RFID Reader Networks : Channel allocation algorithms, performance evaluation and simulator

John Sum, Kevin Ho

Abstract—**Assigning neighboring RFID readers with non-overlapping interrogation time slots is one approach to solve the reader collision problem. In which, Distributed Color Selection (DCS) and Colorwave algorithm have been developed, and simulated annealing (SA) technique have been applied. Some of them (we call them non-progressive algorithms), like DCS, require the user to pre-defined the number of time slots. While some of them (we call them progressive), like Colorwave, determine the number automatically. In this paper, a comparative analysis on both non-progressive and progressive algorithms to solve such a problem in a random RFID reader network is presented. By extensive simulations on a dense network consisting of 250 readers whose transmission rates are 100%, a number of useful results have been found. For those non-progressive type algorithms, it is found that DCS is unlikely to generate a collision-free solution, even the number of time slots is set to 20. On the other hand, heuristic and SA-based algorithms can produce collision-free solutions whenever the number of time slots is set to 16. For the cases when the number of time slots is not specified, heuristic-based, SA-based and Colorwave algorithms are all able to determine the number automatically and thus generate collision-free solution. However, SA-based algorithms require much longer time than the heuristic-based and Colorwave algorithms. In terms of distribution uniformity, it is found that SA-based algorithms can generate almost evenly distributed time slot allocations. That is to say, the number of readers assigned and hence the volume of data sending to the host computer in each time slot will almost be equal. No special transmission protocol is needed to handle the communication between readers and the host computer. Finally, the design of a simulator for the analysis of the effect of reader malfunction on the performance of the progressive channel allocation algorithms is elucidated. With the aid of the simulator, system developer is able to design a more effective management system for RFID reader networks.**

Keywords : Colorwave, Distributed Color Selection, Reader collision problem, Reader network, RFID systems, Simulated annealing, Simulator.

I. Introduction

Applications of RFID technologies have been succeeded in various industrial sectors, particularly in the retail business and medical & health industry [16], [19], [22], [23]. In recent years, adoption and implementation of RFID systems have also indicated a dramatic growth and the growth will be continue in subsequent years [8], [22]. Scope of applications have also been expanded from inventory control [1] to social networking for dogs [4]

A general design of RFID system with unstructured locations of readers is shown in Figure 1. It consists of three types of components, (a) RFID readers, (b) RFID tags and (c) a control computer. Readers are deployed in an unstructured locations and each has a unique id. Readers communicate with the control computer over usual wireless LAN channel [15], and pass the data to it for storage. An RFID tag is attached to a single object. Again, each tag has its own id for identification and specific information, such as product code and customer id, are stored inside for retrieval. Each reader talks with the tags within its vicinity through a specific communication protocol [2], [6], [7].

John Sum is with the Institute of Technology Management, National Chung Hsing University, Taichung, Taiwan. (email: pfsum@nchu.edu.tw)

Kevin Ho is with the Department of Computer Science and Communication Engineering, Providence University, Sha-Lu, Taiwan (email: ho@pu.edu.tw)

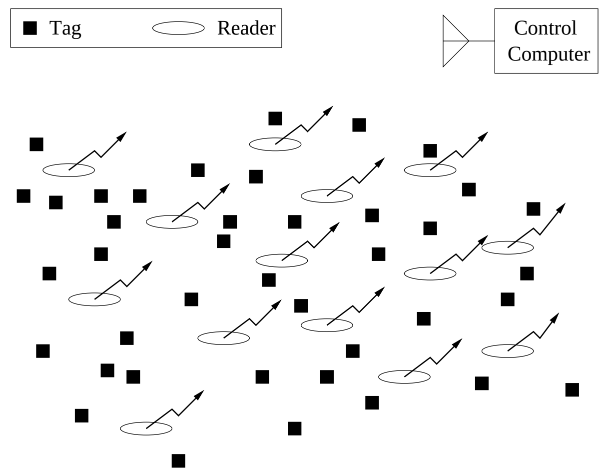

Fig. 1. An RFID reader network, with readers communicating to the control computer via a wireless LAN.

The control computer is served as a centralized server for the readers. On one hand, it coordinates the interrogations between the readers and the tags so as to avoid interferences (i.e. collisions) between one and other. On the other hand, it receives data transmitted from the readers, consolidates and then deposits in a database.

Since the communication amongst readers and tags rely on radio transmission, each tag (or reader) can simultaneously receive multiple signals from different readers. Interference can easily occur amongst readers and tags if two of more readers use the same frequency band for interrogation and their locations are within the transmission range of one another. These two phenomena are usually called the reader-to-tag collision and reader-to-reader collision [5], [17]. In simple words, it is called reader collision problem. As data transmission can largely be hampered by collision, many researches have been done in the last few years in order to overcome such effect [3], [9], [10], [12], [13], [20], [21], [24]. Their approaches to this problem are essentially the same as time divisional multiplexing. In which, each reader is assigned with a dedicated time slot for interrogation. Within the time slot, no other neighbor reader is allowed to use the channel for interrogation. Clearly, this time slot allocation problem is equivalent to a Graph Coloring Problem, meaning that the problem is NP completed. No polynomial time algorithm exists for the solution [14].

As most researches focus on simple network structure and regular topologies, this paper presents a study on dense network of unstructured topology. The performance of a heuristic algorithm and three variants of simulated annealing algorithms for reader collision problem are examined. Empirical analysis on the properties of their solutions obtained will be made. A discussion with Colorwave and Distributed Color Algorithm is remarked. A conceptual model for a reader network management system is presented and the design of a simulator for analysis and design of network management is elucidated.

The rest of the paper is organized as follows. The reader collision problem will be described in the next section. The non-progressive algorithms, for solving such problem, including a heuristic algorithm, variants of simulated annealing algorithms and Distributed Color Selection algorithm will be described in Section 3. While the progressive algorithms including the heuristic-based, SA-based and Colorwave algorithms will be described in Section 4. Section 5 presents the simulation results on a dense reader networks consisting of 250 readers and their average node degree is 15. Section 6 presents the design of the simulator. A few aspects regarding to the allocation algorithms are then discussed in Section 7. Finally, Section 8 gives a conclusion of the paper.

II. Channel Allocation Problem

Channel allocation problem is a key problem in RFID reader network management, in which neighboring readers are assigned with different channels. So that, collision amongst readers can be avoided. To solve the collision-free time slot allocation problem, the following assumptions are made. (1) A pair of readers will have collision if their distance apart is within a range r and they are collecting data in the same time slot. (2) Readers are not able to select their frequency bands. (3) The readers are deployed uniformly random within an area of $100m \times 100m$. Their locations are fixed once they have been deployed. (4) Each reader can only assigned with one time slot in a cycle for interrogation. (5) No mobile reader is allowed within the area of deployment.

Besides, we suppose that the time slot allocation algorithm is solved by the control computer. Once the solution has obtained, the control computer will send message informing the readers the time slot being assigned. The readers will thus record their time slots being assigned and wait for the synchronization signal from the control computer. Once the signal has been received, each reader will then operate to interrogate at the dedicated time slot and reading the tags' data. Whenever its interrogation has been finished, the reader will send data back to the control computer in the consecutive time slot. Communication between the control computer and the RFID readers are implemented by wireless LAN.

