Contents

Chapter X

Anti-Collision Protocols for the RFID System

Jehn-Ruey Jiang and Ming-Kuei Yeh

In the RFID system, tags store unique identifications and are attached to objects; a reader performs the *tag interrogation* procedure to recognize an object by issuing wireless RF signals to interrogate the identification of the attached tag. Like other wireless communication systems, the RFID system also suffers from the signal interference problem. There are two types of signal interference. One is called the *reader collision*, which occurs when multiple readers issue signals to same tags simultaneously. The other is called the *tag collision*, which occurs when multiple tags respond to a reader simultaneously. Collisions hinder and slow down the tag interrogation procedure. Therefore, *reader anti-collision* and *tag anti-collision* protocols are required to respectively reduce reader collisions and tag collisions for improving interrogation procedure performance. In this chapter, we introduce existent reader anti-collision and tag anti-collision protocols. We intend to provide not only an extensive survey of the protocols, but also new research directions of them.

X.1 Introduction

The front-end of a RFID system is composed of two components: readers and tags [1]. Tags store unique identifications and are attached to objects; a reader performs the *tag interrogation* procedure to recognize an object by issuing wireless RF signals to interrogate the identification (ID) of the attached tag. Since tags are designed for an attempt of world-wide deployment in commercial or alike applications, they are supposed to be tiny, low cost and equipped with a simple circuit of limited computation and communication capabilities [2]. Most RFID tags are passive; they do not have on-tag power source and derive energy from the RF field generated by the reader to drive the circuit. When a tag and a reader are close enough, they can communicate with each other. For such a situation, we say that the tag is in the *interrogation zone* of the reader. Like other wireless communication systems, the RFID system also suffers from the signal interference problem [3]. There are two types of signal interference. One is called the *reader collision*, which occurs when multiple readers issue signals to same tags simultaneously. The other is called the *tag collision*, which occurs when multiple tags respond to a reader simultaneously. Collisions hinder and slow down the tag interrogation procedure. Therefore, *reader anti-collision* and *tag anti-collision* protocols are thus required to respectively reduce reader collisions and tag collisions for improving interrogation procedure performance. In this chapter, we introduce existent reader anti-collision and tag anti-collision protocols. We intend to provide not only a comprehensive survey of the protocols, but also new research directions of them.

Because the tag is energized by the reader, the tag's response range (also called the *interrogation range*) is much less than the reader's RF transmission range (also called the *interference range*) [4]. Furthermore, tags and readers have very different computation powers and communication capabilities. Due to all the asymmetries, we cannot rely on common collision avoidance mechanisms, such as the RTS/CTS mechanism used in wireless local area network [3], to solve the collision problem.

Some reader anti-collision protocols are proposed to reduce reader collisions based on the concepts of TDMA (Time Division Multiple Access), FDMA (Frequency Division Multiple Access) or CSMA (Carrier Sense Multiple Access) [5]. TDMA-based reader anti-collision protocol divides the transmission time into intervals and a reader can only transmit messages in its assigned intervals. The assignment of intervals can be done in a distributed or a centralized way [6]. Waldrop et. al [7] proposed two distributed TDMA-based reader anti-collision protocols, called DCS (Distributed Color Selection) and Colorwave. A reader graph is first derived, where any two readers are defined to be adjacent and have an edge between them if they may interfere with each other. Each reader is assigned a color which stands for a reservation of a specific timeslot for transmitting signals. If all the adjacent readers are with different colors, the reader collision is avoided. In DCS protocol, the maximum number of colors (max_colors) is fixed, and a reader transmits only in its assigned color (timeslot). On the contrary, Colorwave protocol has dynamic values of max_colors; it is a dynamic color assignment mechanism to minimize the required number of colors in the reader graph. With the reduction in the number of used colors, the efficiency of message transmission is increased.

FDMA-based protocols divide all available frequency bands into several non-interfering frequency channels. If a frequency channel is only assigned to a transmitter at a time, transmitters can transmit messages simultaneously without causing any interference. Ho et. al proposed HiQ [8], which is a both TDMA-based and FDMA-based protocol. It attempts to minimize reader collisions by learning the collision patterns of the readers and by effectively assigning frequencies over time. HiQ depends on a distributed, hierarchical and online learning scheme called *Q-learning* for determining frequency and time assignments. By interacting repeatedly with the system, Q-learning attempts to discover an optimum frequency assignment over time. EPCGlobal Gen 2 [9] is a famous protocol that adapts FDMA technology to solve the reader collision problem. Readers can choose separate transmission channels to avoid interference by the frequency hopping spread spectrum technique.

CSMA is another mechanism used to solve the reader collision problem. In CSMA mechanism, each reader needs to check before transmitting messages whether the carrier (the shared communication channel) is free or not. If the carrier is sensed to be idle, the reader sends out messages at once. Otherwise, the reader delays a random period of time and then starts sensing carrier again. The ETSI (European Telecommunications Standards Institute) EN 302 208 Standard [10] utilizes "*Listen Before Talk (LBT)"* mechanism that based on the concept of CSMA to solve the reader-collision problem.

Several tag anti-collision protocols are proposed to reduce tag collisions. They can be categorized mainly into three classes: ALOHA-based, tree-based and counter-based protocols [11]. The ALOHA [12], slotted ALOHA [13]_and frame slotted ALOHA [14] protocols are ALOHA-based protocols. In ALOHA protocol, a reader first sends a command to make tags transmit their IDs. On receiving the reader's interrogation signal, each tag in the interrogation zone independently waits for a random back-off time and then responds its tag ID to the reader. If no collision occurs during a tag's ID response, its ID can be identified properly. In slotted ALOHA protocol, the random back-off time must be a multiple of a pre-specified slot time. Frame slotted ALOHA protocol is similar to slotted ALOHA protocol except that the whole interrogation procedure is divided into a set of frames each having a fixed number of time slots, and a tag can send its ID to the reader only in one randomly chosen slot during a frame period. ALOHA-based protocol is simple but has the tag starvation problem that a tag may never be identified properly for the reason that its responses always collide with others'.

The basic idea of tree-based protocols [15][16][17][18][19] is to repeatedly split the tags encountering collisions into subgroups until there is only one tag in a subgroup to be identified successfully. The protocols can be applied to tags with or without writable memory. Tags with memory have higher cost. However, protocols for such a kind of tags have better performance. The query tree protocol [16] is applicable to tags without on-tag writable memory. In the protocol, the reader broadcasts a request bit string S with a variable length to tags. A tag with an ID prefix matching S will respond its ID to the reader. When collisions occur, the reader broadcasts again with a longer bit string S0 or S1 to split colliding tags into two subgroups. The bit-by-bit binary tree [15] is applicable for tags with writable memory. In the protocol, a reader broadcasts a request command first and each tag will respond the first bit of its tag ID. If collisions occur, the reader will acknowledge the tags with 0 (or 1). Only the tag with the first bit being 0 (or 1) will respond the next bit to the reader. In this way, the tags are continuously split into two groups. The other tree-based protocols, such as EPCglobal Class 0 [19], the tree slotted ALOHA (TSA) [17], BSQTA [18] and BSCTTA [18] protocol, also utilize similar concept to split tags to solve the tag collision problem. The main drawback of tree-based protocols is that their performances are affected by the length or the distribution of tag IDs. In general, tree-based protocol has longer identification time latency than that of ALOHA-based

[20], but it does not have the tag starvation problem.

