Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Communications 29 (2006) 2150–2166
computer communications
www.elsevier.com/locate/comcom

Short survey

Taxonomy and survey of RFID anti-collision protocols

Dong-Her Shih [a], Po-Ling Sun [a], David C. Yen [b,*], Shi-Ming Huang [c]

[a] *Department of Information Management, National Yunlin University of Science and Technology, 123, Section 3, University Road, Douliu, Yunlin, Taiwan, ROC*
[b] *Raymond E. Glos Professor, Department of DSC and MIS, Miami University Oxford, OH 45056, USA*
[c] *Department of Accounting and Information Technology, National Chung Cheng University, No. 168 University Road, Ming-Hsiung, Chia-Yi 62102, Taiwan, ROC*

Received 1 December 2004; received in revised form 14 December 2005; accepted 18 December 2005
Available online 18 January 2006

Abstract

Due to the limitless possibilities and low cost, Radio Frequency Identification (RFID) systems are used in a variety of applications to uniquely identify physical objects. The operation of RFID systems often involves a situation in which numerous tags are present in the interrogation zone of a single reader at the same time. The tags can collide and cancel each other out, leading to retransmission of tag IDs that results in wastage of bandwidth and an increase in the total delay. Besides, readers physically located near one another may interfere with one another's operation. Such reader collision must be minimized to ensure the current operation of the RFID system. The main focus of this paper is to survey the collision resolution protocols for multi-access communication and the algorithms that calculate how to minimize reader-to-reader interference. A comparative view of surveyed protocol was concluded for the collision problem in RFID systems.

Keywords: RFID tag; Mobile communication; Tag collision; Reader collision; Anti-collision

1. Introduction

Ubiquitous tagging is a paradigm where everything related has a unique tag associated with it. Picture the scenario that every object in the world can be uniquely identifiable with some form of electronic tags. This scenario would create tremendous benefits in terms of tracking and identifying an object and making ubiquitous identification possible. As ubiquitous identification systems have become commonplace in access control and security applications areas. Radio Frequency Identification (RFID) systems are increasingly being used as the automated identification system for these applications. The first traditional technology to be replaced by RFID is the bar code system – RFID can do everything bar codes can and much more [1]. Optical barcodes suffer from several drawbacks. First, human intervention is required to scan a barcode. Objects must be physically manipulated to align barcodes with scanners. Anyone who has shopped in market has likely witnessed a cashier struggling to scan an item. Second, the readability of barcodes could be affected by dirt, moisture, abrasion, or packaging contours. Third, the ability of storing data on barcode is very low. Fourth, retailers also often affix barcodes, which are unnecessary for them on top of packaging of goods. Finally, the barcodes is easily counterfeited. Among others, these issues limit the performance of optical barcode based on auto-ID systems. Today, over 5 billion bar codes are scanned daily worldwide [2,3] creating just one operation which RFID technology is predicted to take over. The actual idea of RFID has been around since 1960 [4,5].

RFID supporters claim to see an integration of RFID in all businesses. In the world of RFID, Wal-Mart [6] is currently the strongest advocate in promoting this new way to

* Corresponding author. Tel.: +1 513 529 4826; fax: +1 513 529 9689.
E-mail addresses: shihdh@yuntech.edu.tw (D.-H. Shih), yendc@muohio.edu (D.C. Yen), smhuang@mis.ccu.edu.tw (S.-M. Huang).

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.12.011

identify everything that can be marked with a RFID tag. Wal-Mart encourages its suppliers to adopt the technology by 2005, at the latest, for identification at case level [7]. Main competitors to Wal-Mart – e.g., Tesco and Metro group – follow closely behind, and cooperate to a certain extent in evaluating and implementing RFID at trial sites. The Metro Group operates "next-generation" supermarket in Rheinberg, Germany, with RFID implemented, where benefits of the technology have been seen [2]. There are still many other RFID applications. For instance, proximity cards, theft-detection tags, small dashboard devices for automating toll payments [8], cash like Euro [3], and even cattle herding [9].

RFID systems are composed of three main components as shown in Fig. 1 [10]:

- One or more RFID tags, also known as transponders (transmitter/responder), are attached to the objects to count or identify. Tags could be either active or passive. Active tags are those that are partly or fully battery powered, have the capability to communicate with other tags, and can initiate a dialogue of their own with the tag reader. Passive tags, on the other hand, do not need any internal power source but are powered up by the tag reader. Tags consist mainly of a microchip and coiled antenna, with the main purpose of storing data.
- A reader or transceiver (transmitter/receiver) made up of an RF module and control unit. Its main functions are to activate the tags, structure the communication sequence with the tag, and transfer data between the application software and a tag.
- A Data Processing Subsystem, which can be an application or database, depending on the application.

The application software initiates all readers and tags activities. RFID provides a quick, flexible, and reliable way to electronically detect, track and control a variety of items [11]. RFID systems use radio transmissions to send energy to a RFID tag while the tag emits a unique identification code back to a data collection reader linked to an information management system. The data collected from the tag can then be sent either directly to a host computer, or stored in a portable reader and up-loaded later to a computer [12]. The reader's ability of processing a great quantity of tags simultaneously for data collecting is noteworthy. If multiple tags are to be identified simultaneously, messages from tags can collide and cancel each other out at the reader. This situation will lead to retransmission of tag IDs, which results in wastage of bandwidth and increases the total delay in identifying all the objects. Hence, anti-collision algorithms need to be devised between the tags and reader to minimize collisions. If RFID systems can offer features such as anti-collision and high-speed detection combined with 100% data accuracy, then it can enable efficient usage in practically every application. Therefore, the natural problem is: what protocol should the reader and the tags use so that the ID of each tag can be communicated to the reader as quickly and reliably as possible? Without any coordination between the reader and the tags, the responses from the tags to the reader can collide causing the IDs of the tags to become illegible to the reader. Therefore, the RFID collision problems could be summarized and classified into the tags identification problem [13–18] and readers collision problem [19–22].

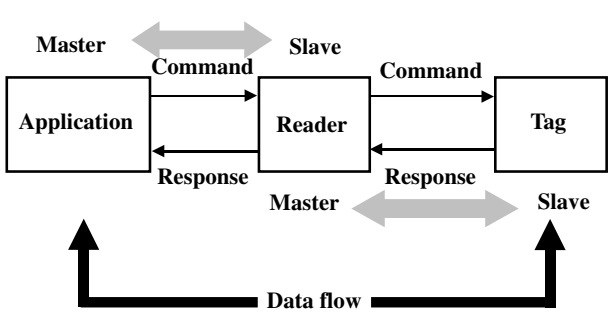

Fig. 1. Master-slave principle between application, reader, and tag.

The tags identification problem is associated with how to efficiently develop an anti-collision protocol in RFID tags. It can be defined as to identify multiple objects reliably without significant delay by utilizing minimal transmission power and computation. Collision-resolution protocols that address this problem cannot be directly applied to the tag identification problem due to various constraints [14]. In multi-access protocols, the main factors for performance evaluation include throughputs, packet delay, and stability. However, in RFID arbitration, total time to identify all objects and the power consumed by tags are more relevant. Abraham [23] claimed minimal delay, power consumption, reliability & completeness, line-of-sight independence, robustness, and scalability are all the desirable characteristics of the collision resolution protocol for communication between the tag and the associated reader. Furthermore, interference may be either frequency interference or tag interference. Frequency interference occurs when physically close readers communicate at the same time with the same frequency. Tag interference, on the other hand, occurs when neighboring readers attempt to communicate with the same tag at the same time.

Readers with interrogation zones intersected can interfere with one another and it will often reach to the point where neither readers will be able to communicate with any tags located within their respective interrogation zones. Readers may also interfere with another's operation even if their interrogation zones do not overlap. Such interfere is due to the use of radio frequencies for communication, and is very similar to the interference experienced in cellular phone systems. Interference detected by one reader and caused by another reader is referred to as a reader collision, and the problem to minimize reader collisions is referred to as the reader collision problem. There are two primary types of controllable interference experienced in RFID systems – reader-to-reader interference and reader-to-tag interference [19].

Reader-to-reader interference occurs when a reader transmits a signal that interferes with the operation of another reader, thus preventing the second reader from communicating with tags in its interrogation zone. This type of interference occurs when the signal transmitted by a reader is of sufficient strength and received at a second reader that the signal masks or jams communication from tag to the second reader. Interrogation zones will not be needed to have an overlap for reader-to-reader interference to occur.

Reader-to-tag interference occurs when one tag is simultaneously located in the interrogation zones of two or more readers and more than one reader attempts to communicate with that tag at the same time. In this type of interference, each reader may believe it is the only reader communicating with the tag while the tag is in fact communicating with multiple readers at the same time. The simple nature of RFID communication can cause the tag to behave and communicate in such an undesirable way to interfere with the communicating readers' capabilities to communicate with that tag and other tags in their respective interrogation zones.

In this paper, the authors surveyed some novel collision resolution protocols that allow the reader to obtain the ID from each tag within its readable range, while the computational and memory requirement for each tag is minimal. Furthermore, the authors also surveyed how to minimize the reader collision problem.

2. Countermeasures of surveyed tags collision problem

In many existing applications, a single contact-less device is sufficient and even necessary. However, in a growing number of new applications, the simultaneous reading of several tags in the same RF field is absolutely critical: library books, airline baggage, garment, and retail applications are a few. The problem of multi-access has been around for a long time in radio technology. Examples include news satellites and mobile telephone networks. To read multiple transponder simultaneously, the contact-less device and reader must be designed to detect the condition that more than one device is active. The RFID interface also requires arbitration so that only one contact-less device transmits data at one time. For this reason, numerous procedures have been developed with the objective of separating the individual participant signals from one another. Basically, there are four different procedures [10]: space division multiple access (SDMA), frequency domain multiple access (FDMA), time domain multiple access (TDMA), and code division multiple access (CDMA) as shown in Fig. 2.