A. Formal definition of collision

Mathematically, a reader network of N readers can be represented by a graph $\mathcal{G} = (V, E)$. Here V the index set of

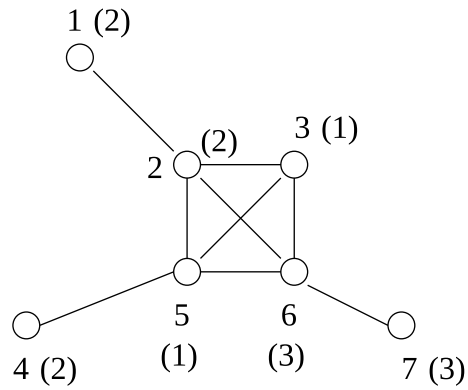

Fig. 2. A simple reader network. The numbers inside the parenthesis correspond to the time slots being assigned.

readers, i.e.

$$V = \{1, 2, \cdots, N\}$$

and the time slots assignment is denoted by τ, where

$$\tau = (\tau_1, \tau_2, \cdots, \tau_N).$$

$E = (e_{ij})_{N\times N} \in \{0,1\}^{N\times N}$ is a binary matrix corresponding to the neighborhood relation, i.e.

$$e_{ij} = \begin{cases} 1 & \text{if } d(i,j) \le r \\ 0 & \text{if } d(i,j) > r. \end{cases} \quad (1)$$

for all $i, j = 1, 2, \cdots, N$ and $i \neq j$. Here $d(i,j)$ is the distance between the reader i and the reader j. Besides, a reader cannot be a neighbor of itself, i.e. $e_{ij} = 0$. Let τ_i (for $i = 1, 2, \cdots, N$) be the time slot being allocated for the i^{th} reader. The collision matrix $C = (c_{ij})_{N\times N} \in \{0,1\}^{N\times N}$ can then be defined. For $i, j = 1, 2, \cdots, N$ and $i \neq j$,

$$c_{ij}(\tau_i, \tau_j, e_{ij}) = \begin{cases} 1 & \text{if } \{e_{ij} = 1\} \text{ and } \{\tau_i = \tau_j\} \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

For illustration, Figure 2 shows a simple reader network comprising seven readers. We assume that there are three time slots available. The matrix E and C are given by

$$E = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Heuristic Algorithm

S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random slots allocation.
S2 Random select a reader, say i.
S3 If reader's slot assignment has no collision to its neighbor readers, then goto S2.
S4 If reader's slot assignment has collision to its neighbors, select a new $\tau_i^* \in \{1, 2, \cdots, T\}$ such that the number of collision pairs between the reader and its neighbors is the minimum.
S5 Repeat steps S2 to S4 until no more improvement can be made.

Fig. 3. Heuristic Algorithm

B. Measure of reader collisions

With this definition, the number of collision pairs (CP) in a reader network can be defined as follows :

$$CP(\tau, E) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}(\tau_i, \tau_j, e_{ij}). \tag{3}$$

Which is a measure to the quality of a slots allocation and a reader collision problem can hence be stated. For a RFID reader network $\mathcal{G}$ of N readers and the maximum number of time slots in a cycle is T, find an allocation τ such that $CP = 0$. Clearly, this problem is equivalent to the Graph Coloring Problem that is NP hard [14].

III. Channel Allocation Algorithms: Non-Progressive Approach

Various algorithms can be applied to solve this collision avoidance problem. Hereby, we simply focus on a heuristic algorithm and three simulated annealing algorithms. All these algorithms carry out in an iterative fashion.

A. Heuristic

For the heuristic approach, a reader is randomly selected in each iteration and is assigned with a time slot that can minimize the number of collision pairs. Let say, the i^{th} reader has been selected, the time slot τ_i^* to be assigned satisfies the following heuristic rule.

$$\sum_{j=1}^{N} c_{ij}(\tau_i^*, \tau_j, e_{ij}) \leq \sum_{j=1}^{N} c_{ij}(\tau_i, \tau_j, e_{ij}), \tag{4}$$

for all $\tau_i \neq \tau_i^*$. The steps can be summarized in Figure 3.

B. Simulated Annealing

For the simulated annealing approach, the steps are similar except that the change of new time slot is not deterministic. Let Δ be the change of total number of collision pairs in the k^{th} iteration and the i^{th} reader has been selected.

$$\Delta = \sum_{j=1}^{N} c_{ij}(\tau_i^*, \tau_j, e_{ij}) - \sum_{j=1}^{N} c_{ij}(\tau_i, \tau_j, e_{ij}). \tag{5}$$

Simulated Annealing Algorithm

S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random slots allocation. $k = 1$.
S2 Random select a reader, say i. $k = k + 1$.
S3 If reader's slot assignment has no collision to its neighbor readers, then goto S2.
S4 If reader's slot assignment has collision to its neighbors, random generate a new $\tau_i^* \in \{1, 2, \cdots, T\}$.
S5 If the number of collision pairs between the reader and its neighbors reduces, $\tau_i = \tau_i^*$.
S6 If the number of collision pairs between the reader and its neighbors increases by Δ, generate a random number u from a uniform distribution in $[0, 1]$. Then,

$$\tau_i = \begin{cases} \tau_i^* & \text{if } u \leq \exp(-\Delta/\lambda(k)), \\ \tau_i & \text{if } u > \exp(-\Delta/\lambda(k)). \end{cases} \tag{7}$$

S7 Repeat steps S2 to S6 for $k \leq MaxRun$.

Fig. 4. Simulated annealing algorithm

The transition probability of the change of a reader's time slot from τ_i to τ_i^* is given by

$$P(\text{New Time Slot} = \tau_i^*) = \begin{cases} 1 & \text{if } \Delta \leq 0 \\ \exp\left\{-\frac{\Delta}{\lambda(k)}\right\} & \text{if } \Delta > 0. \end{cases} \tag{6}$$

$\lambda(k)$ is the cooling temperature and it decreases as k increases, see Step S6 in Figure 4. Simulated annealing can avoid the search being stuck in a local minimum.

Three common cooling schemes are used.

a) Constant Temperature: For all $k \geq 1$ and $a \ll 1$ is a small constant,

$$\lambda(k) = a. \tag{8}$$

b) Geman-Geman Rule: For all $k \geq 1$ and $b > 0$,

$$\lambda(k) = \frac{b}{\log(k+1)}. \tag{9}$$

c) Kirkpatrick et al Rule: For all $k \geq 1$, $0 < \alpha < 1$ and $c > 0$

$$\lambda(k+1) = \alpha\lambda(k), \quad \lambda(1) = c. \tag{10}$$

As mentioned in the literature, Equation (6) or Equation (7) plays a role to avoid the search to be stuck in a local minima. Heuristic algorithm on the contrary provides only local minimum solution.

C. Distributed Color Selection

Distribute Color Selection (DCS) resembles the heuristic algorithm mentioned before. Except that the new slot being selected for a reader if collision has encountered is random, see Step S4 in Figure 5.

IV. Channel Allocation Algorithms: Progressive Approach

Prior to search for a time slot allocation, the number of the time slots (i.e. T) for allocation is usually not available. Owing not to guess the number for T, an algorithm that can

Distributed Color Selection Algorithm

S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random slots allocation.
S2 Random select a reader, say i.
S3 If reader's slot assignment has no collision to its neighbor readers, then goto S2.
S4 If reader's slot assignment has collision to its neighbors, randomly select a new $\tau_i^* \in \{1, 2, \cdots, T\}$ other than the previous assignment.
S5 Repeat steps S2 to S4 until no more improvement can be made.

