ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 5, May 2015

MULTIDIMENSIONAL GROUPED BACKTRACKING BINARY SEARCH FOR REDUCED RFID TAG COLLISIONS

Shilpa Bhatt, Deepak Dhadwal

***Abstract*— Radio Frequency Identification (RFID) technology has been attracting visibility for both suppliers and retailers. Though it is working successfully in current market, but, to widen its scope a lot of improvement in speed is needed. The collision in RFID tags is a major challenge for its efficiency. This paper considers the problem of reducing collisions for RFID tags, and, at the same time proposes an algorithm suited for multiple processing environments. The algorithm divides the tags in range into smaller groups, and, within each group it divides the tags into subgroups making a multidimensional structure. It reduces the concentration of tags within a group. In this way, the chances of collision will be reduced within a lowest level of subgroup, and, at the same time it creates an ideal condition for implementing a multiple receiver support. The paper is limited to the scope of algorithm, and, will not cover the hardware aspect of system. The algorithm is implemented using MATLAB and the results are also plotted using the same software.**

***Index Terms*— Collision Avoidance, Multidimensional Grouping, Multiple Reader, RFID Tags**

I. INTRODUCTION

Recent technology has made human so dependent on them that automation is the key parameter adopted almost everywhere. A major example of this can be seen in hypermarkets, industries, libraries, l**aboratories** and many more where a technology named as automatic identification is used on a large scale. There was a need for fast processing, data accuracy (minimizing human errors), efficient transmission of data, and making the entire data entry/collection process more efficient. Barcode technology is one of the technologies used extensively but it requires close sight reading, manual assistance where RFID has came out as a boom for retailers, as well as, manufacturers.

The first RFID application was used in Second World War by the British .RFID identify physical objects with non light of sight & automatic reading. Radio Frequency Identification (RFID), commonly known as electronic tag, is a non-contact automatic identification technology, which taking the RF signal as the transfer medium of information and energy, so as to complete the information exchange with the measured objects[1]

RFID (Radio Frequency Identification) is a method of identifying unique items using radio waves. An RFID system is always made up of two components: The transponder located on the object to be identified, and, the interrogator or reader, which, depending upon the design and the technology used, may be a read or write/read [2]

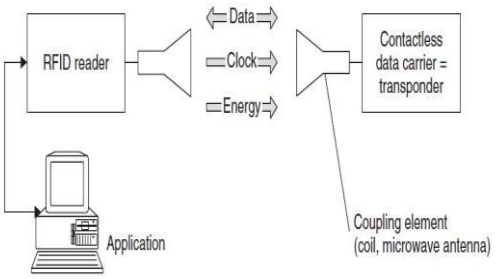

Figure 1: A broad view of RFID System

A reader typically contains a radio frequency module (transmitter and receiver), a control unit and a coupling element to the transponder. In addition, many readers are fitted with an additional interface (RS 232, RS 485, etc.) to enable them to forward the data received to another system (PC, robot control system, etc.). The transponder, which represents the actual data-carrying device of an RFID system, normally consists of a coupling element and an electronic microchip. When the transponder, which does not usually possess its own voltage supply (battery), is not within the interrogation zone of a reader is totally passive. The transponder is only activated when it is within the interrogation zone of a reader. The power required to activate the transponder is supplied to the transponder through the coupling unit (contactless), as are the timing pulse and data [1]. RFID tags fall into two categories: Active and Passive (Memory less). An active tag is equipped with a battery and relatively expensive circuit. Thus it can perform complicated processing, but its size is large. On the other hand, a passive tag is cheap, small, and powered by the radio from reader, but it has no storage for processing history, except for the ID and some embedded information. In usual active tags are used for large goods such as container in port or cars in parking area, while passive tags for small items such as goods in a retail warehouse [2].

When multiple tags respond to single reader at same time, and at same frequency, the radio signal transmitted by all interferes with each other. This interference is referred to as a 'collision', resulting into non-reliable identification. So it

Manuscript received May 11, 2012.