The concept of counter-based protocols [11][20][21][22][23] is similar to that of tree-based protocols. The major difference between these two kinds of protocols is that the former rely on static tag IDs for the splitting, and the latter rely on dynamically changing counters for the splitting. ISO/IEC 18000-6B [22] is a standard adopting the counter-based tag anti-collision protocol. In ISO/IEC 18000-6B, each tag has a counter initially set to 0. When a reader sends request to tags, every tag with counter value 0 can transmit its tag ID to the reader. When a collision occurs, the tags with counters of values greater than 0 then increases their counters by 1, while the tags with counter value 0 randomly generates a random bit, 0 or 1, and add it to their counters. In this way, the tags with counters value 0 are split into two subgroups. Other counter-based protocols, such as ABS protocol [30], utilize similar concept to split tags encountering collisions. The counter-based protocols do not have the starvation problem. Furthermore, they have the stable property that their performances are not affected by the length of tag IDs or the distribution of tag IDs.

The rest of this chapter is organized as follows. In section 2, collision problems are defined first. And in section 3, reader anti-collision protocols, like TDMA, FDMA and CSMA protocols, are described in detail. And tag anti-collision protocols, including ALOHA-, tree-, and counter-based protocols, are elaborated in section 4. In both section 3 and section 4, examples are further given for some protocols to make them easy to understand. At last, we give a summary and suggestions of new research directions in section 5.

X.2 Collision problems in the RFID system

When a reader (or called interrogator) transmits a request to a tag, it also provides energy to power-up a passive tag. If the reader and the passive tag are close enough, the reader can receive the signal reflected from the tag. For such a situation, we say that the tag is in the interrogation zone of the reader. When two or more readers are too close or many tags appear in one reader's interrogation zone, there arise interference problems, which are mainly classified as the reader collision problem and the tag collision problem. Below, we describe the two types of problems.

- The reader collision (or reader interference) problem:

  Because the tag is energized by the reader, the tag's response zone (i.e., the interrogation zone) is much less than the reader's transmission zone (also called interference zone). When a tag is within the interrogation zone of a reader A and within the interference zone of another reader B. Due to the interference of readers, either the tag cannot receive the request command from reader A correctly or reader A cannot interpret the response from the tag properly. This is called the reader collision problem. For example, in Fig. 1, tag T is within the interrogation zone of reader A and within the interference zone of reader B. The reader collision problem occurs for such a situation.

- Tag collision problem:

  To identify tags within the interrogation zone, a reader sends a request to ask tags to send back their IDs. When multiple tags within the reader's interrogation zone respond to the request simultaneously, collision occurs and the reader cannot identify any tag properly. This is called the tag collision problem. For example, in Fig. 1, tags S and T are within the interrogation zone of reader A. If tags S and T send their IDs for responding to reader A's request simultaneously, the tag collision problem occurs and neither tag can be recognized by reader A.

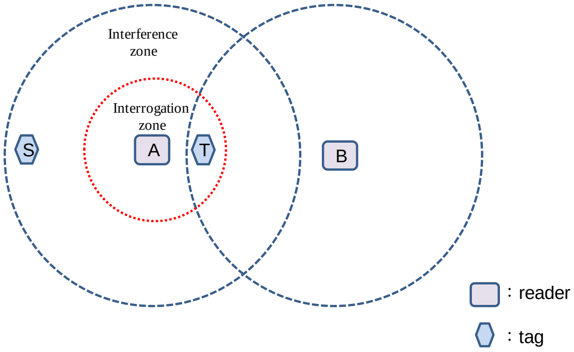


Figure 1. The relation between the interrogation zone and the interference zone

X.3 Reader anti-collision protocols

Several reader anti-collision protocols are proposed to solve the reader collision problem. They are classified into three classes: TDMA (Time Division Multiple Access), FDMA (Frequency Division Multiple Access) and CSMA (Carrier Sense Multiple Access) protocols [5]. Below, we describe some reader anti-collision protocols class by class.

X.3.1 TDMA (Time Division Multiple Access) protocols

The basic idea of TDMA-based reader anti-collision protocols is to divide the whole time period into intervals and to allow a reader to transmit message only within its allocated intervals. In this way, the reader collision can be avoided. Below, we introduce two TDMA-based reader anti-collision protocols: Distributed Color Selection (DCS) and Colorwave algorithms [7].

X.3.1.1 DCS algorithm

Distributed Color Selection (DCS) is a reader anti-collision protocol proposed by Waldrop et. al in [7]. Time slots are assume to be colored by colors 0, 1,…, maxColors cyclically. DCS solves the reader collision problem by first deriving a reader graph, where readers are represented as nodes and two nodes (readers) are defined to be adjacent and have an edge between them if they may interfere with each other. It then assigns each reader a color which stands for a reservation of a specific time slot for transmitting signals. If all the adjacent readers are of different colors, the reader collision is avoided.

DCS is a distributed algorithm that allows each reader to randomly and locally choose a color (time slot) from color set {0, 1,…, maxColors}, where maxColors is an input parameter whose value will never change. When a reader wants to send a message to tags, it will queue the message until the time slot of the chosen color arrives. If a reader transmits message in the time slot of its chosen color but finds that collisions occur, it will re-choose a new color and notify all of its neighbors to change their chosen colors accordingly. Note that, DCS algorithm needs to synchronize the timing of timeslots but needs not to synchronize the value of colors among all readers in the system.

X.3.1.2 Colorwave algorithm

Colorwave algorithm, or called Variable-Maximum Distributed Color Selection (VDCS) algorithm, is an extension of the DCS algorithm. In Colorwave, a mechanism is proposed to optimize the number of colors (i.e., maxColors) required to color the reader graph. If the used colors are reduced, the efficiency of signal transmission can be improved.

When a reader observes by itself or is notified by neighboring readers that the successful transmission rate is below an addition_maxColors threshold, it will increase its local maxColors value and broadcasts the new maxColors to its neighboring readers to make them reselect colors in order to reduce the transmission collisions. On the contrary, a reader will decrease its local maxColors value to decrease the transmission waiting time when the successful transmission rate is above a subtraction_maxColors threshold.