Fig. 5. Distributed color selection.

Progressive Heuristic Algorithm

S0 T is initialized to a small integer.
S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random slots allocation.
S2 Random select a reader, say i.
S3 If reader's slot assignment has no collision to its neighbor readers, then goto S2.
S4 If reader's slot assignment has collision to its neighbors, select a new $\tau_i^* \in \{1, 2, \cdots, T\}$ such that the number of collision pairs between the reader and its neighbors is the minimum.
S5 Repeat steps S2 to S4 until no more improvement can be made.
S6 Check if the solution obtained is collision-free.
S7 Assign all the collided readers with a number $T + 1$.
S8 Repeat S2 to S7 with $T = T + 1$, until the solution is collision-free.

Fig. 6. Progressive heuristic algorithm.

automatically determine the number will be an advantage. One approach is to progressively increase the number of available time slots after each cycle of search. Suppose the total number of available time slots is initially set to a small integer, say 4. Then, apply either heuristic or simulated annealing algorithm to search for a solution for allocation. If the solution obtained is not collision-free, increment the number of T by one and then re-run the algorithm. The steps are repeated until a collision-free solution is obtained. Follow this simple idea, two progressive schemes can be proposed. First of all, a reader (say the i^{th} reader) is called a collided reader if its time slot assignment collides with at least one of its neighbors.

A. Heuristic and SA-Based Algorithms

Figure 6 and 7 show the steps of the progressive heuristic and SA-based algorithms. Their working principles are not complicated. All the collided readers are assigned with the new time slot $T + 1$ if the solution obtained for the case when the number of time slots is set to T is not collision-free. Then, the search is re-run with this new solution as the initial condition for the number of time slots is set to $T + 1$. The steps are repeated until a collision-free solution is obtained. If the solution obtained is again not collision-free, the collided readers are assigned with the time slot $T+2$ and so on until a collision-free allocation is obtained.

Progressive Simulated Annealing

S0 T is initialized to a small integer.
S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random slots allocation. $k = 1$.
S2 Random select a reader, say i. $k = k + 1$.
S3 If the reader has no collision to its neighbor, then goto S2.
S4 If the reader has collision to its neighbors, random generate a new $\tau_i^* \in \{1, 2, \cdots, T\}$.
S5 If the number of collision pairs between the reader and its neighbors reduces, $\tau_i = \tau_i^*$.
S6 If the number of collision pairs between the reader and its neighbors increases by Δ, generate a random number u from a uniform distribution in $[0, 1]$. Then,

$$\tau_i = \begin{cases} \tau_i^* & \text{if } u \leq \exp(-\Delta/\lambda(k)), \\ \tau_i & \text{if } u > \exp(-\Delta/\lambda(k)). \end{cases} \quad (11)$$

S7 Repeat steps S2 to S6 for $k \leq MaxRun$.
S8 Check if the solution obtained is collision-free.
S9 Assign all the collided readers with a number $T + 1$.
S10 Repeat S2 to S9 with $T = T + 1$, until the solution is collision-free.

Fig. 7. Progressive simulated annealing algorithm.

B. Colorwave

Colorwave algorithm is an extension of the Distributed Color Selection (DCS) algorithm. For the DCS algorithm, the maximum number of available time slots for assignment, i.e. T, is fixed. For Colorwave, this number is not fixed in advance. In Colorwave, each reader maintains a local T^i. Whenever a reader has collision, it selects randomly a new τ_i in $\{1, 2, \cdots, T^i\}$ as the time slot for interrogation. The value of T^i can either be up or down by one, depended on the successful rate of assignment. If collision persists for a number of trial assignments, the value of T^i will be incremented by one. On the other hand, if collision-free assignment has succeeded for many times, the value of T^i will be decremented by one. The value of T will be maintained as a global variable indicating the maximum value of T^i in the network, i.e.

$$T = \max_i \{T^i\}.$$

A summary of the Colorwave algorithm is listed in Figure 8.

For illustration, Figure 9 shows the first three steps of a simple update example. The order of update is assumed to be from the Reader-4 to the Reader-3 and then to the Reader-7. The values inside the bracket correspond to the time slot being assigned (τ_i) and the local maximum number of color (T^i). As the assignment of the Reader-4 has no collision, the reader simply set T^2 to 2. For Reader-3, it finds that there is no collision-free slot for selection as $T^3 = 3$. It then increases the value of T^3 by one (i.e. $T^3 = 4$) and sets τ_3 to 4. After that, it kicks this new assignment ($\tau_3 = 4$) to all its neighbors and sends the new T^3 to the central computer. For Reader-7, it finds its old assignment collides with its neighbor. It then randomly selects amongst slots 1, 2 and

Colorwave Algorithm

S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random slots allocation.

S2 For $i = 1, \cdots, N$, initialize $S_i = 0$ as the successful rate of assignment, $K_i = 0$ as the number of trials since last update of T^i.

S3 Random select a reader, say i. $K_i = K_i + 1$.

S4 If reader's slot assignment has no collision to its neighbor readers, $S_i = S_i + 1$.

  S4.1 If $S_i/K_i >$ DnSafe and $\tau_i < T^i$, set (i) $T^i = T^i - 1$ and (ii) $S_i = K_i = 0$.

  S4.2 If $S_i/K_i >$ DnSafe and $\tau_i = T^i$, search for all collision-free time slots for assignment. If the smallest numbered collision-free time slot, τ_i^* is of number smaller than (i) T^i, $T^i = \tau_i^*$, (ii) $\tau_i = \tau_i^*$ and (iii) $S_i = K_i = 0$.

S5 If reader's slot assignment has collision to its neighbors, $S_i = S_i$.

  S5.1 If $S_i/K_i <$ UpSafe, set (i) $T^i = T^i + 1$, (ii) $\tau_i = T^i + 1$ and (iii) $S_i = K_i = 0$.

S6 Update $T = \max\{T^i\}$.

S7 Repeat steps S2 to S6 until no more improvement can be made.

Fig. 8. Colorwave algorithm.

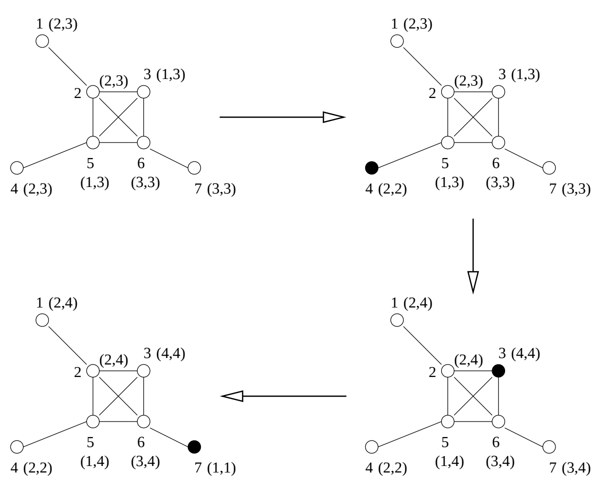

Fig. 9. Illustrative example for the Colorwave algorithm.