***Shilpa Bhatt**, Mtech ECE, Maharishi Markandeshwar University, Sadopur, India*

***Deepak Dhadwal**, Associate Professor (ECE), Maharishi Markandeshwar University, Sadopur, India*

was necessary to eliminate the selection of wrong tag, NRZ encoding was used, which results in an invalid state after collision. For improvisation, a number of algorithms were developed but all had some limitations.

It is an important problem to enhance the tag identification efficiency. When the number of tags is large, for the conventional RFID anti-collision algorithm the number of slots required to read the tags increases exponentially as the number of tags does[3].

II. Related Work

For handling with multiple access in radio frequency transmissions, traditionally space division multiple access (SDMA) was used. In this an 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 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 transponder (adaptive SDMA). So various transponders can be differentiated by their angular position in the interrogation zone of the reader.1 Phased array antennas are used as electronically controlled directional antennas. These consist of several dipole antennas, and therefore adaptive SDMA can only be used for RFID applications at frequencies above 850MHz (typical 2.45 GHz) as a result of the size of the antennas. This was followed by frequency domain multiple access (FDMA) methodology. The term frequency domain multiple access relates to techniques in which several transmission channels on various carrier frequencies are simultaneously available to the communication participants [2].

An Aloha System for multiple accesses in radio communications was proposed in which Time Division Multiple Access principal was used. Due to the fact that from total number of consoles in the systems, only a fraction of the will be active for any given time and because of burst nature of the data, inefficiencies occurred in wired communication .This problem may be partly alleviated by a system of central control and channel assignment. Each tag transmits packets to the reader in the same manner. Only when packet is received without error, and then only acknowledged by the reader. Once the packet is transmitted from a tag, it waits for a certain amount of time for the acknowledgement from reader machine. If it does not get any signal within a certain time, it retransmits the packet. This process goes on till all the packets are successfully received by reader or the machine itself terminates the connection. There are two different errors which could be occurred and interrupt successful transmission, i.e. random noise errors (electronic noise in the circuit)and errors caused by interference caused due to other tag(This error comes when number of tags transmitting information are available in large number). These errors limit the performance of system. The problem in this approach was that it might lead to tag starvation problem, i.e., a tag might not get detected for a very long time[4].

A TDMA type tree algorithm was developed for packet broadcast channels. In which each source corresponds to a leaf on a binary tree. When the sources are infinite, then the tree extends to infinity. This is known as binary addressing scheme. For example, each source has a four-bit binary address. Also, let Tx and Ty, be two rooted sub trees, and assume that no collisions have occurred up to the beginning of the present pair of slots. Then the binary tree algorithm can be stated as follows. Set Tx = T10; Ty = T11. mTransmit all the packets from sources in Tx in the first slot of the present pair of slots, and transmit all the packets from sources in Ty, in the second slot (As has been pointed out above, a source in the Poisson source model can have at most one packet). If any collisions occur in the preceding step, then Until these collisions are resolved, no new packets are transmitted; Resolve the first collision (if any) before resolving the second (if any). A collision in Tx (or Ty) is resolved by dividing Tx (or Ty) into two halves (say A and B), setting Tx = A, Ty = B, and then repeating steps 2 and 3.

Although tree-based protocols do not cause tag starvation, they have relatively long identification delay due to the splitting procedure starting from one set including all tags. So Adaptive Binary Splitting for Efficient RFID Tag Anti-Collision was proposed by Jihoon[5].

When a tag transmits ID, it awaits for the feedback in which reception indicates signals, namely, idle (no tag signal), readable (only one tag signal) or collision (multiple signals). Tag ti decides the timeslot for the transmission with a progressed-slot counter, PC, and an allocated-slot counter, AC(i). PC means the number of tags recognized by the reader in the ongoing process. At the beginning of the process, PC is initialized with 0. AC(i) signifies the timeslot for ti's transmission, i.e., ti transmits when PC = AC(i). The tags, which have the same value of the allocated-slot counter, form a set. Tag collision occurs if a set include multiple tags. According to the feedback, tags act as follows.