SDMA (Space Division Multiple Accesses). The term relates to techniques that reuse a certain resource, such as channel capacity in spatially separated areas. One option is to significantly reduce the range of a single reader, but to compensate by bringing together a large number of readers and antennas to form an array, thus providing coverage

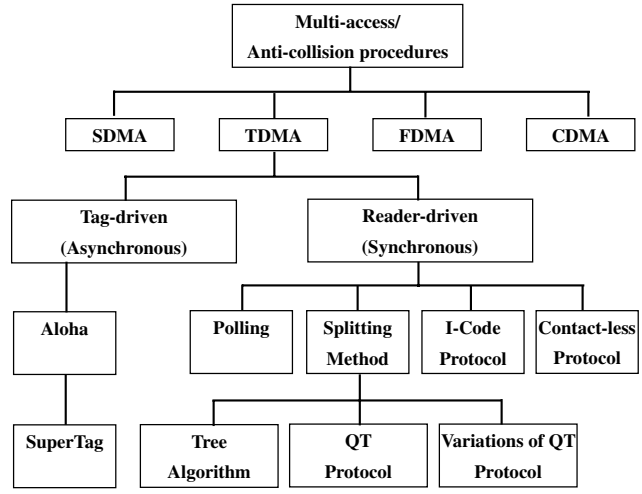

Fig. 2. Taxonomy of anti-collision protocols for RFID tags.

of an area. As a result, the channel capacity of adjoining readers is repeatedly made available. A further option is to use an electronically controlled directional antenna on the reader, the directional beam of which can be pointed directly at a tag (adaptive SDMA). Therefore, various tags can be differentiated by their angular position in the interrogation zone of the reader. Therefore, adaptive SDMA can only be used for RFID applications at frequencies above 850 MHz (typical 2.45 GHz) as a result of the size of the antennas. To address a tag, the space around the reader must be scanned using the directional antenna, until the 'search light' of the reader, which is shown in Fig. 3, detects a tag. The directional beam is pointed at the various transponders one after the other. A disadvantage of the SDMA technique is the relatively high implementation cost of the complicated antenna system. The use of this type of anti-collision procedure is therefore restricted to a few specialized applications. Table 1.

FDMA (Frequency Division Multiple Accesses). The term relates to techniques in which several transmission channels on various carrier frequencies are simultaneously available to the communication participants. In RFID systems, this can be achieved using tags with a freely adjust-

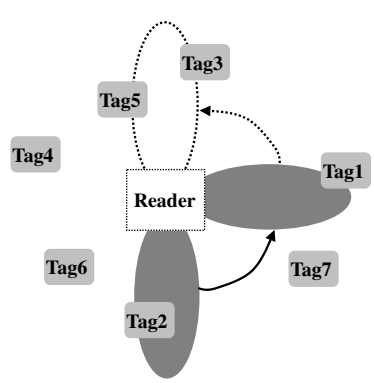

Fig. 3. Adaptive SDMA with an electronically controlled directional antenna.

Table 1
Coin flipping by tags

ID: {000, 001, 1}			
Node	First	Second	Third
T	0	0	1
R	0	0	–
RR	0	1	–

able harmonic transmission frequency. The power supply to the tag and the transmission of control signals (broadcast) takes place at the optimally suited reader frequency f_a. The tags respond on one of several available response frequencies $f_1 - f_N$ as shown in Fig. 4. Therefore, completely different frequency ranges can be used for the data transfer from and to the tags (e.g., reader → tag (downlink): 135 kHz, tag → reader (uplink): several channels in the range 433–435 MHz). One disadvantage of the FDMA procedure is the relatively high cost of the readers, since a dedicated receiver must be provided for every reception channel. This anti-collision procedure, too, remains limited to a few specialized applications.

CDMA (Code Division Multiple Accesses). There are actually a number of different subtypes to CDMA depending on how the spreading is done. But the common factor is that CDMA uses spread spectrum modulation techniques based on pseudo random codes, to spread the data over the entire spectrum. While CDMA would be ideal in many ways, it adds quite a lot of complexity and would be too computationally intense for RFID tags.

TDMA (Time Division Multiple Accesses). The term relates to techniques in which the entire available channel capacity is divided between the participants chronologically. TDMA procedures are particularly widespread in the field of digital mobile radio systems. In RFID systems, TDMA procedures are the largest group of anti-collision procedures. Tag-driven and reader-driven procedures has been differentiated and shown in Fig. 2.

Tag-driven procedures function asynchronously, since the reader does not control the data transfer. For example, in the ALOHA procedure, a tag begins transmitting as soon as it is ready and has data to send. Upon entering a powering field, the tags automatically send their IDs. This is referred to as a "Tag-Talks-First (TTF)" behavior, the opposite of which would be a "Reader-Talk-First (RTF)" behavior, as is seen in several implementations. In the SuperTags approach [11], which operates on the Aloha anti-collision principle, tags continuously retransmit their identifier at random intervals until the reader acknowledges their transmission. After reception of tag data, tags can be muted or their repetition rate can be slowed. This method of muting a tag allows the proper counting of many tags in the same field. Another SuperTag variation involves the muting of all tags except the one being read. This ensures that no collision occurs. After a certain period, the muted tags are activated, one by one, until they are all counted. In other methods, the reader, by sending a gap or power burst, prompts the tags to respond after a randomly generated delay. Although high performance can be achieved via Aloha-based methods, they may not function as well as binary tree searches in high tag density environments.

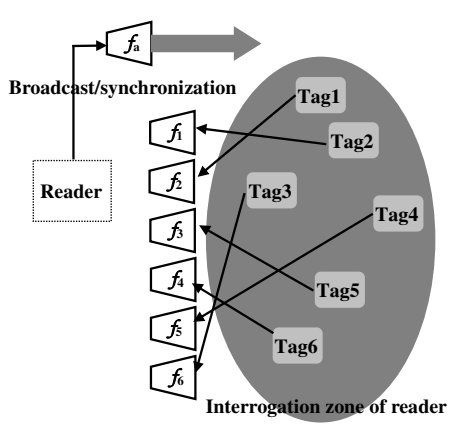

Fig. 4. Available frequency channels in FDMA.

Tag-driven procedures are naturally very slow and inflexible. Most applications therefore use procedures that are controlled by the reader as the master (reader-driven). These procedures can be considered as synchronous, since all tags are controlled and checked by the reader simultaneously. An individual tag is first selected from a large group of tags in the interrogation zone of the reader using a certain algorithm and then the communication takes place between the selected tag and the reader (e.g., authentication, reading, and writing of data). Only one communication relationship is initiated at any one time, but the tags can be operated in rapid succession; therefore, reader-driven procedures are also known as time duplex procedures.

Reader-driven procedures could be subdivided into polling, splitting method, I-code protocol, and contactless protocol, each of which are described in the following.

2.1. Polling

When a master node invites the slave nodes to transmit data in turn is usually called polling. Its major concerns are polling overhead, latency, and single point of failure in general data network. Nevertheless, the polling anti-collision protocol, in an RFID system, is an RFID tag used when communicating with the reader [10]. The polling procedure requires a list of all the tag serial numbers that can possibly occur in an application. The list can be obtained from the preset list or the dynamic census. The reader emits a radio signal, which is picked up by the antenna of the RFID tag and essentially communicates with one RFID tag at a time. The RFID tags do not transmit their entire serial number in one burst, they respond to signals from the reader by revealing one binary digit at a time. The reader interrogates the RFID tags asking "whose serial number starts with a 1 in the first position?" Those RFID tags which do not meet

this test then remain silent, and ignore the rest of the interrogation sequence, whilst the rest of them transmit a "yes that is correct" answer back to the reader and then await a similar question about the next digit in their binary serial number. The process is repeated until the reader has identified each of the RFID tags in range. The reader interrogates each serial number one after the other. This procedure can be very slow and depends upon the number of possible tags, and therefore is only suitable for applications with few known tags in the field.

The above solution can be used to resolve the tag IDs utilized for the object identification problem. Its advantage is that identification of all the tags is one hundred percent guaranteed unlike a probabilistic approach.

However, in selecting a suitable communication process, its drawbacks include:

1. The reader must first detect the presence of tags by some other means before it can initiate communications.
2. The procedure is slow and inflexible.

2.2. Splitting methods

In conventional multi-access systems, a branch of algorithms introduced by Capetanakis [24] called the Splitting or Tree-Search algorithms can be used effectively for RFID arbitration. Nodes transmit packets in time slots, when queried by the receiver. If there is more than one node transmitting in a time slot then a collision occurs at the receiver and no useful information is obtained. In these types of algorithms, collision resolution split the set of colliding nodes into two subsets. Nodes in the first subset transmit in the first time slot. Nodes in the other subset wait until the collision between the first subset of nodes is completely resolved. If the first subset of nodes encounters another collision, then further splitting takes place. This is done recursively till all the collisions have been resolved. Once all the collisions in the first subset of nodes are resolved, then a similar procedure is followed for the second subset.