4 for the new assignment. Suppose it selects 1 for the new assignment (i.e. $\tau_7 = 1$) and this new assignment is of value smaller than its local maximum number of slot. Reader-7 then sets $T^7 = 1$ as the new local maximum color and kicks this new information to all its neighbors. We assume that the central computer broadcasts the readers the global maximum number of color (i.e. T) in the beginning of each cycle. This message also serves as a signal to synchronize all the readers.

V. Performance Evaluation

To compare the performance of the algorithms, we base on an artificial RFID reader network, Figure 10. Within an area of $100m \times 100m$, 250 readers are deployed uniformly

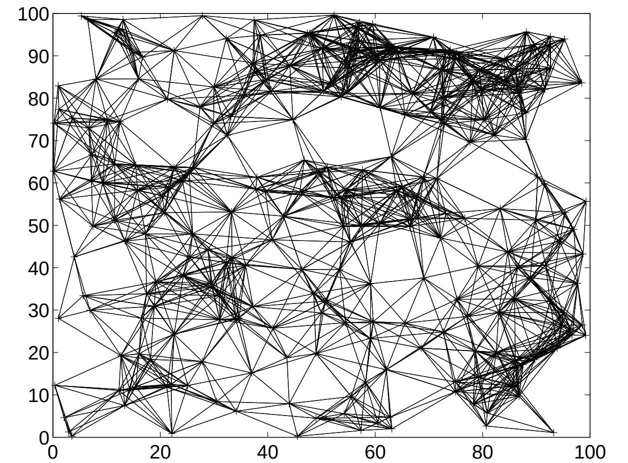

Fig. 10. 250 readers in random locations within an area of 100m×100m. Readers at the end of an edge are neighbors.

random. Two readers are neighborhood if their distance apart is less than or equal to 15m, i.e. $r \leq 15$. Each node in the figure corresponds to reader. The edges represent the neighborhood structure amongst the readers. The number of edges in the experimental network is 3830. In average, each reader has 15.32 neighbors.

A. Non-progressive Approach

For comparison, four non-progressive algorithms, including the heuristic-based algorithm, three SA-based algorithms and the DCS algorithm are running independently with the same initial condition on

$$T = 4, 5, 6, \cdots, 16.$$

For the heuristic-based algorithm, the total number of iterations is fixed to 2000. The total number of iterations (i.e. $MaxRun$) for SA-based algorithms is 50000. These numbers are determined after a few trial runs. The constants a, b, c and α in the SA-based algorithms are 0.01, 1, 2 and 0.99.

1) Collision pairs: Again, two neighbor readers are in collision if their time slots for interrogation are the same. Since the selection of a reader to update is random in each step, each experiment is repeated 5 times to avoid bias. The average number of collision pairs (defined in Equation (3)) being counted for different T are depicted in the following table.

The column with label RAND corresponds to the total number of collision pairs being counted right after the initial random allocation. The data shown in this column is for reference. One can see that all three simulated annealing algorithms perform similar to each other. Their solutions are slightly better than the heuristic algorithms. To have collision-free allocation, heuristic algorithm will need to have 16 slots in a cycle. While the SA algorithms employing constant temperature and the Geman-Geman rules require only 14 slots.

The DCS algorithm fails to generate a collision-free allocation, even for the number of available number slots is set

TABLE I
AVERAGE NUMBER OF COLLISION PAIRS BEING COUNTED FOR NON-PROGRESSIVE ALGORITHMS.

T	RAND	HEU	CT	GE	KP	DCS
4	497	255.4	229.6	230	228	498
5	384	170	141	143.2	143.6	403
6	309	112.8	91.8	93	92.8	333
7	262	81.6	59.2	61	60.2	272
8	229	52.2	37.2	37.2	38.4	224
9	192	35.4	23.2	23	22.6	216
10	183	21.4	11.8	12.4	12.2	195
11	157	12.2	6.6	6.8	6.8	175
12	146	7	3.4	3.4	3.6	158
13	144	3.6	1	1.2	1	149
14	121	1.8	0	0	0.2	122
15	127	0.4	0	0	0	115
16	115	0	0	0	0	97

RAND: Initial Random Allocation, HEU: Heuristic Algorithm
CT: Constant Temperature SA, GE: Geman-Geman Rule
KP: Kirkpatrick *et al* Rule, DCS: Distributed Color Selection.

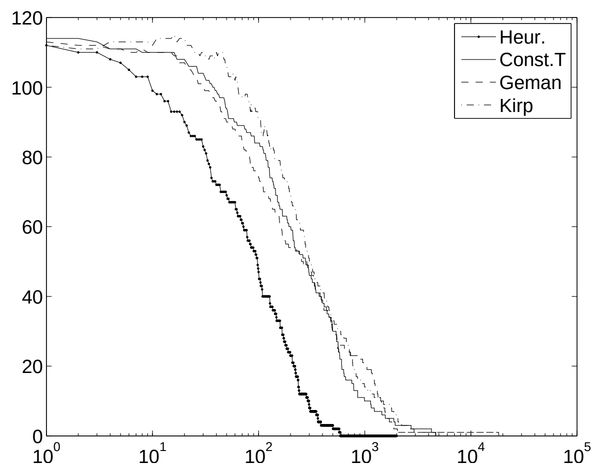

Fig. 11. Number of reader collision pairs against the number of iterations. The total number of slots is 16.

to 16. As a matter of fact, the DCS algorithm has also been applied with 20 time slots but the result is still negative. No collision-free allocation has been generated. It is believe that the failure is due to the fact that DCS does not take into the account of the neighborsassignment. Whenever a reader has encountered collision, it only selects randomly an alternative slot. In such case, even infeasible slots will be selected and collision can easily happen again. Therefore, the performance of DCS algorithm is so bad.

2) Convergence: In terms of convergence speed, heuristic algorithm converges much faster than the simulated annealing algorithms. It only takes 600 iterations to complete the search. All three SA algorithms will need to take around 20000 iterations. Figure 11 shows a typical convergence plot of all four algorithms for the case when T is 16. Note that the horizontal axis is in logarithm scale.

3) Slots distribution: As mentioned before, each reader once interrogation has been finished will have to send the data to the control computer in the next time slot. In this regards, a large number of readers assigned with the same time slot will lead to a large volume of data transmission

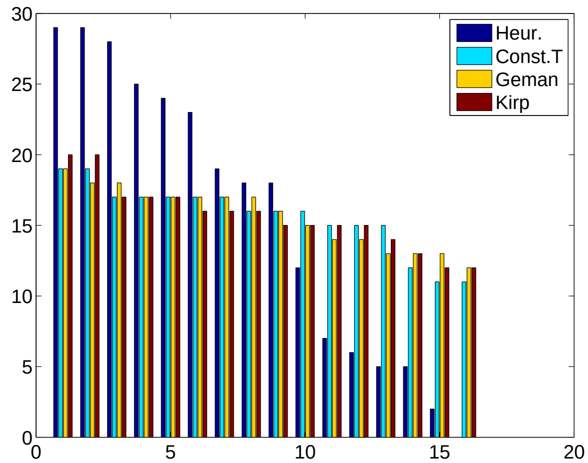

Fig. 12. Number of readers assigned to different time slots for non-progressive algorithms. The total number of slots is 16.

within a short time. Congestion and data lost might happen in return. To alleviate this effect, one method is to evenly allocate the time slots to the readers.