Readable: Tags add 1 to PC. Idle: Tag ti decreases AC(i) by 1 to pull the schedule of the transmission if PC < AC(i). Collision: To split a set of tags, tag ti generates a random binary number and adds it to AC(i) if PC = AC(i). Note that colliding tag ti has AC(i) equal to PC. Since PC is not changed, the first subset (tags which generate 0) retransmits at the following timeslot and the second subset (tags which generate 1) retransmits after the first subset is recognized. To prevent the second subset and another set of tags, which have already had the allocated slot counter of AC(i) + 1, from integrating, tag tj adds 1 to AC(j) if PC < AC(j). At the end of the process, every tag has a unique allocated slot counter. Therefore, maintaining the allocated-slot counter at the boundary of two consecutive processes enables the splitting procedure to start from multiple sets of tags. For terminating the identification process at once after identifying all tags, the reader acts as the tag which has the largest allocated-slot counter. Reader r concludes all tags to have been recognized and terminates the process if PC = RC(r) (RC(r) is reader r's allocated-slot counter). The time taken in this for allotting unique AC(i) to each transponder based on a selected random no is also very exhaustive method. So this also consumes considerable amount of time [6].

In Binary tree algorithm it was observed that, at each iteration, the nodes were traversed in same order as in previous. So A Novel Anti-collision Backtracking Algorithm based on Binary-tree Search was developed. For its implementation, the tags were added to the dormant depth counter need, also the reader need to add the collision bit counter and jump flag bit. This depth counter can implement the backtracking. The collision bit counter can only judge whether a collision bit occur. It defines the three stages: Activation state; Dormant state; Quiet state. The dormant

state and the dormant depth counter work together to complete the collision record and reduce the impact of information search.

However this was a significant improvement over the conventional binary tree algorithm, and, it exponentially saved the time for tag identification, still it did not look at the possibility of reducing tag collisions. [7]

Research of Matrix-based Grouping Method on Anti-collision Algorithm for RFID Tag Identification" .In this concept was introduced where grouping was done among the tags. Grouping is done to reduce the search time to identifying tags. In result, it continuously identifies the tags by performing multiple searches. The grouping of tags should be reasonable according to the current number of tags. The key is how to reasonably group the tags. So, a method was developed in which the value is calculated by doing square root according to the number of tags. A row value of matrix is then selected with the smallest integer which was greater than the value of square root, that is: Line=√N. The matrix can be calculated as: Column=N/√N .Hence, collisions were reduced by this algorithm [8].

III. Multidimensional Grouped Backtracking Binary Search Algorithm

This is based on the fact that larger the number of tags in a system, higher is the probability of collisions. So the objective is to divide these tags into smaller groups so that the probability of collision is reduced.

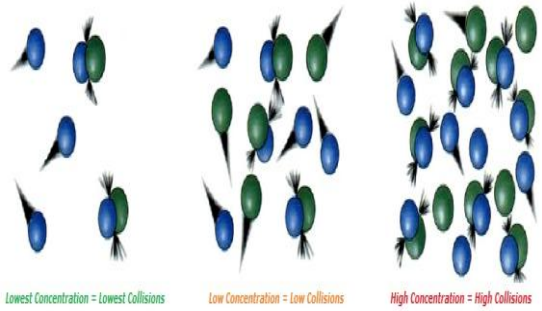

Figure 2: Diagram depicting consequences of concentration change

This algorithm is an extension to "Matrix-based Grouping Method on Anti-collision Algorithm for RFID Tag Identification". So the algorithm will use same methodology for grouping of N tags into √N groups as under.

$$\begin{bmatrix} tag1 & tag2 & tag3 & tag4 & tag5 & tag6 \\ tag7 & tag8 & tag9 & tag10 & tag11 & tag12 \\ tag13 & tag14 & tag15 & tag16 & tag17 & tag18 \\ tag19 & tag20 & tag21 & tag22 & tag23 & tag24 \end{bmatrix}$$

Figure 3: Matrix depicting 2D grouping

Further to this grouping, the reader will send a second level grouping signal to all the tags in range where tags will group themselves into subgroups. Although, the level of sub-division of tags, and, the logic for doing it can be customized based on the nature and properties of application, but for simplicity purpose it may subdivided into two sub-groups, say A & B, as under.