The nodes correspond to tags in the RFID arbitration and the receiver corresponds to the reader. Tags send their IDs in response to the query from the reader. If all collisions in one subset are resolved, it implies that the reader has successfully identified all tags in that subset. The nodes get divided into subsets based on different approaches. One common approach, which will be discussed as tree algorithm in Section 2.2.1, is to use a randomly generated number. This can be visualized as flipping an unbiased coin by each node involved in the collision and splitting them into two subsets based on the outcome. Another approach is to use the unique identifier of the tags, which will be discussed as QT protocol in Section 2.2.2 and in Section 2.2.3 as some variation of QT protocol.

2.2.1. Tree algorithm

Hush and Wood [14] show how the Tree algorithm can be applied to RFID systems to uniquely identify the set of tags that are within the range of the reader. The algorithm works by splitting the group of colliding tags into B disjoint subsets (where B is an integer greater than 1). The subsets get smaller and smaller until the number of tags within a subset reduces to 1, in which case the tag would be uniquely identified. Once a subset is completely resolved, waiting subsets are resolved in a 'first-in last-out' order.

Algorithms of this type can be viewed as a tree search. Each split moves the algorithm one level deeper in the tree. In Capetanakis' original splitting algorithm [24], a binary tree was used to describe the communication paradigm between the reader and the tags. An example is shown in Fig. 5 for the case where the number of tags is shown as $m = 3$. Nodes in the tree are labeled according to their activities: W, Wait; C, Collision; S, Single Reply; and Z, Zero Reply. The reader first communicates with all the tags within its range. The tags respond to the reader's query. All the tags within the range, represented as T in Fig. 5, are the ones that collided in the current slot. Each collided tag then generates a random number by flipping an unbiased B-sided coin. Assume $B = 2$, thus each collided tag would generate a number 0 or 1. Based on the random value generated, the subset is split into two groups L and R, where L is the set of tags, which generated the value 1, and R is the set of tags, which generated 0. In the next slot, those tags, which belong to the subset R, would transmit. If there were more than 1 tag in the subset, then another collision would occur. This set of tags would generate another random number and the subset is split again. This continues recursively till the subset is reduced to 1 tag, which on transmission would be successfully identified by the reader. Successful transmissions occur in order from right to left in this example. The sequences of coin flips made by the tags in this example are:

First Tag Identified: 0, 0, 0
Second Tag Identified: 001
Third Tag Identified: 1

The reader always sends a feedback informing the tags whether 0 packets, 1 packet or more than one packet is transmitted in the previous slot. This feedback is required, as each tag needs to keep track of its position in the tree

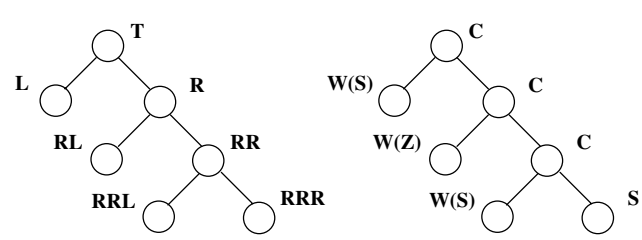

Fig. 5. Example of splitting algorithm.

and should know which subset it belongs to and when to transmit. Bertsekas and Gallager [25] described how this is implemented by the use of a counter. The algorithm can operate as a stack. On the occurrence of a collision, the subset is split and each resulting subset is then pushed on to the stack. The subset at the top of the stack is then removed and those tags belonging to the subset will transmit. Each tag can know when to transmit if it knows where in the stack its subset is currently positioned. Maintaining a counter at each tag does this. When the tag is involved in a collision, it sets the counter to 0 or 1 depending on which subset it is placed in after splitting. For example, in the scenario mentioned earlier, those tags which generated 0 as the random number would set their counters to 0, while the tags in the other subset would set their counter to 1. Depending on the reply from the reader, the counter at each of the tags is incremented by 1 for each collision and decremented by 1 for each success or idle state. The tag would transmit only if the counter value is 0.

2.2.2. QT protocol

Ching et al. [13] described each tag $i \in \{1, \ldots, n\}$ as having a unique ID string in $\{0,1\}^k$ which k is the length of the ID string. A binary string of *k-bits* uniquely identifies each tag. The value of the parameter k will depend on the number of objects that need to be identified uniquely. Each tag is memory-less, i.e., the current response of each tag only depends on the current query of the reader but not on the past history of the reader's queries. Moreover, the only computation required for each tag is to match its ID against the binary string in the query.

The QT algorithm consists of rounds of queries and responses. In each round, the reader asks the tags whether any of their IDs contains a certain prefix. If more than one tag answers, then the reader knows that there are at least two tags having that prefix. The reader then appends symbol 0 or 1 to the prefix, and continues to query for longer prefixes. When a prefix matches a tag uniquely, that tag can be identified. Therefore, by extending the prefixes until only one tag's ID matches, the algorithm can discover all the tags. The QT protocol [13,23] is shown in Fig. 6.

After each cycle, the reader sends a message informing the tags, the ID of the tag that it identified in the previous cycle. This is necessary because the tag that got successfully identified should not transmit in subsequent cycles (this is done by setting "quiet" bit to 1).

Table 2
Communication between the reader and the tags with the QT algorithm

ID: {000, 001, 101, 110}									
Step	1	2	3	4	5	6	7	8	9
Query	Empty String	0	1	00	01	10	11	000	001
Response	C	C	C	C	Z	S (101)	S (110)	S (000)	S (001)

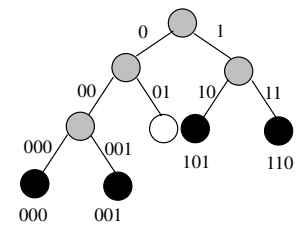

Fig. 7. Example of Query Tree Algorithm.

For example, assume that in a room there are 4 objects that have unique identifiers 000, 001, 101, 110. The task of the reader is to identify these tags uniquely. Table 2 below describes all the steps that the algorithm goes through. The item – Response in the table is labeled according to their activities: C, Collision; S, Single Reply; and Z, Zero Reply. To identify 4 tags in this case the reader has to send the prefixes 9 times.

A query tree, as shown in Fig. 7, is a full binary tree capturing the complete reader-tags dialogue of the QT algorithm. For a given execution of the protocol, there is a one-to-one correspondence between every node in the query tree and every query sent by the reader. Therefore, the node number in the query tree is equal to the number of queries sent by the reader. The edges connecting the nodes in the query tree contains the prefix string sent in that message. Whenever there are no tags, the corresponding sub tree rooted at that node is pruned. The nodes colored black are the tags with their IDs that have been identified by the reader, the gray nodes indicate colliding tags and the white nodes indicates that no tag responded.

A The reader sends out a string prefix p. Initially it will start with a 0 or 1.

B Three possible cases can arise based on the tags' reponse :

1. More than one tag has p as a prefix : All the tags that have p as a prefix will send a reply. Early in the identification process, it is more like for more than one tag to have same prefixes. Replies sent by the tag reach the reader simultaneously leading to a collision.
2. Exactly one tag has p as a prefix : The reader receives a specific reply (complete ID) from a tag and thus the tag gets identified uniquely.
3. No tag has p as a prefix : If none of the tags has p as a prefix the reader does not get any reply from the tags.

C Prepare another string (p) by appending 0 or 1 (as appropriate) which has to be sent to all the tags subsequently.

D Repeat steps A to C until all tags are identified. The steps A to C constitute a cycle.

Fig. 6. QT protocol pseudo-code.

2.2.3. Variations of QT protocol

Ching et al. [13] also discussed and suggested some variations of QT protocol in reducing the running time or the number of bits transmitted of the QT algorithm.

In reducing the running time, Ching et al. [13] suggested shortcutting, aggressive advancement, and categorization techniques.

Shortcutting. It is a smart manifestation of skipping internal nodes where collision is bound to happen. Consider any internal node of the query tree. This corresponds to a collision for certain prefix q during an execution of the QT algorithm. The algorithm will continue to search for the tags by appending 1 and 0 to the prefix q. Without loss of generality, assume that the algorithm chooses to search for 0 first. If it turns out that there are no tags with prefix $q0$, and then we know that there are at least two tags with prefix $q1$. Therefore, the reader should skip the prefix $q1$ as well. It has been shown to give an improved expected running time bound of $2.665n - 1$.

Aggressive advancement. Assume the reader knows that there are at least n unrecognized tags with prefix q. For example, this could be a prior knowledge: maximum number of items in a checkout counters or the reader can detect the strength of the response from the tags to estimate the number of tags. When n is large, it is very likely that the responses for $q1$ and $q0$ will collide. The probability that either one of the queries does not result in a collision is $2 \times (\frac{1}{2^n} + n \times \frac{1}{2^n}) = \frac{n+1}{2^n-1}$. Suppose we extend the prefix string by two bits. That is, the reader will query $q00$, $q01$, $q10$, and $q11$. Two queries $q1$ and $q0$ with probability $1 - \frac{n+1}{2^n-1}$ is saved in this case. Note that we send more queries (compared with the original QT algorithm) only in the case where both $q0$ and $q1$ have exactly one tag with a matching prefix. This cannot happen when $n \geqslant 3$.

Categorization. If the reader has some information about the types of the tags, then it is possible to speed up the protocol. For example, suppose a set S of IDs is given. Assume the reader knows that set S can be partitioned into $S_1, \ldots, S_m$ such that all IDs in S_i have prefix q_i. Now the reader can just identify each set S_i independently. In particular, if we can partition the tags into m groups, then the upper bound on the expected running time is improved to $2.887n - m$.

In reducing the number of bits transmitted, Ching et al. [13] also suggested Query-Tree short-long protocol and Query-Tree incremental-matching protocol.