Figure 12 shows the slot distributions generated by the algorithms in one experiment. The result for the DCS algorithm is not shown in the figure, as it fails to generate a collision-free solution. The distributions are obtained by the following steps. Let $\tau_1, \tau_2, \cdots, \tau_N$ be the slot assignment. The number of readers being assigned with the k^{th} slot is counted. That is, for $k = 1, 2, \cdots, T$,

$$f_k = \sum_{i=1}^{N} A(i,k), \quad A(i,k) = \begin{cases} 1 & \text{if } \tau_i = k \\ 0 & \text{otherwise.} \end{cases}$$

Then, the values of f_k (for $k = 1, 2, \cdots, T$) are sorted in descending order, i.e.

$$f_{\pi_1} \geq f_{\pi_2} \geq \cdots \geq f_{\pi_T}.$$

The values of f_{π_k} against k are plotted in the figure.

It is clear that the time slot allocation obtained by the heuristic algorithm is not evenly distributed. It concentrates in a few time slots. The maximum number of readers assigned to a time slot is 29. For some time slots, fewer than 5 readers are assigned. On the other hand, the distributions for the SA-based algorithms are rather even. The number ranges from 11 to 20. Their corresponding entropies, given by

$$\text{Entropy} = -\sum_{k=1}^{T} \frac{f_{\pi_k}}{N} \log\left(\frac{f_{\pi_k}}{N}\right) \quad (12)$$

are evaluated and depicted in Table II. For reference, the entropy for 16 slots in which the readers are evenly assigned is 2.7726.

B. Progressive Approach

For the progressive approach, the simulations are carried out with the same reader network and the initial total number of available time slots (i.e. T) is set to 4. Other conditions are the same as for the non-progressive simulations. The number

TABLE II
ENTROPIES OF THE TIME SLOT DISTRIBUTIONS FOR NON-PROGRESSIVE ALGORITHMS.

Algo.	Entropy	No. of Slots
Heuristic	2.5249	16
SA CT	2.7605	16
SA GE	2.7633	16
SA KP	2.7622	16

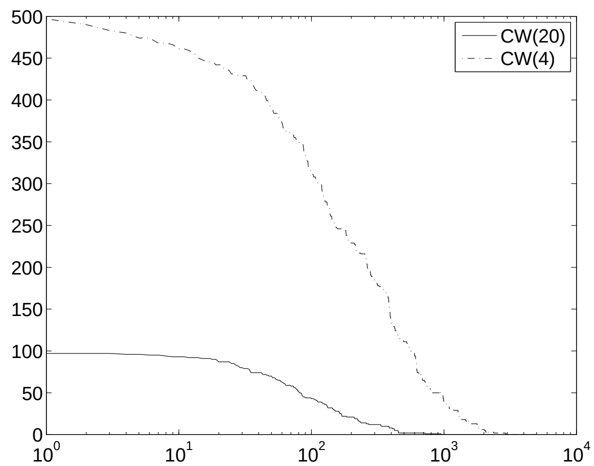

Fig. 13. The convergence behavior of Colorwave with T equals 4 and 20 .

of iterations for each round in the SA-based simulation is set to 10000. For the Colorwave algorithm, two experiments in which T equals to 4 and 20 have been conducted. In both simulations, the number of iterations is set to 5000 and their convergence behaviors are shown in Figure 13.

The total number of time slots required for a collision-free allocation is depicted in Table III. It is clear that the number of time slots required for a collision-free allocation is similar for all algorithms, from 14 to 16. Two progressive simulated annealing algorithms perform slightly better. While the progressive heuristic, the PSA-GE and the Colorwave algorithms need more time slots for collision-free allocation. The values of f_{π_k} against k are shown in Figure 14. Their entropies are depicted in Table IV. For reference, the entropies for 14 slots in which the readers are evenly assigned is 2.6391. While the entropies for 15 slots in which the readers are evenly assigned is 2.7081.

One can see that assignments obtained by progressive heuristic and the Colorwave algorithms are similar. Most readers are concentrated in a few time slots (2.4453 – 2.5040). While those obtained by progressive SA-based algorithms are more evenly distributed (2.6325 – 2.7020).

C. Hybrid Approach

Observe that the heuristic algorithm is of lower computational complexity than the others and the SA-based algorithms generate solution with evenly time slots distributions, a hybrid algorithm combining the progressive heuristic algorithm and the SA-based algorithm has been investigated. In the simulation, the progressive heuristic algorithm is

TABLE III
TOTAL NUMBER OF TIME SLOTS REQUIRED FOR A COLLISION-FREE ALLOCATION USING THE PROGRESSIVE ALGORITHMS AND COLORWAVE.

Algo.	Initial 4 slots
P. Heuristic	15-16
PSA CT	14-15
PSA GE	15-16
PSA KP	14-15
Colorwave	15-16

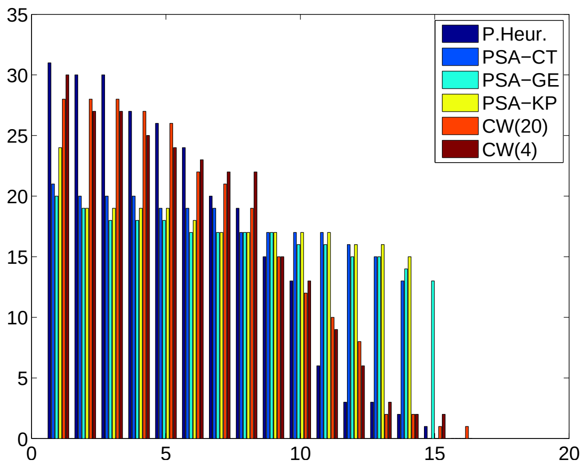

Fig. 14. Time slots distributions generated by different progressive algorithms.

applied as the first phase. Once a collision-free allocation has been obtained, it follows by the constant temperature SA algorithm. It is found that the number T obtained by this hybrid algorithm is similar to those obtained in the simulations above, which is in between 15 and 16. However, the distribution of the time slots is much better than the one obtained by the progressive heuristic algorithm alone, Figure 15. Once a collision-free allocation has been obtained by the progressive algorithm, the entropy of the time slot distribution is 2.4517. This value is increased to 2.7002 after the SA-based algorithm has been applied.

VI. SIMULATOR DESIGN

The analyses presented in the previous sections are based on the assumption that all readers are working under perfect condition. Clearly, it is too ideal to be practical. As a electronic device, reader can malfunction because of many reasons, such as component failure, radiation and interference. Once a reader is malfunction, its effect is unpredictable. For the worst situation, the reader can randomly re-assign a channel for interrogation. Reader collision can then be re-appeared. Therefore, one additional attribute that every practitioner will need to know is the reliability of the reader network. Simulator that can simulate the behaviors of the reader network operating under different reader allocation algorithms will definitely help. Figure 16 shows a prototypical reader network management system (RNMS) that we develop for this purpose.