X.3.2 . FDMA (Frequency Division Multiple Access) protocols

FDMA (Frequency Division Multiple Access) protocols divide all available frequency bands into several non-interfered channels. Readers can use different channels to communicate with tags simultaneously. Below, we introduce two protocols, HiQ [8] and the EPCglobal Gen

2 [9], that adapt the FDMA mechanism to solve the reader collision problem.

X.3.2.1 HiQ protocol

HiQ [8] is a hierarchical, distributed and online learning algorithm based on TDMA and FDMA to solve the reader collision problem. The designed goal is to maximize the number of concurrent communication channels between readers and tags while minimizing the number of reader collisions by learning the collision patterns of readers to assign frequencies in each time slot to the readers effectively.

The hierarchical control structure of HiQ consists of readers, R-servers, and Q-servers, as shown in Fig.2. RFID readers are at the lowest tier and each server in R-server tier manages several readers. When a reader needs to send messages to the tags in its interrogation zone, it must request resources, namely the frequency channel and the time slot, from its master R-server. The reader can send messages at a specific frequency channel in a time slot only after the channel and time slot are granted by its master R-server.

With the distributed architecture, the neighboring readers are possible to send messages in the same time slot or in the same frequency channel to cause collisions. It is the responsibility of readers to detect collisions with neighboring readers. Each reader should report the number of collisions and, type of collisions and the number of successful reads to its master R-server. The R-server can then determine which slave readers are interfering mutually by the feedback reports and reallocates the resources dynamically in order to avoid the collisions.

The resources that the R-server can allocate are from its master Q-server (Q-learning server) in the hierarchical structure. For greater flexibility and scalability, Q-Servers may themselves work in a hierarchical architecture. But there is always only one root Q-Server in the whole system that has the power of full control over the allocation of all frequency channels and time slots.

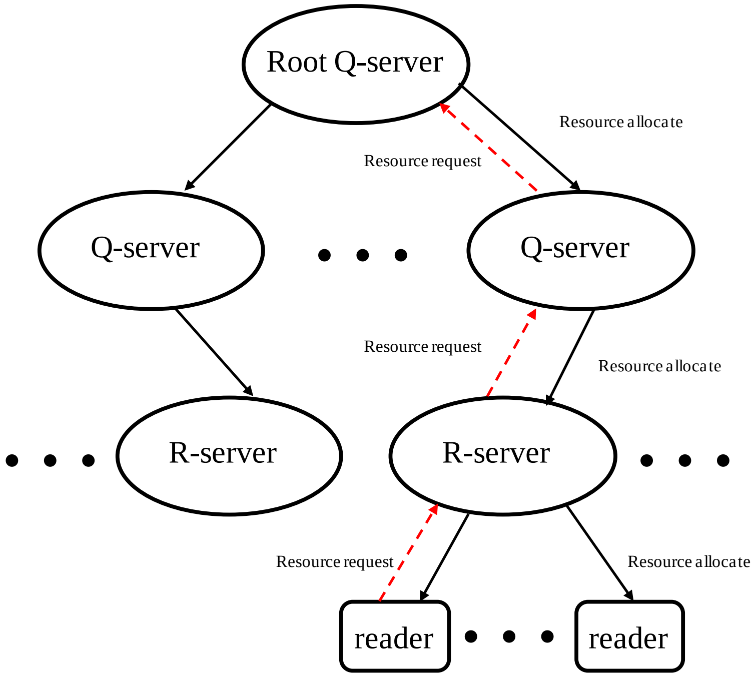


Figure 2. The hierarchical control structure of HiQ protocol

X.3.2.2 EPCglobal Gen 2 protocol

The Class 1 Generation 2 UHF standard [9] proposed by EPCglobal uses FDMA technology to reduce reader interference. The entire allocated frequency band is divided into channels. A reader will only use a certain channel for communication. The carrier frequency used by readers and tags are separate. That is, readers (resp., tags) will collide with readers (resp., tags) only. Readers use frequency hopping spread spectrum technique to avoid interference. In Europe, a bandwidth of 200 kHz is regulated for frequency allocation [24]. It is suggested that readers use even-numbered channels while tags backscatter signals in odd-numbered channels. In USA, a wider bandwidth of 500 KHz is regulated for frequency allocation. All channels are available for reader interrogation but the tag can backscatter signals at the boundaries of these channels. EPCglobal Gen 2 protocol can solve the reader collision problem. Because most low cost tags do not have frequency selection capability, the tag collision problem still exists [3].

X.3.3 . CSMA (Carrier Sense Multiple Access) protocols

CSMA (Collision Sensing Multiple Access) is a common mechanism used in wired or wireless systems to avoid collisions. In this mechanism, each device needs to check whether the media channel is free before transmitting messages. If the media is occupied, the device will wait until it is released.

ETSI (European Telecommunications Standards Institute) 302 208 is a European regulation that adopts a CSMA mechanism called "*Listen Before Talk (LBT)*" to solve the reader collision problem. It allocates the frequency band of 865 to 868 MHz for RFID applications [10] [24] and divides the band into 15 channels, each of 200 kHz bandwidth. With the maximum effective radiation power (ERP) of 2W, only 10 channels are available for communication and 5 channels are defined as guard bands or reserved for lower power readers. The receiver module of a reader is first activated to monitor selected channel for a specified time period (5 ms) before transmission. If it senses that the channel is idle over the specified time period, the reader can send the message directly for up to 4s and then the reader activates the receiver module to detect signal interference. If the channel is occupied by other readers, the reader will search for another free channel for transmitting messages.

X.4 Tag anti-collision protocols

Several tag anti-collision protocols are proposed for reducing tag collisions. They can be categorized into three classes: ALOHA-based, tree-based and counter-based protocols [11]. Below, we introduce some of the protocols class by class.

X.4 .1 ALOHA-based protocols

ALOHA-based tag anti-collision protocols [21] [25] [26] [27] are based on the time-division multiple access (TDMA) mechanism that operates in a probabilistic manner. They try to stagger the response times of tags in the interrogation zone. Below, we introduce several ALOHA-based protocols: ALOHA [12], slotted ALOHA [13] and frame slotted ALOHA [14].

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

X.4 .1.1 ALOHA protocol

ALOHA protocol [12] is the simplest ALOHA-based tag anti-collision protocol. When a reader requests tags to respond their IDs, each tag in the interrogation zone chooses a random back-off time individually and responds its tag ID to the reader after the back-off time. If no collision occurs during the transmission of a tag ID, this ID is identified successfully and acknowledged by the reader. A tag with acknowledged ID will stop responding to the reader. And a tag will repeatedly select a random back-off time and send its ID until the ID is identified and acknowledged by the reader.