QT-sl (Query-Tree short-long) protocol. Note that in QT algorithm; a lot of the *k-bits* responses from the tags would end up in collisions. To minimize these wastes, we can have two types of queries from the reader. The short queries will only induce 1-bit responses from the tags, while the long queries will induce the full tag IDs. The reader will send a long query only when it knows that only one tag matches the prefix. The QT-*sl* protocol [23] is shown in Fig. 8. Let there be n tags to be identified. The expected reader communication complexity of QT-*sl* protocol is at most $3.89kn + 3.89n$. The expected tag communication complexity of QT-*sl* protocol is at most $2.21\log_2^n + k + 4.19$.

QT-im (Query-Tree incremental-matching) protocol. It is very similar to QT-sl protocol and it can reduce the expected reader communication's complexity. However, this optimization requires a tag to remember the bit position of the prefix it has matched so far. Therefore, the modified protocol is no longer to be memory-less. Each tag has a bit marker $b \in \{1, \ldots, k\}$. When the tag is active, upon receiving the query, the tag matches the query string starting from bit b. If the matching is successful, then bit marker b is incremented by 1. Any active tag that mismatches would go into the transient state, which is equivalent to become inactive in the next query unless that query contains the reactivate command. Moreover, it is no longer necessary to supply a prefix with the long query. The tag communication complexity of QT-*im* protocol is the same as that of QT-*sl* protocol. However, the number of bits sent by the reader is reduced. The expected reader communication complexity of QT-*im* protocol is at most $4.42nlong_2^n + 12.18n$.

2.3. I-code protocol

I-Code protocol is a stochastic passive tag identification protocol based on the framed-slotted Aloha concept. Each tag transmits its information in a slot that it chooses randomly based on the seed sent by the reader. In each slot, as shown in Fig. 9, can happen: empty slots (no tag), filled slots (a single tag), or garbled slots (multiple tags transmit). The reader can detect the identity of the tag when a single tag transmits in a time slot. When multiple tags use the same slot a collision occurs and data gets lost. The reader can vary the frame size, the actual size of a slot is chosen according to the amount of data requested [15,17].

Harald [17] shows a tag reading cycle consists of two steps:

1. Reader $\rightarrow I$, rnd, N
2. $T \rightarrow_{S_T(N,rnd)}$ Reader: $data_{T,I}$ for all tags T

In the first step, the reader device broadcasts a request for data. I denotes what data are requested by specifying an interval of the available 64 bytes of tag memory; $rnd \in [0,31]$ is a random value whose use is explained below; $N \in \{1,4,8,16,32,64,128,256\}$ is the frame size and denotes the number of available slots for responses. In second step, tags that are in the proximity of the antenna response, where $\rightarrow_s$ denotes a tag sending in slot s and $0 \leqslant s < N$. A tag T uses a tag-specific function S_T to compute its response slot number by using the frame size and the random value as parameters and the random value are supposed to avoid the same collisions occurring repeatedly.

The reader starts with an estimate of the number of slots required to identify all the tags within its detection range. The tags select a slot at random from those available in a read cycle and transmit the information

The QT – sl Protocol

Let $A = \bigcup_{i=0}^{k} \{0,1\}^i$ be the set of binary strings with length at most k. The state of the reader is a pair (Q, M), where:

1. Queue Q is a sequence of strings in A;
2. Memory M is a set of strings in A.

A query from the reader is a pair (c, w), where $c \in \{short, long\}$ and $w \in A$.

A reply from a tag is a string 1 or a string in $\{0,1\}^k$.

Reader

For convenience, let us define the queue Q be $\langle \varepsilon \rangle$, where ε is the empty string, and memory M be empty initially.

1. Let $Q = \langle q_1, q_2, \cdots, q_i \rangle$.
2. Broadcast $query(short, q_1)$ to the tags.
3. Update Q to be $\langle q_2, \cdots, q_i \rangle$.
4. On receiving the responses from the tags:
   - If the reply is 1, then
     i. Broadcast $query(long, q_1)$ to the tags;
     ii. Insert the resulting reponse string w into memory M.
   - If a collision is detected at the communication channel, then set Q to be $\langle q_2, \cdots, q_1, q_1 0, q_1 1 \rangle$.
   - If there is no reply, do nothing.

Repeat the above procedure until Q is empty.

Tag

Let $w = w_1 w_2 \cdots w_k$ be the tag's ID. Let (c, q) be the query received from the reader. If $q = \varepsilon$ or $q = w_1 w_2 \cdots w_{|q|}$, then

- If command e is short, send string 1 to the reader;
- If command e is long, send string w to the reader.

Fig. 8. QT-*sl* protocol pseudo-code.

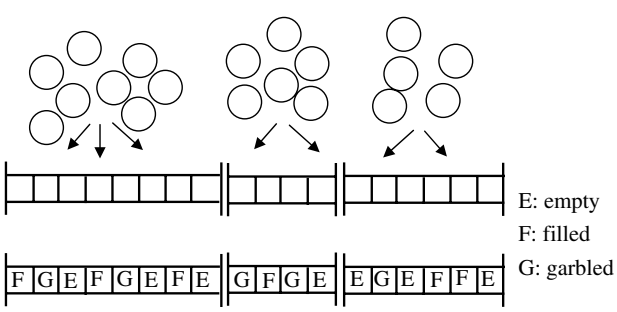

Fig. 9. Tags are randomly allocated to slots within a frame (above). This result in some slots remaining empty and others contain one or more tags (below).

requested by the reader. The reader detects the number of slots by a triple of numbers $c = \langle c_0, c_1, c_k \rangle$, where c_0 stands for the number of slots in the read cycle in which 0 tags have transmitted, c_1 denotes the number of slots in which a single tag transmitted and c_k stands for the number of slots in which multiple tags are transmitted.

There are two estimation functions that yield approximations for n. The first estimation function is obtained through the observation that a collision involves at least two different tags. Therefore a lower bound on the value of can be obtained by the simple estimation function ε_{lb}, which is defined as [17]:

$$\varepsilon_{lb}(N, c_0, c_1, c_k) = c_1 + 2c_k.$$

Alternative estimation function uses the distance between the read result and the expected value vector to determine the value of n for which the distance becomes minimal. This estimation function ε_{vd} is defined as

$$\varepsilon_{vd}(N, c_0, c_1, c_k) = \min_n \left| \begin{pmatrix} a_0^{N,n} \\ a_1^{N,n} \\ a_{\geqslant 2}^{N,n} \end{pmatrix} - \begin{pmatrix} c_0 \\ c_1 \\ c_k \end{pmatrix} \right|.$$

The reader re-estimates the number of slots for the next cycle based on its estimate of the number of tags computed in the current cycle. Let *n_new* be the new estimated value of the number of tags as computed from $\langle c_0, c_1, c_k \rangle$ of the current read cycle. A range for the estimated number of tags is defined as (*n_low*,*n_high*) {*n_low* corresponds to the lower limit of the range and *n_high* the upper limit}. Various N values corresponding to specific ranges have been found from experiments and tabulated in Table 3 [7]. If *n_new* falls in the range (*n_low*,*n_high*) {i.e., *n_low* - *n_new* $\leqslant$ *n_high*} then based on Table 3 the number of slots N (for the next cycle) is chosen corresponding to this particular range of (*n_low*,*n_high*). For example, if $n \in [17, 27]$, both 32 and 64 are appropriate choices for N since both settings yield similar times. The Table 3 lookup can be implemented in a way that is shown in Fig. 10 [17].

Once the reader fixes the number of slots for detecting the tags, it uses a stochastic function to calculate the total number of read cycle(s) that are necessary to detect all the

Table 3
Optimality intervals for frame sizes

N slots	1	4	8	16	32	64	128	256
n_low	–	–	–	1	10	17	51	112
n_high	–	–	–	9	27	56	129	∞

```
int adaptFrameSize(N, n_est){
    while (n_est < low(I(N))) {N = N/2}
    while (n_est > high(I(N))) {N = 2*N}
}
```

Fig. 10. Choosing a frame size.

tags with a certain level of accuracy. For a fixed frame size N, the time T_α required to achieve an assurance level α is given by [17]

$$T_\alpha = s_0 * t_N,$$

where s_0 is the minimum number of read cycles required to identify all n tags in the field with probability α. s_0 is therefore the minimum value of s for which the following condition holds:

$$Q^s q(0)[n] \geqslant \alpha,$$

where Q refers to the transitional probability matrix of the number of tags detected in successive read cycle, s stands for the number of read cycles, $q(0)$ refers to the vector of random variables and α refers to the level of accuracy required for detecting the tags from the available set. The resulting vector of the product $Q^s q(0)$ contains the probabilities of identifying k tags after s read cycles, $k = 0, 1, \ldots, n$. The above process of tag detection is thus an adaptive one achieved by iterating through the same sequence of steps to get an estimate of the number of tags in the environment which is reasonably close to the actual one. The algorithm is shown in Fig. 11 [15,17].

```
identifyStatic( ){
    N = 16; n_est = 0; stepN = 0;
    do {
        stepN++;
        c = performReadCycle(N);
        t = estimate(N, c);
        if (t > n_est){
            n_est = t;
            N0 = adaptFrameSize(N, n_est);
            if (N0 > N){
                stepN = 0; // restart with new frame size
                N = N0;
            }
        }
    }while(stepN < max Step(N, n_est));
}
```

Fig. 11. Procedure to adaptively read a static set of tags.

2.4. Contact-less protocol

The contact-less protocol [18] is based on the tree splitting methodology described earlier. The terminologies that the authors use for the reader and tag are transceiver and transponder, respectively. The basic idea is to identify one bit of the identification code in every arbitration step. One instance of an arbitration process will identify a tag uniquely. Each arbitration process will have N arbitration steps, where N is the number of bits in the identification code.