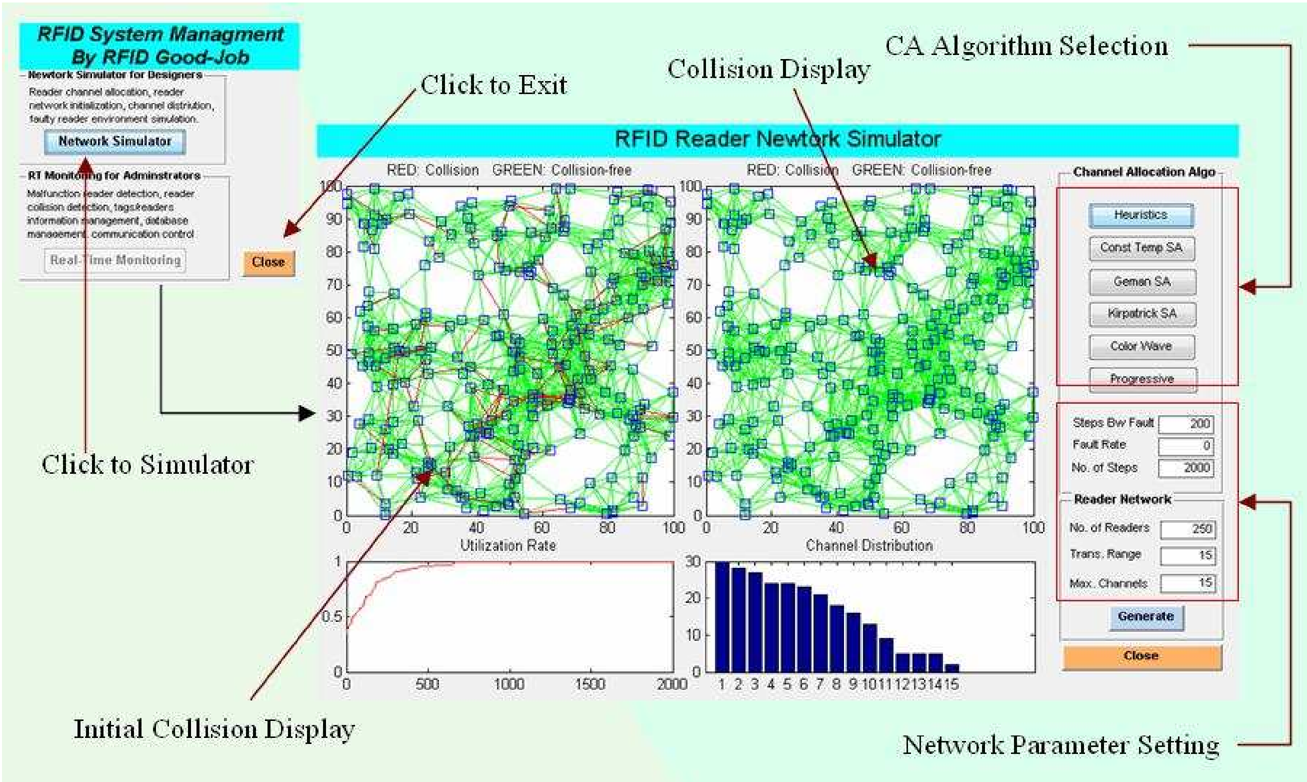

Fig. 16. Simulator for reader network management.

TABLE IV
ENTROPIES OF THE TIME SLOT DISTRIBUTIONS FOR THE PROGRESSIVE ALGORITHMS AND THE COLORWAVE (WITH INITIAL 20 SLOTS AND INITIAL 4 SLOTS).

Algo.	Entropy	No. of Slots
P. Heuristic	2.4453	15
PSA CT	2.6315	14
PSA GE	2.7020	15
PSA KP	2.6325	14
CW(20)	2.5040	16
CW(4)	2.4997	15

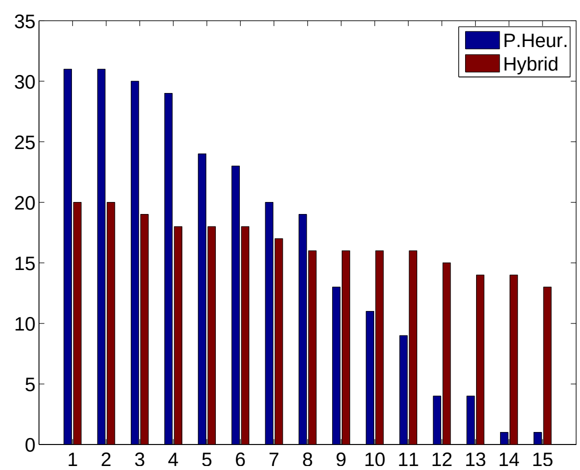

Fig. 15. Number of readers assigned to different time slots (Hybrid Approach).

A. Reader network management system

Basically, the RNMS is a system for system developer to analyze the performance of the reader allocation algorithms, and then design protocol for the system administrator to monitor the reader network, as shown in Figure 17. To do so, the RNMS is constituted by three modules : reader network simulator, real-time reader network monitoring module and I/O interface module, Figure 18. The reader network simulator is for running channel allocation simulations. The real-time reader network monitoring module is for the system administrator to monitor the utilization of the reader network, the network traffic and the functioning of the readers. The I/O interface module is responsible for data transfer amongst the RNMS, the physical reader network, the file system and the database system.

B. Simulator

For the simulator[1], it is developed by the MATLAB GUIDE and its program design is shown in Figure 19. This M-file consists of multiple MATLAB functions for network parameters setting, including (i) Number of Readers, (ii)

[1]The panel is designed by using MATLAB GUIDE together with the following program code. Reader who is interested in this programs can email to John Sum (email:pfsum@nchu.edu.tw) for dot-fig and dot-m files.

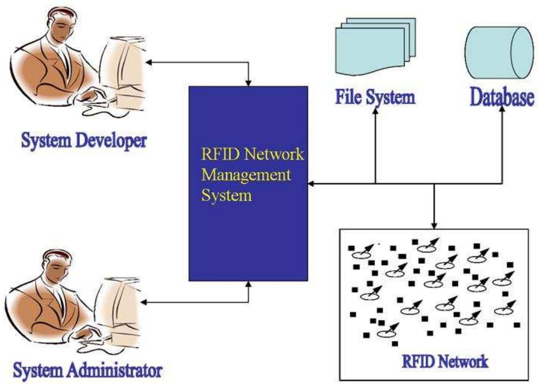

Fig. 17. Reader network management system.

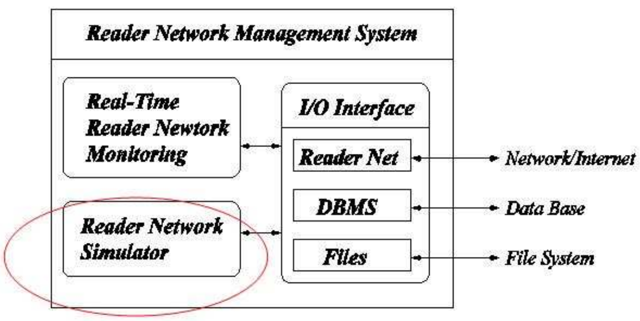

Fig. 18. Three modules in the reader network management system.

Transmission Range, (iii) Number of Channel, (iv) Number of Steps, (v) Reader Fault Rate and (vi) Channel Allocation Simulation.