X.4 .1.2 Slotted ALOHA protocol

In slotted ALOHA protocol [13], the random back-off time must be a multiple of a pre-specified slot time. Note that a slot time is usually set to be a time period that is long enough for a tag to send out its ID and for the reader to recognize the ID and acknowledge the ID. The reader needs to synchronize the slot times for all the tags in the interrogation zone. If only one tag transmits its ID in a period of a slot time, it can be identified and acknowledged by the reader properly. Tags not identified by the reader will repeatedly select a time slot randomly for transmitting their IDs. It is shown in [28] that the performance of slotted ALOHA protocol is twice that of the ALOHA protocol since there is no partial collision of tag ID responses in slotted ALOHA protocol.

X.4 .1.3 Frame slotted ALOHA protocol

In frame slotted ALOHA protocol [14], the whole interrogation procedure is divided into a set of frames, each having several time slots. On receiving reader's REQUEST command, each tag can respond just in one randomly chosen slot during a frame period. If there is only one tag response in a slot, the reader can identify the tag successfully. Tags not identified successfully will re-select a time slot in the next frame for retransmitting their IDs. At the time when no collision occurs, all tags are identified successfully. The frame rounds continue until that time.

In Fig. 3, we show an example of frame slotted ALOHA protocol in which each frame has 4 time slots. Suppose that there are 6 tags with unique 5-bit IDs in the interrogation zone of a reader. The execution procedure of the protocol is described as follows.

1. The reader sends REQUEST command first to synchronize the beginning of a frame.
2. Each tag randomly chooses one of the four available time slots in frame 0 to respond its tag ID after receiving REQUEST command. In our example, in frame 0, only tag ID (01110) in time slot 1 can be identified successfully. Collisions occur in time slots 2 and 4, and no tag responds in time slot 3.
3. The identified tag can be selected by SELECT command for reading and/or writing data. It will stop responding to REQUEST commands in later frames.
4. The reader sends REQUEST commands repeatedly until all tags are identified successfully, as shown in frames 1 and 2 of Fig. 3.

	frame0					frame1					frame2				
		timeslot1	timeslot2	timeslot3	timeslot4		timeslot1	timeslot2	timeslot3	timeslot4		timeslot1	timeslot2	timeslot3	timeslot4
reader	request					request					request				
tag1			10010						10010						
tag2		01110													
tag3					00101		00101							00101	
tag4			11011				11011						11011		
tag5			10110					10110							
tag6					01001					01001					
state		success	collision	idle	collision		collision	success	success	success		idle	success	success	idle

Figure 3. An example of frame slotted ALOHA protocol

One drawback of frame slotted ALOHA protocol is that its performance will degrade when the number of slots in the frame does not match properly the number of tags in the interrogation zone. Dynamic frame slotted ALOHA protocols [25] [26] [27] [29] try to eliminate the drawback by dynamically adjusting the frame size according to the estimated number of tags. Their performances are better than that of frame slotted ALOHA protocol.

X.4 .1.4 ISO/IEC 18000-6A protocol

ISO/IEC 18000-6 [22] is a standard that defines the air-interface communication at 860-960 MHz for the RFID system. There are three different types (A, B and C) of communication protocols defined in this standard. Among them, types A and C are ALOHA-based protocols. Since the type C protocol is a derivation of the type A protocol, below we only introduce the type A protocol.

In ISO/IEC 18000-6A protocol, a reader initiates a round of the identification procedure by sending out Init_round command. In this command, the number of slots in a round, namely the *round size*, is given. It is noted that the reader can dynamically determine a proper round size for the next round according to the number of collisions in the current round. After receiving the command, a tag randomly selects a time slot to respond its ID to the reader. The tag keeps a *slot counter* to track the current time slot. When the selected time slot arrives, the tag waits a random delay time in the range of 0 to 7 periods and responds a randomly chosen four-bit tag signature. If there is only one responding tag whose signature is received by the reader properly, the reader will send Next_slot command containing the received signature to the tag as an acknowledgment; otherwise, Close_slot command is sent. The tag has the following behaviors:

- The tag increases slot counter by one if it does not respond in the current slot and the received command is Close_slot or Next_slot.
- The tag increases slot counter by one if it responds in the current slot and the received command is Close_slot .
- The tag changes to *Quiet state* if it responds in the current slot and the received command is Next_slot with the same tag signature as its.

During a round, the reader can suspend the round by sending Standby_round command to tags. The suspension of the round allows the reader to conduct a dialogue with a selected tag for data reading/writing. When the slot count equals the round size specified in Init_round command, the round is finished and all tags not in Quiet state (i.e. tags not yet identified) will randomly select a new slot and a new random signature to enter a new round.

X.4 .2 Tree-based protocols

The basic idea of the tree-based tag anti-collision protocol is to repeatedly split the tags encountering collisions into subgroups until there is only one tag in a subgroup to be identified successfully. The protocols can be applied to tags with or without writable memory. Tags with memory have higher cost. However, protocols for such a kind of tags have better performance. In general, the tree-based protocol has longer identification time latency than that of the ALOHA-based protocol, but it does not have the tag starvation problem. A further drawback of the tree-based protocol is that its performance is affected by the length or the distribution of tag IDs. Below, we introduce some tree-based protocols: query tree [16], bit-by-bit binary tree [15], EPCglobal Class 0 [19], TSA [17], BSQTA [18] and BSCTTA [18] protocols.

X.4 .2.1 Query tree protocol

In the query tree protocol (QT) [16], a reader first broadcasts a request bit string S to tags. A tag with an ID prefix matching S will respond its whole ID to the reader. If only one tag responds at an instance, the tag is identified successfully. But if multiple tags respond simultaneously, the responses collide. In such a case, the reader broadcasts again with a longer bit string that has one more bit, 0 or 1, appended to S, i.e. S0 or S1. Obviously, the tags with prefix S are split into two subgroups S0 and S1. The splitting procedure will be performed repeatedly until every tag in the interrogation zone is identified successfully. The query tree protocol is a *memory-less* protocol because it does not require tags to be equipped with additional writable on-chip memory. We can observe that QT protocol's identification delay is affected by the distribution and the length of tag IDs. Specifically, if the tags have continuous tag IDs, the request bit string will grow longer and longer for identifying them. The delay time of the identification procedure will then increase significantly.

Below, we show an example of QT protocol. We assume that there are 6 tags with unique IDs 0010, 0011, 1001, 1100, 1101, and 1110. The tag interrogation process of QT protocol is described step by step as follows.

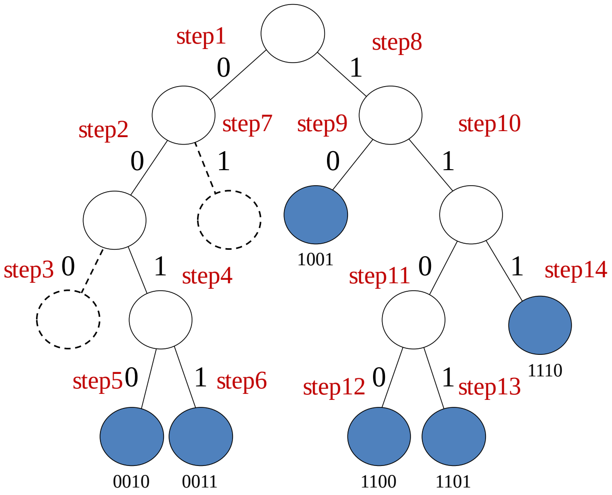
1. The reader sends out a request bit string S="0" first and pushes another request bit string

"1" into the stack. The tags with IDs 0010 and 0011 have the first bit of tag ID matching the request bit string S. They respond their tag IDs to the reader simultaneously and collision occurs.

