Springer, and Wiley journals.

ABDULLA AL-ALI received the master's degree in software design engineering and the Ph.D. degree in computer engineering from Northeastern University, Boston, MA, USA, in 2008 and 2014, respectively. He is currently an Active Researcher in cognitive radios for smart cities and vehicular ad-hoc networks. He is also the Head of the Technology Innovation and Engineering Education with the College of Engineering, Qatar University. He has published a number of peer-reviewed papers in journals and conferences. He received the Platinum medal in the Educational Excellence Day Prize for Ph.D. holders in 2015.