Snapshot of the main menu program and the simulator are shown in Figure 16. Once the RNMS main menu has been invoked, it will pop up a small window with two options for click. One is for simulator and the other is for real-time monitoring, as shown at the top left hand corner in the Figure. To run simulation, click the *simulator* button. Then, a window for the simulator will appear, as shown on the right hand side in the Figure. Inside the simulator panel, there are four windows on the left and two blocks of buttons on the right.

The four windows on the left show the initial collision map, the collision maps while the channel allocation algorithm is running, the utilization rate of the network and the channel distribution. In the collision map, a small square corresponds to a reader. An edge connecting two readers indicates that these two readers are close to each other. If two neighbor readers use the same channel, collision exists. Then, the color of the corresponding edge will be in red. Otherwise, the color is in green. The utilization rate[2] is

[2]We take a simple management scenario. Once a reader has been identified to be collided, the reader will be set to idle. It will not perform interrogation. Until it has been assigned with a collision-free channel, it will re-start interrogation.

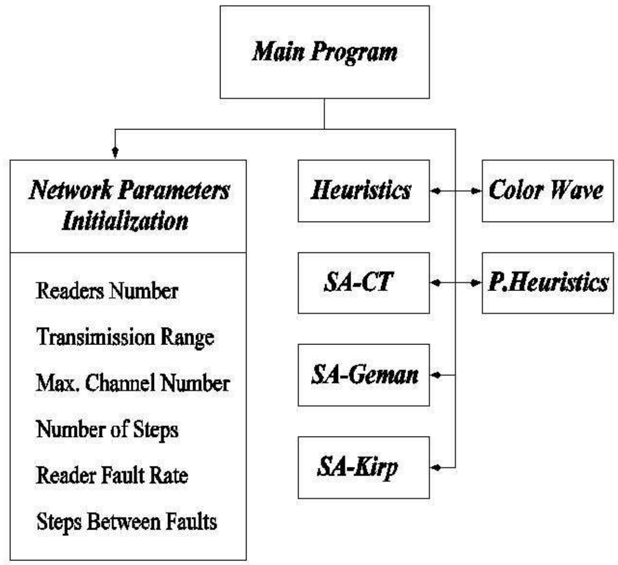

Fig. 19. Program design of the simulator module.

defined as follows :

$$\text{Utilization Rate} = 1 - \frac{\text{No. of Faulty Readers}}{\text{Total No. of Readers}}. \qquad (13)$$

On the right hand side, there are buttons for the options for channel allocation algorithms and the boxes for entering the network parameters.

In the example shown in Figure 16, the network parameters are set as below.

No. of readers	250
Reader fault rate	0
Trans. range	15 meter
Channels	15
Algorithm	Prog. heuristic
Steps between fault	0

The terrain in which the readers are deployed is defined as $100m \times 100m$ square, and the readers are deployed uniform randomly over the terrain. As the fault rate is zero, the algorithm can eventually search for a collision-free channel allocation for the readers.

Figure 20 shows an example in which the fault rate is non-zero. The network parameters are set as below.

No. of readers	250
Reader fault rate	0.05
Trans. range	15 meter
Channels	15
Algorithm	Prog. heuristic
Steps between fault	200

To simulate the fault, we make use of the following procedure. After every 200 steps, all the readers will assign a uniformly random number in $[0, 1]$. For readers with random number less or equal to 0.05, their channel numbers will be randomly re-assigned. We do that to simulate the worst case scenario. Clearly from Figure **??**, the utilization fluctuates about 0.8 to 1 after 1000 running steps.

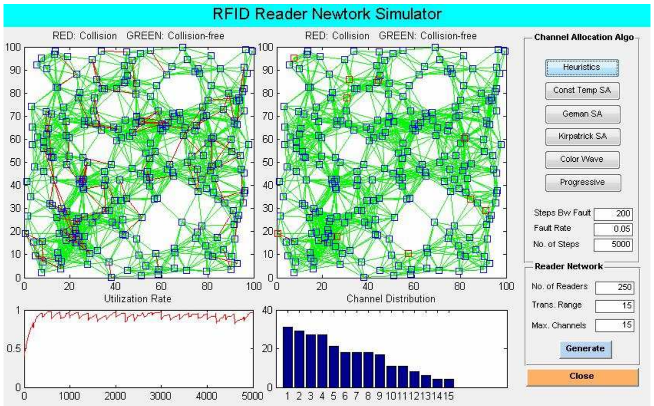

Fig. 20. Simulation of a reader network with faulty readers.

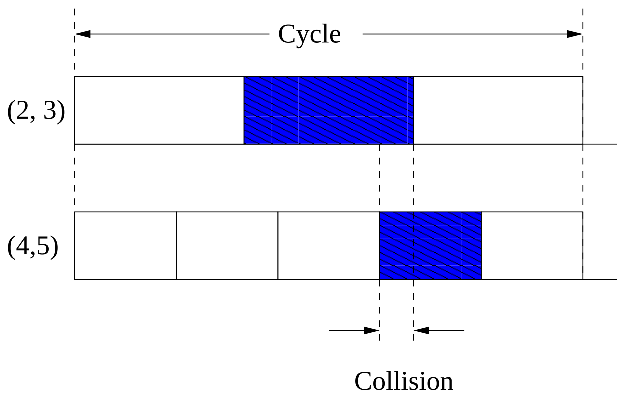

Fig. 21. Reader collision due to different values of local maximum color.

VII. Discussion

A. Original Colorwave

In the original Colorwave algorithm [20], each reader sends a 'kick' including both its current color and its local maximum color (i.e. τ_i and T^i) to its neighbor whenever there is an update on either information. A reader once has received a kick from its neighbor will have to select a new color in order not to overlap with color in the kick. A central computer is assumed to be a server synchronizing the start of each cycle. Through the *'kick' wave*, all readers might obtain the global maximum number of time slot T after a period of time. However, this might cause reader collision effect since there is no guarantee all readers have the information of T, see Figure 21. As the partitioning of a cycle is determined locally by the value T^i in a reader, neighbor readers with different T^i will have different partitioning. As a results, two neighbor readers with different τ_i (collision-free) might still cause collision in the end.

In our setting, we assume that the central computer is not simply a server synchronizing the start of each cycle. But it is also responsible to broadcast the T value to all the readers in the beginning of a cycle. The readers can thus refer to the same number to partition the time slots.

Besides, the procedure depicted in [20] has not stated the mechanism how a color should be selected if the local maximum color of a reader has been decremented. We explicitly define these steps by introducing two values for a reader to maintain, the local maximum color (T^i) and the global maximum color (T). A simple rule is proposed to supplement the decrement update of local maximum color : If $\tau_i = T^i$ and there is no collision-free time slot can be found from $\{1, 2, \cdots, T^i - 1\}$, the value of T^i will not be changed. If there is a collision-free time slot (say τ_i^*) with number smaller or equal to $T^i - 1$, both the values of τ_i and T^i will be assigned to τ_i^*.

TABLE V
Comparative analysis with other methods.