Initially, all the tags are in wait state and listen actively to the reader commands as shown in Fig. 12. Thus, all the tags are active in the beginning. The reader requests the active tags for a given bit position of their identification code during a particular slot in the arbitration step. The tags use a modulation scheme [22], which identifies a logical "0" in the specified bit position with "00*ZZ*" in a slot (where Z can be thought of as a tag transmission with no modulation). Logical "1" is identified with the sequence "*ZZ*00". In this way, the reader can recognize the responses from all the tags even though they have different bits in their identification sequence. This step divides the uniden-

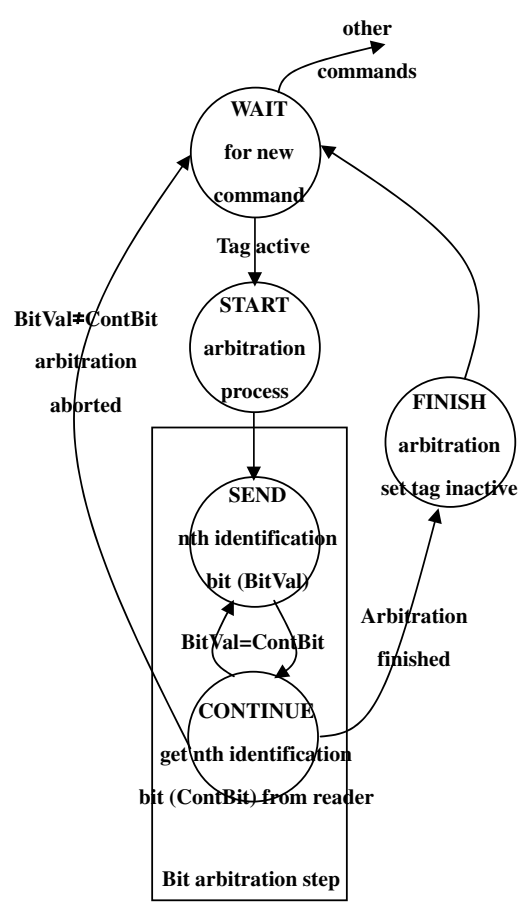

Fig. 12. A full arbitration process.

1. The reader sends out request for bit position i in the tags identification code.
2. The tags respond to this request by transmitting their i^{th} bit in the next slot. Due to the modeulation scheme used the reader is able to distinguish between **0** and **1** transmitted by the tag.
3. Three cases can be identified fepending on the response from the tags :
   i. All the tags have **0** or **1** in the requested bit position which puts all the tags in either the wait state or the bit arbitration step.
   ii. Some tags with **0** in their i^{th} bit position from the wait set and those with **1** in their i^{th} bit position continue with the bit arbitration process.
   iii. The ith bit was the Nth bit of the N bit long tag IDs. In this case a tag would be identified whose identification string is $\{0,1\}^{N-1}1$. Thus each bit arbitration step would lead to the discovery of a single tag. The reader would then retrace up the tree and discover the tag whose ID differs from that of the discovered tag in the least number of positions.
4. The reader attempts to discover the tags continuing with the bit arbitration. It makes the wait set tags increment their counter to remember their discovery statr in the tree. The reader recursively visits the nodes in the wait set after discovering existing nodes of asbitration step. All those nodes continue with the bit arbitration until all the N bit positions have been examined for each leaf node in the tree.

Fig. 13. Step by step for contact-less algorithm.

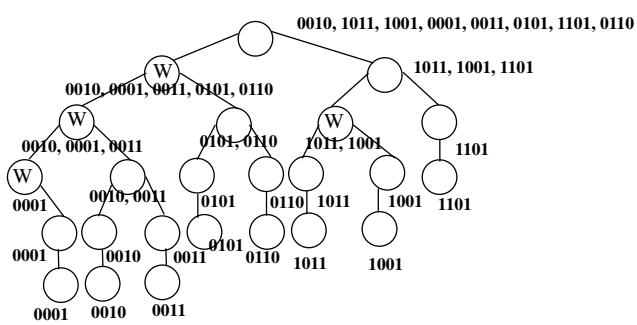

Fig. 14. Illustration of contact-less protocol.

tified tags into two sub groups, one had *0*'s in the requested bit position and the other had *1*'s. This is termed as the *Bit-Val* step.

The reader then chooses a continuation bit (*ConBit*), which could be *0* or *1*. It sends this *ConBit* to all the active tags. The tags with the same *ConBit* and *BitVal* values will remain active in this Bit arbitration step and the other tag group goes into the wait state. The algorithm will thus split a set of unidentified tags into smaller subsets in conformity with the tree splitting algorithm. Thus, each unidentified tag set is broken down into smaller subsets until a tag has been identified. The reader performs this mechanism recursively till it has identified all the tags in that group.

The algorithm can be described in a step-wise fashion as Fig. 13 [13].

The working of the protocol is illustrated with a small example in Fig. 14. Each tag has a 4 bit identification code and those that transmit a *0* in a particular bit arbitration step moves into the wait set with the others continuing with the arbitration process. The tags in the wait state have been labeled "W". The bit strings specified against each tag identifies the IDs of the tags in that subset.

3. Countermeasures of surveyed reader collision problem

A reader may interfere with the operation of other readers in the RFID system. For the reader collision problem, we need to minimize the number of frequencies needed to keep frequency interference at an acceptable level while simultaneously minimizing the amount of time required for all readers to communicate with tags in their respective interrogation zones.

The classification and solution of reader collision problems was shown in Fig. 15. In the following subsections, we described Color-wave and HiQ algorithms for finding the optimum solution to the reader collision problem and reviewed some examples of dynamic channel assignment algorithms, such as neural networks, simulated annealing and genetic algorithms.

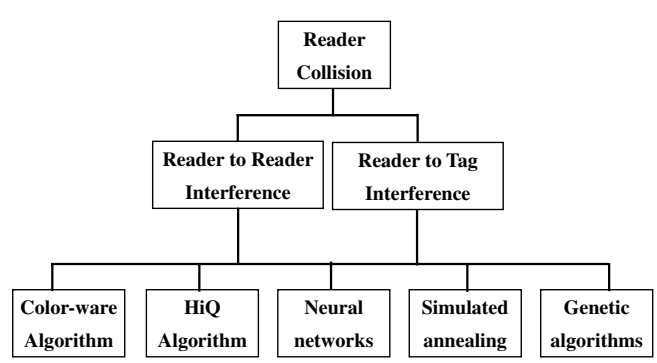

Fig. 15. Taxonomy and solution of reader collision problems.

3.1. Color-wave

To achieve optimal frequency channel assignment, a pair of distributed algorithms called Distributed Color Selection (DCS) and Variable-Maximum Distributed Color Selection (VDCS or Color-wave) are introduced [20,21]. The goal of both algorithms is to color a reader network such that each reader node has the smallest possible number of adjacent nodes with the same color. This approach allows easy reservation of time slots; a color is a periodic reservation for collision-free transmission of data. Color-wave additionally attempts to optimize the graph to the smallest number of total colors required to achieve a particular percentage of successful transmissions.

3.1.1. Distributed color selection

This approach allows easy reservation of time slots; a color is a periodic reservation for collision-free transmission of data. A reader with a queued request for transmission transmits only in its color (time slot). If the transmission collides with another reader, the transmission request is discarded. Furthermore, the reader randomly chooses a new color and reserves this color, causing all of its neighbors to select a new color. This switch and reservation action is referred to as a "kick".

The maximum colors variable (max_colors) is an input to the algorithm, and does not change throughout the functioning of the algorithm. Each reader keeps track of what color it believes the current time slot to be (timeslot_ID). The distributed nature of the algorithm does not require synchronization between timeslot_ID. DCS is implemented with three separate subroutines detailed in Fig. 16 [20]. The first subroutine manages transmissions. The second subroutine manages collisions and the reservation of a new "color" or time slot. The third subroutine manages kick resolution, or communication with other readers. Each reader must calculate for each kick received whether its current color is the color referenced by the kick.

DCS is a greedy algorithm; a node's chances of colliding immediately after experiencing a collision are minimized at the expense of its neighbors. However, a greedy algorithm caters more to our needs than an altruistic algorithm, as we wish a communicating reader to communicate with its tags as soon as possible after queuing the request.

3.1.2. Variable-maximum distributed color selection

In DCS, the maximum colors variable (max_colors) is fixed. In a reader network with a variable probability of transmission between nodes and times of day, a single input for colors in the algorithm does not provide for flexibility. Thus, a mechanism for dynamically changing the maximum number of colors available at a reader is needed. Color-wave [20,21] is a simple, distributed, on-line algorithm that finds a feasible solution to the reader collision problem. The distributed nature of the algorithm allows each reader to minimize collisions based upon local information. Global communication and information sharing are not required. The Color-wave algorithm is representative of algorithms that use a distributed system with no guaranteed method of communicating between neighboring nodes to manage a TDMA reservation anti-collision system among a network of readers.

In Color-wave, each reader monitors the percentage of successful transmissions. Five inputs to the algorithm determine when a reader changes its local value of max_colors:

1. UpSafe: the safe percentage at which to increase max_colors.
2. UpTrigs: the trigger percentage at which to increase max_colors if a neighboring reader is also switching to a max_colors higher than that of this reader.
3. DnSafe, DnTrig: analogues of UpSafe, UpTrig, except decreasing max_colors.