Say each tag is of k bits.

A = {x: x contains at least k/3 0 valued bits},

B = {x: x contains less than k/3 0 valued bits}

So a multidimensional grouping will be done as under.

$$\begin{bmatrix} \{tag1 \\ \{tag2 \end{bmatrix} \begin{bmatrix} \{tag3 \\ \{tag4 \end{bmatrix} \begin{bmatrix} \{tag5 \\ \{tag6 \end{bmatrix}$$
$$\begin{bmatrix} \{tag7 \\ \{tag8 \end{bmatrix} \begin{bmatrix} \{tag9 \\ \{tag10 \end{bmatrix} \begin{bmatrix} \{tag11 \\ \{tag12 \end{bmatrix}$$
$$\begin{bmatrix} \{tag13 \\ \{tag14 \end{bmatrix} \begin{bmatrix} \{tag15 \\ \{tag16 \end{bmatrix} \begin{bmatrix} \{tag17 \\ \{tag18 \end{bmatrix}$$
$$\begin{bmatrix} \{tag19 \\ \{tag20 \end{bmatrix} \begin{bmatrix} \{tag21 \\ \{tag22 \end{bmatrix} \begin{bmatrix} \{tag23 \\ \{tag24 \end{bmatrix}$$

Figure 4: Matrix depicting 3D grouping

So with this enhancement we can have multiple readers in a RFID system, which would work at different frequencies, but, for the same database. This difference in frequency will prevent collisions, and, each of the reader will have its own white list of tags. In above case we can have two readers, R1, which will accept all the tags that belong to set A, and, reader R2, which will accept the tags that belong to set B.

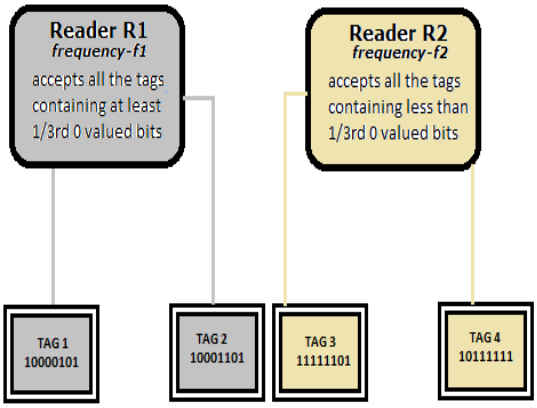

Figure 4: Diagram proposing Multi-Reader System

So if there are N tags in the scope of the reader *rf* field. So the number of nodes is "2N- 1" in the Binary Tree that is formed by N tags, and the average depth of leaf nodes is:

Sum = [Log2 N] + 1

Therefore, the number of the average search times is [Log2 N] + 1 to identify Tag-N from N tags. The time complexity is O (N Log2 N) after all of N tags are identified. It can be seen that the number of the average search times is becoming large when the value of N increases. To identify all of N tags by using the basic BST algorithm, the total number of identification is:

$$SumBST = \sum_{k=0}^{n} (\log_2 k + 1)$$

The proposed algorithm will divide the transponders in range into one group, and within each group it will further divide the group into some sub groups. As now the transponder will communicate group wise, it results in reduced parallel transmission thereby reducing the collisions. So as a result the time taken for collision resolution is also reduced. So the total time taken may be evaluated as under.