Algo.	Slots	Network	Node Degree
DCS	10	Regular Grid	4
CW	4 to 4.5	Random Net	4 (Average)
CW	8	Random Net	8 (Average)
Heur.	15 to 16	Random Net	15 (Average)
SA	14 to 15	Random Net	15 (Average)
CW	15 to 16	Random Net	15 (Average)

B. Number of time slots

In our simulations on a random network of 250 readers with the average number of neighbor is 15, the best solution is 14 time slots. Waldrop *et al* have reported in [20] that the number is in between 4 to 4.5 when Colorwave algorithm is applied in a random network with the average number of neighbors is 4. When the algorithm is applied in a random network with the average number of neighbors is 8, the number of time slots is 8. Applying DCS algorithm for a square grid network with 4 neighbors, the number of time slots is 10. Table VII-B compares the results obtained in their paper and in this paper. Two comments can be made from these results. First, the result for square grid seems to be far from optimal. It can be shown that [10], a collision-free allocation can easily be obtained even if the number of time slots is 2. Second, it seems that the number of time slots (color) obtained by Colorwave for a collision-free allocation is correlated to the average number of neighbors. Further investigation is worthwhile noticed.

C. Other Reader Collision Problems

The problem addressed in this paper is to assign each reader one dedicated time slot for interrogation and the transmission rate of each reader is 100%. However, it is not the only problem formulation for reader collision problem. At least three other formulations have been defined in the literature.

In [20], [21], the authors applied the Colorwave algorithm for a problem similar to the one presented in this paper except that the transmission rate of a reader is not 100% full. Lin and Lin in [13] studied the application of simulation annealing method in a problem that a reader can have multiple time slots in a cycle for interrogation. For a reader that can use different frequency channels for interrogation, Ho *et al* extended the formulation to a frequency-time space allocation problem and developed a Q-Learning based algorithm to solve it [9].

As the assumption of their formulations are different from us, except the problem addressed in [20] comparative analysis amongst the corresponding algorithms and the algorithms presented in this paper has not been taken.

VIII. Conclusion

In this paper, a number of algorithms for solving reader collision problem have been elucidated. They include two heuristic-based algorithms, six SA-based algorithms, the DCS algorithm, the Colorwave algorithm and a hybrid heuristic-SA algorithm. In which, five of them are non-progressive type that the maximum number of time slots for interrogation is predefined in their algorithms. Six of them are progressive type that the number of time slots for interrogation is determined automatically. Based on the simulation that 250 readers are randomly deployed and the total number of neighbor pairs is 3830, the number of time slots determined by the algorithms for collision-free interrogation is in between 14 and 16. Non-progressive heuristic algorithm needs 15 to 16 time slots. Non-progressive SA-based algorithms need 14 to 16. However, DCS algorithm fails to generate a collision-free allocation even though the number of time slots is set to 20. Furthermore, it is revealed that the progressive heuristic-based, progressive SA-based and the Colorwave algorithms produce similar results in terms of time slots number. All of them determine 14 to 16 time slots. In terms of convergence speed, the progressive heuristic-based and the Colorwave algorithms converge much faster than the progressive SA-based algorithms. It is also found that all non-progressive and progressive SA-based algorithms are able to generate evenly distributed time slot allocation. It means that the assignment of readers does not concentrate in a few time slots. If a reader has to finish data-upload by the end of next time slot, the bandwidth required for the central computer to collect data from the readers could be reduced. Apart from the comparative study, a few issues related to our work have been discussed in this paper to highlight the differences between our works and those reported in other papers. Finally, we have described a conceptual design for real-time reader network management system and the design of a simulator for investigating the performance of different channel allocation algorithms if faulty readers exist. While many algorithms have been developed to solve reader collision problems, our study focuses on dense reader network. The results reported here should have provided the RFID system designers a richer understanding on the performance of the reader collision avoidance (RCA) algorithms, and paved the way for the development of reader network management systems for practical use.

References

[1] C. Abraham, V. Ahuja, A.K. Ghosh and P. Pakanati, Inventory Management using Passive RFID Tags: A Survey, Department of Computer Science, the University of Texas at Dallas, Richardson, Texas, October, 2002.

[2] AutoID Center, 13.56 MHz ISM Band Class 1 Radio Frequency Identification Tag Interference Specification: Candidate recommendation, Version 1.0.0, Technical Report MIT-AUTOID-WH-002, MIT Auto ID Center, 2003.

[3] S. M. Birari, S. Iyer, Mitigating the reader collision problem in RFID networks with mobile readers, *IEEE Conference on Communication*, Vol. 1, page: 6, 16-18 Nov. 2005

[4] C. Boyd, Social Networking for Dogs, in MIT Technology Review, May 03, 2007. (http://www.technologyreview.com /Infotech/18628/)

[5] D.W. Engels, S.E. Sarma, The reader collision problem, in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vol. 3, pp.6, 2002.

[6] EPC Global, EPC Generation 1 Tag Data Standards Version 1.1 Rev.1.27, EPC Global Technical Report, May 2005.

[7] Etsi, ETSI EN 302 208-1 V1.1.1 September 2004. [Online] www.etsi.org.

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

[9] J. Ho, Daniel W. Engels, and Sanjay E. Sarma, HiQ: A Hierarchical Q-Learning Algorithm to Solve the Reader Collision Problem, *SAINTW'06*, 2006.

[10] Y.H. Huang, S.W. Hung and J. Sum, Simple algorithms for the assignment problem in structure reader networks, to appear in *Procceedings of Internation Conference in Advanced Information Technologies*, Chaoyang University of Technology, April 25-26 2008, Taiwan.

[11] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, Optimization by simulated annealing, *Science*, vol. 220, no. 4598, pp. 671V680, 1983.

[12] J.K.Lee,S.K.Das and K.Kim, Analysis of RFID Anti-Collision Algorithms using Smart Antennas, *SenSys'04*, Baltimore, Maryland, USA, pp. 265-266, 2004.

[13] C.F. Lin and F.Y.S. Lin, 2007, A Simulated Annealing Algorithm for RFID Reader Networks, *Proceedings IEEE WCNC'07*, pp.1669 - 1672, March 2007.

[14] C.H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, 2 Edition, Dover, 1998.

[15] T. S. Rappaport, *Wireless Communications- Principles and Practice*, 2nd Edition, Prentice Hall, 2002.

[16] S. Sarma, D. Brock, and D. Engels, Radio Frequency Identification and the Electronic Product Code, *IEEE Micro*, vol. 21, no.6, pp.50-54, 2001.

[17] 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.

[18] W. Stallings, *Wireless Communications & Networks*, Second Edition, 2005.

[19] W. Stallings, R. Want, An Introduction to RFID Technology, *IEEE Pervasive Computing*, vol.5, no.1, pp. 25-33, Jan.-March 2006

[20] J. Waldrop, D.W. Engels, S.E. Sarma, Colorwave: an anticollision algorithm for the reader collision problem, *IEEE Wireless Communications and Networking Conference*, pp.1206-1210, March, 2003.

[21] J. Waldrop, D.W. Engels, S.E. Sarma, Colorwave: A MAC for RFID reader networks, *IEEE Wireless Communications and Networking Conference*, pp. 1701-1704, 2003.

[22] R. Want. *RFID Explained: A Primer on Radio Frequency Identification technologies*, Morgan and Claypool, 2007

[23] R. Weinstein, J. Landt, The history of RFID, *IEEE Potentials*, Vol.24, no.4, pp. 8-11, Oct-Nov 2005

[24] T. Zhijun, and H. Yigang, Research of Multi-access and Anti-collision Protocols in RFID Systems, *Proc. IEEE Anti-counterfeiting, Security, Identification* pp.377-380, 16-18, April 2007.