DCS Subroutine 1 - Transmission :

- If transmission requested :
  - If (timeslot_ID % max_colors) = = current_color
    - * then transmit
    - * else idle until (timeslot_ID % max_colors) = = current_color

DCS Subroutine 2 - Collision :

- If attempted transmission but experienced collision :
  - current_color = = random(max_colors)
  - broadcast kick ststing new color

DCS Subroutine 3 - Kick resolution :

- If kick received stating current_color
  - randomly change to different color within max_coloes.

Fig. 16. DCS pseudo-code.

Colorwave Subroutine 1 - Color Change

- If collision percentage is past SAFE threshold AND time spent in current max_color exceeds min_time threshold
  - Change max_color up or down one (depending on threshold exceeded)
  - Next iteration, initiate kick to new max_color.

Colorwave Subroutine 2 - Kick Resolution

- If kick received stating current_color
  - change to random color within max_colors OTHER THAN current_color.
- If kick received stating change to new max_color AND collision percentage is past TRIGGER threshold AND time spent in current max_color exceeds min_time threshold.
  - Change max_color to kicked value.
  - Next iteration, initiate kick to new max_color.

All DCS subroutines are also in use.

Fig. 17. Color-wave pseudo-code.

4. MinTimeInColor: the minimum number of time slots before the Color-wave algorithm will change max_colors again after initialization or changing max_colors.

Color-wave builds upon the DCS algorithm; the additional subroutines for Color-wave are detailed in Fig. 17 [20]. When a reader executing Color-wave reaches a Safe percentage to change its own value for max_colors, it will send out a kick to all neighboring readers. If the phenomenon that is causing it to exceed a Safe percentage is local to that reader, the other readers will not have passed their own Trig percentages and will not respond. However, if the phenomenon causing the collision value to exceed a Safe threshold is widespread, neighboring readers will most likely have exceeded their own Trig thresholds, and a "kick wave1" will ensue. As kicks spread from the initiating reader throughout the entire system, a large portion (or all) of the readers in a reader system may change their value of max_colors.

3.2. HiQ algorithm

Another anti-collision protocol for optimal frequency channel assignment is the HiQ algorithm [22]. The HiQ is an online algorithm based on Q-learning to solve the reader collision problem. Q-learning is a form of reinforcement learning [26,27]. Fig. 18 diagrams the agent-environment interaction. The environment is modeled as finite-state discrete-time stochastic dynamical system. The agent and environment interact at each of a sequence of discrete time steps, $t = 0, 1, 2, \ldots$ At each time step t, the agent receives some representation of the environment's state, $s_t \in S$, where S is the set of possible states, and on that basis selects an action, $a_t \in A(s_t)$, where $A(s_t)$ is the set of actions available in state s_t. One time step later, in part as a consequence of its action, the agent receives a numerical reward, $r_{t+1} \in R$, and finds itself in a new state, s_{t+1}. The goal of the Q-learning agent is to find an optimum policy $\pi*(s,a)$, which maximizes/minimizes the cumulative measure of the return (reward/cost) $r_t = r(s,a)$ received over time.

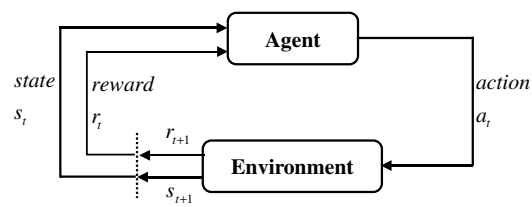

Fig. 18. The agent-environment interaction in reinforcement learning.

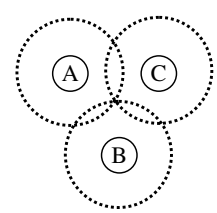

Fig. 19. Three readers configuration.

The HiQ algorithm utilizes a hierarchical control structure with three basic tiers. At the lowest tier are readers. The readers communicate when they have been granted a frequency and a time slot to do so. Readers only have knowledge of the frequency and time slot they have been allocated. They are capable of detecting collisions with other readers by communicating with other readers within its interrogation zone, those with overlapping interrogation zones. For example, in Fig. 19, Readers B and C are within the interrogation zone of Reader A. When A is granted a frequency and a time slot, it begins reading. During this time, it also pings its neighbors (B and C) to see if they are reading. Once Readers B and C receive the ping, they respond with whether or not they are reading at the time, and if they are, what frequency they are using. Reader A is responsible for the collisions detection processing. When Reader A receives the ping responses from Readers B and C, it checks to see if there are collisions, and if so, what type of collisions. If two readers are communicating using the same time slot, there will be tag interference. If these readers are also using the same frequency, the reader will

also experience frequency interference. The reader stores the number of collision a reader may experience. This information is also known at the tier above the readers – the reader-level server (R-server) tier. The reader tier consists of only a single level, and the readers communicate directly with only one R-server.

Q-learning servers (Q-servers) comprise the highest tier of the hierarchy in this algorithm. Q-servers allocate resources to the servers directly below them in the hierarchy. These are typically R-servers, but can also be other Q-servers. Q-servers can be arranged hierarchically themselves. Regardless of how many Q-server tiers there are, there is always a single root Q-server. The root Q-Server has global knowledge of all frequency and time slot resources, and is able to allocate them all. By interacting with the servers below, each Q-server allocates available frequencies and time slots to those servers. Unlike R-servers, Q-servers have no knowledge of constraints between individual readers. This information is inferred through interaction with the servers at the tier directly below.

The cost function is essential to the success of the Q-learning agent and is the same at each tier in the Q-learning hierarchy. The cost function is the return $r(s,a)$ described as outlined previously in the Q-learning algorithm. Q-servers attempt to find an optimum policy that allocates resources to the tiers below in such a way as to minimize the cost function. The details implement for Q-learning algorithm for the reader collision problem had been explained in [22].

3.2.1. Optimality function

The return $r(s,a)$ evaluates the cost of the action taken. The cost is calculated by the following equation:

$$c(s,j,k) = n_1(j,k)c_1 + n_2(j,k)c_1 + n_3(j,k)c_3 + n_4(j,k)c_4 + n_5(j,k)c_5 + n_6(j,k)c_6.$$

Table 4 is the information fields contained in above cost equation.

It is important that the sub-costs relating the number of interferers are small enough that they do not dominate the other parts of the equation. This information is designed to help the Q-learning agent infer more information about the interference patterns in the system [22].

Table 4
The information contained in above cost equation

$n_1(j,k)$	The rate of successful communication	$n_1(s,j,k) = \frac{\text{grants}-\text{collisions}}{\text{grants}}$
$n_2(s,j,k)$	The number of tag interferers per collision	$n_2(s,j,k) = \frac{\text{tag interferers}}{\text{collisions}}$
$n_3(s,j,k)$	The number of frequency interferers per collision	$n_3(s,j,k) = \frac{\text{frequency interferers}}{\text{collisions}}$
$n_4(j,k)$	Tag collision rates	$n_4(s,j,k) = \frac{\text{tag collisions}}{\text{grants}}$
$n_5(j,k)$	Frequency collision rates	$n_5(s,j,k) = \frac{\text{frequency collisions}}{\text{grants}}$
$n_6(j,k)$	The number of rejections per request	$n_6(s,j,k) = \frac{\text{rejections}}{\text{requests}}$

3.2.2. State description

In a network of N nodes and R available resources, the state s at time t is given by

$$s_t = (i, A(i))_i,$$

where $i \in \{1,2,\ldots,N\}$ specifies the index and of the node making a resource request and $A(i) = \{1,2,\ldots,R\}$ which is the number of available resources to node i at time t. The set of R available resources is different than the set of available frequencies described in the Frequency Assignment Problem (FAP) version. In addition to frequencies, another available resource in RFID networks is time slots. Given a set of F frequencies and L time slots, the total number of available resources in the network is

$$R = F^{*}L.$$

To obtain $A(i)$, the number of unique frequency and time slot combinations available to a node i, several other quantities must be defined.

(1) Frequency usage status for node $q, q = 1,2,\ldots,R$, is defined as a $F \times L$ matrix where

$$u_{j,k}(q) = \begin{cases} 1 & \text{if frequency } k \text{ and time slot } j \text{ are in use in node } q, \\ 0 & \text{otherwise,} \end{cases}$$

and $q = 1,2,\ldots,F$ and $j = 1,2,\ldots,L$, and $k = 1,2,\ldots,F$.

(2) Resource availability matrix $r(q) \in \{0,1\}^{\mathrm{LF}}$ where

$$r_{j,k} \begin{cases} 0 & \text{if frequency } k \text{ and time slot } j \text{ available for use in node } q, \\ 1 & \text{otherwise,} \end{cases}$$

and q, j, and k take on the same values as in the usage status matrix.

(3) Available resources: $A(i)$ is given by a set $x_{j,k}(q)$ where

$$x_{j,k}(q) = \begin{cases} 1 & \text{if frequency } k \text{ and time slot } j \text{ canbe used in node } q, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$x_{j,k}(q) = r_{j,k}(q) \wedge \neg u_{j,k}(q)$$
$$q = 1,2,\ldots,R$$

Given $x_{j,k}(q)$, $A(i)$ is obtained with the summations

$$A(i) = \sum_{j=1}^{L} \sum_{k=1}^{F} x_{j,k}(i).$$

3.2.3. Actions

The agent performs an action in this environment by assigning a frequency k and time slot j from $A(i)$ to node i's request. Here, a is defined as

$a = j, k \in \{1, 2, \ldots, R\}$ and $x_{j,k}(i) = 1$.