2. The reader then sends out a longer request bit string S="00" and pushes "01" into the stack. The tags with IDs 0010 and 0011 respond the request simultaneously and collision again occurs.
3. The reader sends out a still longer request bit string S="000" and pushes "001" into the stack. None of the tags has an ID prefix matching S, so there is no response.
4. For the case of no response, the reader pops "001" from the stack and sends it out as a request bit string. The tags with IDs 0010 and 0011 respond the request simultaneously and collision again occurs.
5. The reader sends out a request bit string S="0010" and pushes "0011" into the stack. Only the tag with ID 0010 responds the request and is identified successfully.
6. For the case of successful identification, the reader pops "0011" from the stack and sends it out as a request bit string. Only the tag with ID 0011 responds the request and is identified successfully.

The identification procedure is executed repeatedly until the stack is empty. And then all tags can be identified successfully. The steps of the whole procedure and the associated tree diagram are shown in Table 1.

Table 1. The steps of the identification procedure of query tree protocol

Step	Request Bit String S	Response	Tree Diagram
1	0	Collision	
2	00	Collision	
3	000	Null	
4	001	Collision	
5	0010	0010	
6	0011	0011	
7	01	Null	
8	1	Collision	
9	10	1001	
10	11	Collision	
11	110	Collision	
12	1100	1100	
13	1101	1101	
14	111	1110	

X.4 .2.2 Bit-by-bit binary tree protocol

With the assistance of writable on-tag memory, bit-by-bit binary tree protocol [15] can reduce the tag collision efficiently. In this protocol, a reader broadcasts a request command first and each tag will respond to the request with the first bit of its tag ID. If collisions occur, the reader will acknowledge the tags with 0 (or 1). Only the tag with the first bit being 0 (or 1) will respond with the next bit to the reader. The above procedure repeats bit by bit until there is only one responding tag. The reader can then ask the tag to send out the remaining bits of its ID for the purpose of identification. With the on-tag memory, tags can keep track of the on-going status of the identification procedure and response a certain bit properly. Unlike QT protocol, bit-by-bit binary tree protocol does not require a reader to send long ID prefixes; the reader only sends out one bit at a time. Consequently, the delay time of the identification procedure is reduced.

X.4 .2.3 EPCglobal Class 0

In the EPCglobal Class 0 protocol [19], the tag will respond to reader's request with its first bit of the tag ID. Each tag responds with a single bit through one of two sub-carrier frequencies, one for binary 0 and the other for binary 1, so that the reader can recognize 0 and 1 at the same time. If the reader receives 0 and 1 simultaneously, it will acknowledge 0 to the tags; otherwise, the reader will instead acknowledge the receiving bit value. Only tags with the first bit matching the acknowledgement bit can respond the next bit to the reader, while the other tags will enter a mute state and keep silent temporarily until the reader requests the tags to start over to respond for a new round of tag interrogation. The above procedure repeats bit by bit until one tag can respond with full bits of its ID to be identified successfully. The tag can then enter a dormant state to sleep until the reader request all tags to start the next interrogation procedure.

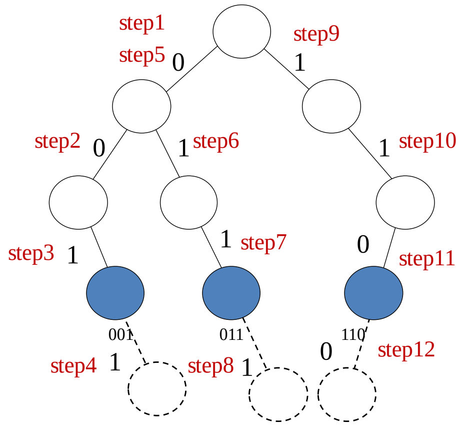
Below, we give an example to explain the details of EPCglobal Class 0 protocol. We assume there are three tags with unique IDs 001, 011 and 110, respectively. Some steps of the tag interrogation procedure are described as follows.

1. At the beginning, the reader sends a request command to ask tags to start a round of tag interrogation. On receiving the request, tags respond with the first bit of their tag IDs. Specifically, tag1 (with ID 001) responds with '0', tag2 (with ID 011) responds with '0', and tag3 (with ID 110) responds with '1'.
2. The reader receives both bits '0' and '1' from two separate subcarrier channels and acknowledges bit '0' to the tags.
3. Tag1 and tag2 will respond with the second bit of its tag ID (i.e., tag1 responds with '0' and tag2 responds with '1'). Tag3 enters the mute state and will keep silent temporarily until a next request command is received.
4. The reader still receives both '0' and '1'. It acknowledges with '0' to the tags.
5. Tag1 responds with the third bit '1' of its ID, while tag2 enters the mute state.
6. Since there is only tag1 responding and the number of responding bits is equal to the ID length, the reader acknowledges with '0' to the tags.

7. Tag1 responds with the third bit of its ID again and it is then identified successfully. It keeps in the dormant state until a next interrogation procedure starts.
8. The reader requests tags to start a round of tag interrogation. All tags in the mute state start responding to the reader.

The steps of the interrogation procedure continue until all tags in the interrogation zone are identified successfully. The complete steps and the associated spitting tree diagram are shown in Table 2.

Table 2. The identification procedure of EPCglobal Class 0 protocol

Step	Ack bit	Response Bit	Status	Tree Diagram
1		Tag 001：0 Tag 011：0 Tag110：1		 step1 step5 0 step9 1 step2 0 1 step6 1 step10 step3 1 1 step7 0 step11 001 011 110 step4 1 step8 1 0 step12
2	0	Tag 001：0 Tag 011：1 Tag110：Mute		
3	0	Tag 001：1 Tag 011：Mute Tag110：Mute		
4	1	Tag 001：1 Tag 011：Mute Tag110：Mute	identified	
5		Tag 001：Dormant Tag 011：0 Tag110：1		
6	0	Tag 001：Dormant Tag 011：1 Tag110：Mute		
7	1	Tag 001：Dormant Tag 011：1 Tag110：Mute		

8	1	Tag 001：Dormant Tag 011：1 Tag110：Mute	identified	
9		Tag 001：Dormant Tag 011：Dormant Tag110：1	success	
10	1	Tag 001：Dormant Tag 011：Dormant Tag110：1	success	
11	1	Tag 001：Dormant Tag 011：Dormant Tag110：0	identified	
12	0	Tag 001：Dormant Tag 011：Dormant Tag110：Dormant		

X.4 .2.4 TSA protocol

Tree slotted ALOHA (TSA) protocol [17] is a hybrid protocol which integrates the concepts of tree splitting and dynamic frame slotted ALOHA protocol. In TSA, all tags randomly select a time slot to transmit their tag IDs on receiving a reader's request. If there is only one responding tag in a time slot, the tag is identified properly. However, if there are multiple responding tags in a time slot, the reader remembers the slot number and demands only those tags to respond in the next frame. It is noted that the number of time slots in a frame is calculated by using a particular estimation function defined in [14]. The action performed is similar to splitting tags responding in the same slot into groups. This is why the protocol is called tree slotted ALOHA.