3.3. Other algorithms

There have been a great number of algorithms developed for solving the frequency assignment problem. The algorithms vary greatly and have many fundamental differences, depending on the type of frequency assignment problem they are trying to solve. Many rely on centralized control for channel assignment, while others boast distributed control where individual transmitters are responsible for communication. Some examples of dynamic channel assignment algorithms are neural networks, simulated annealing and genetic algorithms [28–30].

In a neural network, a system of artificial neurons is mathematically created. In this system, the neurons represent a single cellular base station and a single channel that base station can use. The constraints and optimization criteria of the system are translated into an energy function, which the neural network tries to minimize. Calculating this energy function and combining it with weighted input parameters results in a state for each neuron. The output of these neurons is based on the internal state of each neuron and determines whether or not a channel can be used in a base station. With this algorithm, a network of 25 base stations with 73 channels each was able to converge to an optimum assignment [28].

Simulated annealing is a generalization of local search. The algorithm starts with an initial solution and begins making random changes to channel assignments in a radio network. The algorithm always accepts assignment changes that improve the overall reward, and accepts changes that do not improve the reward (deteriorations) with some probability. The algorithm uses a control variable T to restrict the number of deteriorations and stops when T reaches its terminal value. Simulated annealing algorithms have been shown to be capable of converging to assignment that did not violate frequency constraints 32–52% of the time, depending on the complexity of the network [29].

Genetic algorithms are a form of blind local search. In a genetic algorithm approach to channel assignment, a single solution is an assignment scheme for all the base stations. The genetic algorithm takes a set of solutions as the population, and begins the evolutionary process. First, members of the population are selected based on their fitness (performance measure). To avoid converging at local minima (when only the fittest are selected) the selection is done in a biased random manner: fitter members are more likely to be chosen than the weaker members. Through crossover, information is exchanged between members of the selected population, resulting in reproduction that creates new solutions. During reproduction, genes (base stations) may undergo additional random changes at a mutation rate. The process is repeated until a solution is found where no interference constraints are violated [30].

4. Comparative view of surveyed protocol

The problems that occur in data transferring between readers and tags in an RFID system are shown in Fig. 20. In Fig. 20(a), it explains the problem in transmission of data from many individual tags to the reader. When a reader attempts to obtain the unique ID number of each tag, messages from the tags can collision and cancel each other out. Fig. 20(b) explains the problem when two readers with overlapping interrogation zones communicate using the same frequency at the same time, each may encounter difficulty in communication with tags due to interfering signals transmitted by the other reader. Fig. 20(c) explains the problem when one tag is simultaneously located in the interrogation zones of two or more readers and more than one reader attempts to communicate with that tag at the same time. Regardless of Figs. 20(b) or (c), interference detected by one reader or caused by another reader is referred to as a reader collision and tag interference can occur without reader interference while reader interference always has tag interferences as well. We have surveyed all possible protocols, classified these problems and shown them in Fig. 21.

4.1. Tag collision

Anti-collision methods in tags have similarities to anti-collision algorithms in networking. However, RFID tags pose a number of problems that arise from the very limited resources. A common classification of anti-collision algorithms is based upon how the tags respond. In probabilistic

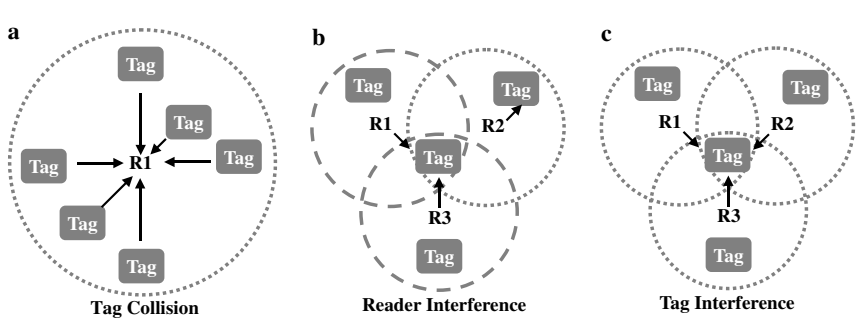

Fig. 20. Data transferring between readers and tags in an RFID system.

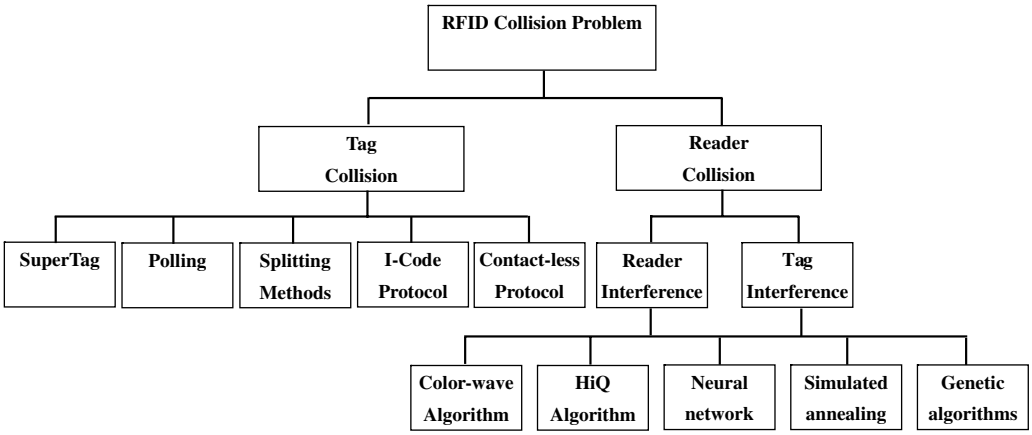

Fig. 21. Taxonomy of anti-collision resolution for RFID collision problem.

algorithms, the tags respond at randomly generated times. Many probabilistic algorithms are based on the Aloha scheme in networking [31]. The times at which readers can respond can be slotted or continuous. Deterministic schemes are those in which the reader sorts through tags based on their unique identification number. The simplest deterministic scheme is the splitting method (binary tree-walking scheme), in which the reader traverses the tree of all possible identification numbers.

Sarma et al. [32] described the performance metrics that are traded-off by these algorithms and their variants include: (1) the speed at which tags can be read, (2) the outgoing bandwidth of the reader signal, (3) the bandwidth of the return signal, (4) the amount of state that can be reliably stored on the tag, (5) the tolerance of the algorithm to different types of noise in the field, (6) the cost of the tag, (7) the cost of the reader, (8) the ability to tolerate tags which enter and leave the field during the inventory-taking process, (9) the desire to count tags exactly as opposed to sampling them, and finally, and (10) the range at which tags can be read.

Table 5 shows a contrast with most multiple access collision-resolution methods surveyed in this paper. An interesting conclusion to draw from Table 5 is that no particular algorithm seems to emerge as the preferred method to resolve collision. The choice of algorithm for implementation in systems aimed for the market is fairly well distributed between Aloha and its two derivatives, Slotted-Aloha and Framed-slotted Aloha. In deterministic anti-collision algorithms, polling, tree algorithm, and QT protocol have the same time complexity and read accuracy rate for $n_1 = n_2$. If there is a cost consideration, polling protocol is the choice. If we emphasize cost and time both, QT protocol will be better. In probabilistic anti-collision algorithms, tag-driven procedures are slower and more inflexible than reader-driven procedures. Therefore, if there is time concern, I-Code protocol is a better choice. Table 5 also shows that most

Table 5
Comparison of tag collision-resolution

Criteria	Protocols					
	Tag-driven	Reader-driven				
	SuperTag	Polling	Splitting methods		I-Code	Contact-less
			Tree algorithm	QT protocol		
Deterministic		∨	∨	∨		∨
Probabilistic	∨				∨	
TTF/RTF	TTF	RTF	RTF	RTF	RTF	RTF
Aloha	∨					
Slotted-Aloha			∨	∨		
Framed-slotted Aloha					∨	
Switch-off	∨	∨	∨	∨		∨
Slow-down	∨					∨
Terminating/Muting					∨	
Reader "Carrier Sense"	∨					∨
Accuracy level	Close to 100%	100%	100%	100%	Close to 100%	100%
Time complexity	–	$O(n_1)$	$O(n_2)$	$O(n_1)$	$t_0 * s + T_N$	$O(2^N)$
Message complexity	–	$O(m(N+1))$	$mlong_2^m$	$2.21klong_2^n + 4.19k$	$N(m*p) + N(m*s)$	$O(m(N+1))$

Where n_1, total number of unique tags possible with a possible with a tag id of length k; n_2, the number of tags that need to be identified; T_N, time required to estimate N; N, length of the tag identification number; m, the number of tags that needs to be identified; $(m*p)$, number of read cycles required to estimate N; $(m*s)$, number of read cycle performed with fixed N.

Table 6
Comparison of reader collision-resolution

Method	Criteria				
	Optimize function	Centralized control	Distributed control	Fixed Channel Assignment (FCA)	Dynamic Channel Assignment (DCA)
Colorwave algorithm	Smallest possible color number		∨	∨	∨
HiQ algorithm	Cost function	∨	∨		∨
Neural network	Energy function		∨	∨	
Simulated annealing	Reward value	∨		∨	
Genetic algorithm	Performance value	∨		∨	

systems encountered implement at least one of the extensions mentioned.

4.2. Reader collision

The solution of a reader collision problem is to allocate frequencies over time to a set of readers without collision. Table 6 shows a contrast with five algorithms for resolving the reader collision problem. The algorithms vary greatly and have many fundamental differences. Many rely on centralized control for channel assignment, while others boast distributed control where individual transmitters are responsible for communication. Reader collision problems have some similarities to frequency assignment problems in mobile telephone systems. However, RFID systems are unable to discriminate between two readers communicating with them simultaneously. Therefore, two readers that communicate with the same tag must communicate at different times. In a cooperative, trusted environment, reader collisions can be handled in a fairly seamless way. Nevertheless, complications may arise in the execution of commands that change the state of the tag. If another reader interrupts the reader executing a series of state changing actions, it may be forced to relinquish control over the tag. The new reader that acquires the tag may further change the state of the tag without the cooperation of the first reader. Transactions between readers and tags must therefore be brief and atomic.

5. Conclusion

We have surveyed and classified a series of anti-collision protocols for RFID arbitration in this paper. In tag identification problem, the challenge that emerges for RFID systems where multiple tags are present in the reader's field is maximizing the number of tags accessed for information, while simultaneously minimizing the number within the time needed to do so. In many ways, the reader collision problem is simpler than the frequency assignment problem. However, the time and frequency usage constraints on the readers and the RFID system have a more significant impact on the performance of the RFID system. The phenomenon of tag interference is specific to the reader collision problem and is not found in the previously studied frequency assignment problem.

A comparative view of surveyed protocol was concluded in Section 5 which are provided for other researchers to improve. In further research, it remains challenging to find a procedure that would take environmental effects into account. We hope this paper will help implement pervasive computing environments that employ RFID systems.

Acknowledgments

The author thanks to the National Science Council of Taiwan for Grants NSC 94-2218-E-194-005 and NSC 94-2213-E-224-036 to part of this research.

References

[1] B. Johansson, An Introduction to RFID – Information Security and Privacy Concerns, TDDC03 Projects, Spring, 2004.
[2] S.A. Weis, S. Sarma, E.R.L. Riovest, D.W. Engels, Security and privacy aspects of low-cost radio frequency identification systems, in: Lecture Notes in Computer Science, vol. 2802, 2004, pp. 201–212.
[3] A. Juels, R. Pappu, Squealing euros: privacy protection in RFID-enabled banknotes, in: Lecture Notes in Computer Science, vol. 2742, 2003, pp. 103–121.
[4] S.H. Weigart, Physical security devices for computer subsystems: a survey of attacks and defenses, in: Lecture Notes in Computer Science, vol. 1965, 2000, pp. 302–317.
[5] Royal Air Force, History, 1940. <http://www.raf.mod.uk/history/line1940.html/>.
[6] R. Anderson, M. Kuhn, Low cost attacks on tamper resistant devices, in: Lecture Notes in Computer Science, vol. 1361, 1997, pp. 125–136.
[7] A. Juels, Minimalist cryptography for low-cost RFID tags, in: Lecture Notes in Computer Science, vol. 3352, 2004, pp. 149–164.
[8] A. Juels, R.L. Rivest, M. Szydlo, The blocker tag: selective blocking of RFID tags for consumer privacy, in: ACM Conference on Computer and Communications Security 103–111, Washington, DC, USA, October, 2003.
[9] A.R. Koelle, S.W. Depp, J.A. Landt, R.E. Bobbett, Short-range Passive Telemetry by Modulated Backscatter of Incident CW RF Carrier Beams, Biotelemetry, vol. 3, 1976, pp. 337–340.
[10] K. Finkenzeller, RFID Handbook: Radio-Frequency Identification Fundamentals and Applications, John Wiley, New York, 2000, pp. 200–219.
[11] D. Krebs, M.J. Liard, White Paper: Global Markets and Applications for Radio Frequency Identification, Venture Development Corporation (2001).
[12] T.S. Flor, W. Niess, G. Vogler, RFID: the integration of contact-less identification technology and mobile computing, in: Seventh Inter-

national Conference on Telecommunications, vol. 2, Zagreb, Croatia, June 11–13, 2003, pp. 619–623.
[13] C. Law, K. Lee, K.Y. Siu, Efficient memory-less protocol for tag identification, in: Proceedings of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Boston, Massachusetts, USA, August 11, 2000, pp. 75–84.
[14] D.R. Hush, C. Wood, Analysis of tree algorithms for RFID arbitration, in: IEEE International Symposium on Information Theory, 16–21 August, 1998, p. 107.
[15] Harald Vogt, Multiple object identification with passive RFID tags, in: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, vol. 3, Hammamet, Tunisia, 6–9 October, 2002, 6 pp.
[16] F. Zhou, D. Jin, C.L. Huang, M. Hao, Optimize the power consumption of passive electronic tag for anti-collision schemes, in: Proceedings of the Fifth International ASIC Conference, October, 2003, pp. 1213–1217.
[17] Harald Vogt, Efficient object identification with passive RFID tags, in: Proceedings International Conference on Pervasive Computing, April, 2002, pp. 98–113.
[18] M. Jacomet, A. Ehrsam, U. Gehrig, Contact-less identification device with anti-collision algorithm, in: IEEE Computer Society, Conference on Circuits, Systems, Computers and Communications, Athens, 4–8 July, 1999.
[19] D.W. Engels, S.E. Sarma, The reader collision problem, in: Proceedings of IEEE International Conference on Systems, Man and Cybernetics, vol. 3, Hammamet, Tunisia, 6–9 October, 2002, 6 pp.
[20] J. Waldrop, D.W. Engels, S.E. Sarma, Colorwave: an anticollision algorithm for the reader collision problem, in: IEEE Wireless Communications and Networking Conference, March, 2003, pp. 1206–1210.
[21] J. Waldrop, D.W. Engels, S.E. Sarma, Colorwave: A MAC for RFID reader networks, in: IEEE Wireless Communications and Networking Conference, 2003, pp. 1701–1704.
[22] J.K. Ho, Solving the Reader Collision Problem with a Hierarchical Q-learning Algorithm, S.B. Electrical Engineering and Computer Science Massachusetts Institute of Technology, 2001.
[23] C. Abraham, V. Ahuja, A.K. Ghosh, P. Pakanati, Inventory Management using Passive RFID Tags: A Survey, Department of Computer Science, The University of Texas at Dallas, Richardson, Texas, pp. 1–16, October, 2002.
[24] J.I. Capetanakis, Tree algorithms for packet broadcast channels, IEEE Trans. Informat. Theory 25 (1979) 505–515.
[25] D. Bertsekas, R. Gallager, Data Networks, second ed., Prentice-Hall, Englewood-Cliffs, NJ, 1992.
[26] S. Haykin, J. Nie, A dynamic channel assignment policy through Q-learning, IEEE Trans. Neural Networks 10 (1999) 1443–1455.
[27] R.S. Sutton, A.G. Barto, Reinforcement Learning: An Introduction, A Bradford Book, 1998, pp. 51–52.
[28] D. Kunz, Channel assignment for cellular radio using neural networks, IEEE Trans. Vehicular Technol. 40 (1991) 188–193.
[29] M. Duque-Anton, D. Kunz, B. Ruber, Channel assignment for cellular radio using simulated annealing, IEEE Trans. Vehicular Technol. 42 (1993) 14–21.
[30] W.K. Lai, G.G. Coghill, Channel assignment for a homogenous cellular network with genetic algorithms, IEEE Trans. Vehicular Technol. 45 (1996) 91–96.
[31] B. Bing, Broadband Wireless Access, Kluwer Academic Publishers, Boston, 2000.
[32] S.E. Sarma, S.A. Weis, D.W. Engels, RFID Systems and security and privacy implications, in: Lecture Notes in Computer Science, vol. 2523, 2003, pp. 454–469.

Dong-Her Shih received his Ph.D. degree in Electrical Engineering from National Cheng Kung University, Taiwan, in 1986. He is a professor of Information Management Department, National Yunlin University of Science and Technology, Douliu, Yunlin, Taiwan. He has published over 30 articles in refereed information system journals. He also serves as an Asian Editor in International Journal of Mobile Communications. His current researches include network security, intrusion detection, wireless network, E-commerce security, RFID systems and peer-to-peer network.

Po-Ling Sun is a graduate student in Information Management, National Yunlin University of Science and Technology, Douliu, Yunlin, Taiwan, from 2003. Her current interests include E-commerce security and RFID systems.

David C. Yen is a professor of MIS and chair of the Department of Decision Sciences and Management Information Systems at Miami University. He received a Ph.D. in MIS and Master of Sciences in Computer Science from the University of Nebraska. Professor Yen is active in research, he has published three books and many articles which have appeared in Communications of the ACM, Decision Support Systems, Information and Management, International Journal of Information Management, Information Sciences, Journal of Computer Information Systems, Interfaces, Telematics and Informatics, Computer Standards and Interfaces, Information Society, Omega, International Journal of Organizational Computing and Electronic Commerce, Communications of AIS, and Internet Research among others. He was also one of the co-recipients for a number of grants such as Cleveland Foundation (1987–1988), GE Foundation (1989), and Microsoft Foundation (1996–1997).

Dr. Shi-Ming Huang received his Ph.D. degree at the School of Computing and Information Systems, University of Sunderland, U.K. He is currently a Head of Accounting and Information Technology Department and a Director for the Research Center of e-Manufacturing and e-Commerce at National Chung Cheng University, Taiwan. He is also a professor in the Department of Information Management. He is 2006 President for International Chinese Information Systems Association (ICISA). He has published five books, three business software and over 40 articles in refereed information system journals, such as Decision Support Systems, Information and Management, Journal of Computer Information Systems, European Journal of Operational Research, Journal of Database Management, ACM SIGMOD, etc. He has received over 10 achievement awards in information system area and serves as editorial board member in several international journals. He also serves as an associate editor in International Journal of Management and Enterprise Development.