In TSA protocol, the reader includes in every request the number of slots in a frame, the slot number for splitting and the level of the tag splitting tree. By memorizing the slot number selected and keeping a level variable of the tag splitting tree, tags can keep track of the status of the identification procedure. Therefore, the identification procedure can be performed properly and all tags can then be identified successfully.

X.4 .2.5 BSQTA and BSCTTA protocols

BSQTA and BSCTTA protocols are proposed by Choi et al. in [18] to improve the query tree protocol. In the identification procedure of the query tree protocol, when the reader sends the request bit string S of length *k* to the tags, the tag that has ID prefix matching S will respond its partial tag ID of bits *k*+1, …, *n* to the reader, where *n* is the length of the ID. If collision happens, the reader needs to send the request bit string S0 and S1 to tags latter. Choi et al [20] observes that the request bit string S0 and S1 are the same in the first *k* bits and are different only in the last bit. On the basis of the observation, two methods, bi-slotted query tree algorithm (BSQTA) and bi-slotted collision tracking tree algorithm (BSCTTA), are proposed to reduce the identification time with the help of two response time slots. Below we introduce the procedure of the two methods step by step.

1. A reader sends the request bit string S of length *h*-1 to tags.
2. The tag in the interrogation zone of the reader will respond with its tag ID to the reader in one of two time slots if S matches with the first *h*-1 bits of the tag ID. If the h^{th} bit of the ID is ‘0’, the tag responds in the first response time slot; otherwise, it responds in the second time slot.
   •For BSQTA, the tag responds with its ID from the $(h+1)^{th}$ bit to the last bit.
   •For BSCTTA, the tag responds with its ID from the $(h+1)^{th}$ bit to the last bit until it receives an ACK command, which is sent by the reader to indicate the collision occurrence.
3. If there is no collision in a time slot, the tag can then be identified successfully.
4. If collisions occur in a response time slot (numbered with 0 or 1), then the reader should send a new request bit string to tags.
   •For BSQTA, the new request bit string will be S appended by the time slot number (0 or 1).
   •For BSCTTA, the new request bit string will be S appended by the bits received before collisions occur.

The above procedure is repeated until all tags are identified successfully. As shown in [24], the performance of query tree protocol can be improved significantly by BSQTA and BSCTTA.

X.4 .2.6 AQS protocol

AQS (Adaptive Query Splitting) protocol is an adaptive tag anti-collision protocol proposed by Myung et al. [30] to improve query tree protocol. The basic concept of this protocol is to reduce the collisions by referring the tag ID information obtained from the last identification round under the assumption that the tag population does not change greatly in consecutive rounds. The identification procedure of AQS protocol is the same as that of query tree protocol except that the request bit strings in the ready-to-send string queue is copied from the last identification round. The queue includes not only the request bit strings of steps of successful tag identification but also those of steps without any tag response. If the population of tags in the interrogation zone remains the same, all tags can be identified successfully without modifying any request bit string in the queue. But if there are tags joining or leaving after the last identification round, the following actions must be done.

- Tags joining:

  If tag collisions occur for the request bit string S provided by the last identification round, there must be new tags moving into the interrogation zone of the reader after the last identification round. For such a case, the tree splitting procedure is performed and longer request bit strings are added into the queue.

- Tags leaving:

  If some tag leaves, there will be no response for some request bit string S provided by the last identification round. In order to improve the identification performance, the reader should merge the request bit string S with the one in the queue that has the same bit string as S except for the last bit.

X.4 .3 Counter-based protocols

Counter-based protocols [11] [22] [23] [30], like tree-based protocols, do not have the tag starvation problem. The basic idea of the two classes of protocols is to repeatedly split the tags encountering collisions into subgroups until there is only one tag in a subgroup to be identified successfully. The major difference between these two classes of protocols is that the tree-based protocol relies on static tag IDs for the splitting, but the counter-based protocol relies on dynamically changing counters for the splitting. The tag splitting of the former and the latter is deterministic and probabilistic, respectively. Since the counter-based protocol does not rely on tag IDs for the splitting, it has the stable property that its performance is not affected by the ID distribution or the ID length. In this section, we introduce two counter-based tag anti-collision protocols: ISO/IEC 18000-6B and ABS protocols.

X.4 .3.1 ISO/IEC 18000-6B protocol

ISO/IEC 18000-6B [22] is a standard adopting the counter-based tag anti-collision protocol. In ISO/IEC 18000-6B, each tag uses a dynamically changing counter and a random bit generator for tag identification. All tags' counters are initially set to 0 and every tag with counter value 0 can transmit its tag ID to respond to the request of a reader. When a collision occurs, the reader notifies all tags of the collision. The tags with counters of values greater than 0 then increase their counters by 1, while the tags with counter value 0 randomly generate a random bit, 0 or 1, and add it to their counters. In this way, the tags with counters value 0 are split into two subgroups, one for tags with counter value 0 and the other for tags with counter value 1. The splitting procedure will be repeated until only one or none tag is of counter value 0. In the case of only one tag having counter value 0, this tag can be identified successfully and should keep silent until the end of the tag interrogation procedure. Either in the case of only one tag or in the case of no tag having counter value 0, the reader sends a command to inform all tags to decrease their counters by 1. The procedure will continue until all tags are identified successfully.

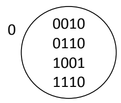
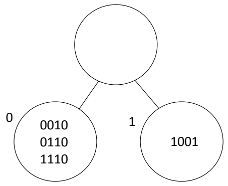
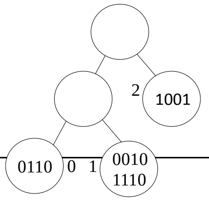
Below, we show an example to illustrate the procedure of ISO/IEC 18000-6B protocol. We assume there are four tags with unique IDs 0010, 0110, 1001 and 1110. The steps of the tag interrogation procedure are as follows.

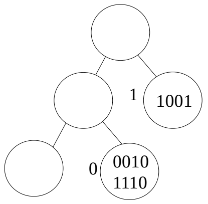
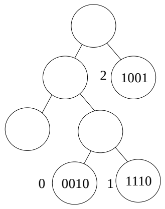
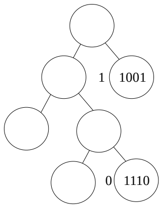
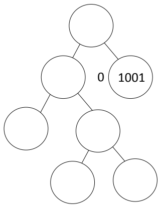
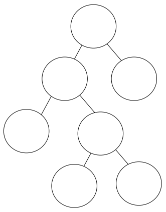
1. At the beginning, the reader requests tags to start a round of tag interrogation. On receiving the request, tags reset their counters to 0.
2. Tag1 (with ID 0010), tag2 (with ID 0110), tag3 (with ID 1001) and tag4 (with ID 1110) respond with their IDs to the reader simultaneously and collisions happen.

3. The reader sends a command to make all tags randomly add 0 or 1 to their counters.
4. Tags 1, 2 and 4 are with counter value 0. They respond with their IDs simultaneously and collisions occur again.
5. The reader sends a command to make tags 1, 2 and 4 randomly add 0 or 1 to their counters, while tag3 increases 1 to its counter.
6. Tag2 with counter value 0 responds with its ID to the reader and is identified successfully.
7. The reader acknowledges the ID with a command. All unidentified tags 1, 3 and 4 decrease their counters by 1.

The identification procedure is repeated until all tags are identified successfully. The whole steps of the interrogation procedure and the associated tree diagram are described in Table 3.

Table 3. The identification procedure of ISO/IEC 18000-6B protocol

Steps	Reader command	Tag ID	Counter value	Random bit	New Counter Value	Response	Tree Diagram
1	REQUEST	1	--		0	0010	
		2	--		0	0110	
		3	--		0	1001	
		4	--		0	1110	
2	Collision	1	0	0	0	0010	
		2	0	0	0	0110	
		3	0	1	1		
		4	0	0	0	1110	
3	Collision	1	0	1	1		
		2	0	0	0	0110	
		3	1		2		

		4	0	1	1		
4	Success	1	1		0	0010	
		2	0		--		
		3	2		1		
		4	1		0	1110	
5	Collision	1	0	0	0	0010	
		2	--		--		
		3	1		2		
		4	0	1	1		
6	Success	1	0		--		
		2	--		--		
		3	2		1		
		4	1		0	1110	
7	Success	1	--		--		
		2	--		--		
		3	1		0	1001	
		4	0		--		
8	Success	1	--				
		2	--				
		3	--				
		4	--				

X.4 .3.2 ABS protocol

ABS (Adaptive Binary Splitting) protocol [30] is proposed to improve ISO/IEC 18000 6B tag anti-collision protocol. A tag in ABS protocol keeps two counters, Progressed Slot Counter (PSC) and Allocated Slot Counter (ASC). PSC represents the number of tags identified successfully. PSC is initialized to 0 at the beginning and is increased by 1 when a tag is successfully identified. By PSC and ASC, a tag can decide if it can transmit its ID to respond to a reader request. All tags with ASC equal to PSC can transmit their tag IDs. When there is no response, all tags with ASC larger than PSC decrease ASC by one. When collisions occur, the reader notifies all tags of the collisions. For such a case, the tags with ASC larger than PSC then increase ASC by 1, while the tags with ASC equal to PSC randomly generate a random bit, 0 or 1, and add it to ASC. Note that tags with ASC less than PSC do not increase ASC; they do not even attempt to transmit their IDs until the tag interrogation procedure completes because they have already been identified.

After all tags are identified, tags have unique and successive ASC values. These values can be reserved for use in the next tag interrogation round to speed up the interrogation procedure. If there are tags joining or leaving after the last interrogation round, the following actions must be taken.

- Tags joining:

  When a new tag receives the reader's command to start a new interrogation round, it sets its PSC to 0 and sets its ASC to a random value R within a proper range passed by the reader. The new tag's response will collide with that of the old tag with ASC value R. The processes of ABS protocol mentioned above can deal with the collision properly by adjust all tags' counters.
- Tags leaving:

  If the reader detects that no tag respond to a request, there must be a leaving tag. All tags with ASC larger than PSC will decrease ASC by one to deal with the case.

As shown in [30], the performance of ISO/IEC 18000-6B tag anti-collision protocol is improved significantly by the ABS protocol. This justifies that the counter information obtained from the last interrogation round is very useful when the tag population does not change greatly in consecutive interrogation rounds.

X.5 Conclusion

X.5.1 The summary and new directions for reader anti-collision protocols

Because a passive tag is tiny and is energized by the reader, it only has limited computation power and communication capability. The common mechanisms, such as

RTS/CTS, used in wireless communications field to avoid collisions are not suitable for the RFID system. New mechanisms are thus needed to reduce collisions. In section X.3, we survey several reader anti-collision protocols for reducing reader collisions. They can be classified as TDMA (Time Division Multiple Access), FDMA (Frequency Division Multiple Access) and CSMA (Carrier Sense Multiple Access) protocols. With the popularity of the RFID system, there exist the following new research directions for reader anti-collision protocols.

1. Reader anti-collision protocols for mobile reader environments:

   In a static reader environment, we can allocate resources like frequency channels and time slots to reduce as many collisions as possible. But if the reader can move around, the signal interference will be dynamic and unpredictable. No prior fixed plan is suitable for the dynamically changing environment of mobile readers. New reader anti-collision protocols are thus needed for such environments.

2. Reader anti-collision protocols for dense reader environments:

   How to fairly allocate the resources (for example, frequency channels or time slots) among the readers is more complex for environments of dense readers. More efficient anti-collision protocols are thus needed for such environments. In some cases, it is required for a reader to cooperate with others to track tags. Reader cooperation can extend the area where tags can be tracked.

X.5.2 The summary and new directions for tag anti-collision protocols

In section X.4, we have categorized tag anti-collision protocols into three classes, namely ALOHA-, tree- and counter-based. We have then introduced some of them class by class. ALOHA-based protocols are simple and have fair performance. However, they have the tag starvation problem that a tag may never be identified since its responses always collide with others’. Tree-based protocols have longer identification latency than ALOHA-based protocols, but they do not have the tag starvation problem. Tree-based protocols also have the drawback that their performances are affected by the length or the distribution of tag IDs. Like tree-based protocols, counter-based protocols do not have the starvation problem. And they have the stable property that their performances are not affected by the tag ID distribution or ID length.

A good tag anti-collision protocol should have some characteristics. We list some characteristics that should be kept in mind when we develop new tag anti-collision protocols.

1. A reader need to recognize all the tags in its interrogation zone. If some tags cannot be identified properly for some reason (e.g., due to the tag starvation problem), this may cause problems in some applications. Therefore, a good tag anti-collision protocol should try to not miss any tag.
2. For many applications, tags are usually attached to mobile objects. Because a reader can only successfully identify the tags when the tags are within the interrogation zone, the reader needs to identify the tags as soon as possible so that mobile tags can be identified before they leave the interrogation zone.
3. Due to the limitation of communication and computation capabilities of tags, anti-collision mechanism should not be too complex. That is, we should keep the tag anti-collision protocol as simple as possible.
4. When a tag attached to an object is identified by readers of malicious people, the privacy of the person owning the object may be harmed. Therefore, there is a need to integrate anti-collision protocols with privacy-protection mechanisms so that tags can be identified efficiently without leaking privacy.

References

[1] Peter Schaar, “Working document on data protection issues related to RFID technology,” *Working Document Article 29 - 10107/05/EN*, European Union Data Protection Working Party, January 2005.

[2] H. Chae, D. Yeager, J. Smith, K. Fu, “Maximalist Cryptography and Computation on the WISP UHF RFID Tag,” in *Proc. of the Conference on RFID Security*, 2007.

[3] Shailesh M. Birari, Sridhar Iyer, “PULSE: A MAC Protocol for RFID Networks,” in *Proc. of the* EUC Workshops 2005: 1036-1046

[4] Kim, D. Y., B. J. Jang, H. G. Yoon, J. S. Park, and J. G. Yook, “Effects of reader interference on the RFID interrogation range,” in *Proc. the 37th European Microwave Conference (EuMC’07)*, pp. 728 – 731, Munich, Oct. 2007.

[5] D.-Y. Kim, H.-G. Yoon, B.-J. Jang, and J.-G. Yook, “Interference Analysis of UHF RFID

Systems, ” *Progress in Electromagnetics Research*, Vol. 4, pp. 115-126, 2008.

[6] Yoshinori Tanaka, Iwao Sasase, “Interference Avoidance Algorithms for Passive RFID Systems Using Contention-Based Transmit Abortion,” *IEICE Transactions 90-B(11)*, pp. 3170-3180, 2007.

[7] J. Waldrop, D. W. Engels, S. E. Sarma, “Colorwave : An Anti-collision Algorithm for the Reader Collision Problem,” in *Proc. of IEEE*, vol. 2, pp. 1206-1210, May 2003.

[8] Junius Ho, Daniel W. Engels, Sanjay E. Sarma, “HiQ: A Hierarchical Q-Learning Algorithm to Solve the Reader Collision Problem, ” in *Proc. of SAINT Workshops 2006*, pp. 88-91, 2006

[9] EPCglobal Class-1 Generation-2 UHF RFID Protocol. Version 1.0.9, Apr. 2004.

[10] EN 302 208-2 Protocol Version 1.1.1 Sep.2004

[11] Ming-Kuei Yeh and Jehn-Ruey Jiang,“Adaptive *k*-Way Splitting and Pre-Signaling for RFID Tag Anti-Collision, ”in Proc. of the 33rd Annual Conference of the IEEE Industrial Electronics Society (IECON'07), 2007.

[12] N. Abramson, “The ALOHA System-Another Alternative for Computer Communications,” in *Proc. of Fall Joint Computer Conference of AFIPS*, Vol. 37, pp. 281-285, 1970.

[13] Leian Liu, Shengli Lai, “ALOHA-Based Anti-Collision Algorithms Used in RFID System,” in *Proc. of Int’l Conf. on Wireless Communications, Networking and Mobile Computing 2006 (WiCOM 2006)*, pp.1 – 4, Sep. 22-24, 2006.

[14] H. Vogt, “Efficient Object Identification with Passive RFID Tags,” in *Proc. of Pervasive Computing 2002*, pp.98–113, 2002.

[15] H. Choi, J. R. Cha and J. H. Kim, “Fast wireless anti-collision algorithm in ubiquitous ID system”, in *Proc. of IEEE VTC '04*, 2004.

[16] Feng Zhou et al., “Evaluating and optimizing power consumption of anti-collision protocols for applications in RFID systems,” in *Proc. of the 2004 international symposium on Low power electronics and design*, 2004.

[17] M. A. Bonuccelli, F. Lonetti, F. Martelli. “Tree Slotted Aloha: a New Protocol for Tag Identification in RFID Networks,” in *Proc. of the 4th IEEE International Workshop on Mobile Distributed Computing (MDC'06)*, 2006.

[18] Ji Hwan Choi, Dongwook Lee, Hyuckjae Lee ,” Bi-slotted tree based anti-collision

protocols for fast tag identification in RFID systems,” *IEEE Communications Letters*, Vol. 10, Issue 12, pp. 861-863, 2006.

[19] Draft protocol specification for a 900 MHz Class 0 Radio Frequency Identification Tag

[20] D. H. Shih, P. L. Sun, and D. C. Yen, “Taxonomy and Survey of RFID Anti-Collision Protocols,” *Computer Communications*, Vol.29, No.11, pp.2150-2166, 2006.

[21] D. Krebs and M.J. Liard, “White Paper: Global Markets and Applications for Radio Frequency Identification,” Venture Development Corporation, 2001.

[22] ISO/IEC, “Information technology automatic identification and data capture techniques – radio frequency identification for item management air interface - part 6: parameters for air interface communications at 860-960 MHz,” *Final Draft International Standard ISO 18000-6*, Nov. 2003.

[23] Philips Semiconductors, UCODE, http://www.semiconductors.philips.com, 2005.

[24] “Dense RFID Reader Deployment in Europe using Synchronization,” *Final draft ETSI EN 302 208-1 V1.2.1*, Jan, 2008

[25] Jae-Ryong Cha , Jae-Hyun Kim, “Novel Anti-collision Algorithms for Fast Object Identification in RFID System,” in *Proc. of the 11th International Conference on Parallel and Distributed Systems -Workshops (ICPADS'05)*, pp.63-67, 2005.

[26] Girish Khandelwal et al., “ASAP: A MAC Protocol for Dense and Time Constrained RFID Systems," in *Proc. of IEEE International Conference on Communications (ICC’06)*, 2006.

[27] S. Lee, S.D. Joo, and C.W. Lee, “An enhanced dynamic framed slotted aloha algorithm for RFID tag identification,” in *Proc. of Mobiquitous 2005*, pp.166-172, 2005.

[28] L. G. Roberts, “Extensions of Packet Communication Technology to a Hand Held Personal Terminal,” in *Proc. of AFIPS Spring Joint Computer Conf.*, vol. 40, pp. 295-298, 1972.

[29] M. Kodialam and Thyaga Nandagopal, “Fast and Reliable Estimation Schemes in RFID Systems,” in *Proc. of ACM Mobicom 2006*, 2006.

[30] Jihoon Myung, Wonjun Lee, “Adaptive splitting protocols for RFID tag collision arbitration, ” in *Proc. of MobiHoc 2006*: pp. 202-213, 2006.