1) Let say the no of groups are l, and, within each groups

there are m elements. So total time for each group may be calculated as

$$SumG = \sum_{k=0}^{m}(\log_2 k + 1)$$

2) And the overall time may be calculated as

$$SumALL = \sum_{i=0}^{l}(\sum_{k=0}^{m}(\log_2 k + 1))$$

However if we look at the theoretical analysis of this, it is no better than the BST algorithm. But it is already concluded in [8] that because of reduced collisions the time taken for identifying each transponder is reduced dramatically. Further to this, because of multiple receiver support enabled in the system as a result of multidimensional grouping, the time required for tag identification is reduced.

IV. IMPLEMENTATION

For implementation of Multidimensional Grouping Based Binary Search Algorithm, oops concept is used within MATLAB for better code management. So, for each hardware, i.e. transmitter and receiver, there is a corresponding class in program. As all the operations, viz. data receiving, extracting most significant bit, maintaining depth counter for each tag, are executed from same machine, that too sequentially, so the time taken for overall execution is much larger than the actual time it would take. For implementation, an excel file is used as input, which when read is interpreted as tag collision. The NRZ code is then artificially derived using various bitwise operations. Based on this code, the most significant collision bit is recognized, and, the depth counter for all those tags having their bit value '0', or, already having their depth counter greater than zero, is incremented. The process continues till all, but one, tags are masked. The depth counter is then decremented for all, and, the process is iterated till all the tags are read. So the algorithm is divided into many sub algorithms for simplicity.

The Object contains following data structures

i. tag_matrix_orig- 1D array of original tag matrix to be processed
ii. tag_matrix_curr- 1D array for tag matrix being processed in current iteration
iii. tag_matrix_depth- 1D array for maintaining corresponding depth counter of each tag
iv. tag_matrix_temp- 1D array for temporary use in each iteration

a. Initialize is the first algorithm that is used for reading the tags from excel and dividing it into groups and subgroups with the help of depth matrix.

Inputs:

filename – absolute path of excel file containing tag ids

Step 1: Start
Step 2: Set tag_matrix_orig := xlsread(filename)
Step 3: Initialize tag_matrix_depth array with size equal to that of tag_matrix_orig, and, all the values set to zero
Step 4: rows := floor of (square root of (size of tag_matrix_curr))
Step 5: col := floor of ((size of tag_matrix_curr)/rows)Step 6: Initialize status := false, count := -1, i :=1
Step 7: Initialize no_one := floor of(no of bits in tag_matrix_curr(1) / 3)
Step 8: Repeat through Step 16 till i<rows
Step 9: if status = true then
increment count by 2
else
increment count by 1
Step 10: Set j:=1, status :=false
Step 11: Repeat through step 15 till j<col
Step 12 Set index := ((i-1)*col) + j;
Step 13: Set NumberOfOnes := number of ones in tag_matrix_orig(index)
Step 14: if NumberOfOnes>no_one then
tag_matrix_depth(index) := count;
else
tag_matrix_depth(index) := count+1;
status := true;
Step 15: Increment j by 1
Step 16: increment i by 1
Step 17: Stop

b. ReadAll is an algorithm at the top most abstract level. It calls the underlying algorithms till all the tags are read.

Step 1: Start
Step 2: set tag_matrix_curr := tag_matrix_orig
Step 3: repeat through Step 4 till size(tag_matrix_curr) > 0
Step 4: Call getNextIterateMatrix(tag_matrix_curr)
Step 5: Stop

c. GetNextIterateMatrix is an algorithm that maintains the depth counter and with the help of another algorithm gets the resolved tag in each iteration

Input:

tag_matrix_curr- 1D array of all the tags out of which one is to be selected

tag_matrix_depth- corresponding current depth counter for each tag

Output:

tag_matrix_curr- 1D array of all the tags but for the one selected in current iteration

tag_matrix_depth- corresponding updated depth counter for each tag

Step 1: Start
Step 2: increment all non zero values in tag_matrix_depth by 1
Step 3: selected_tag := getResolvedTag(tag_matrix_curr)
Step 4: Initialize temp := zeros(0,1), temp_depth := zeros(0,1), is_found :=0
Step 5: Repeat through Step 7 till is_found<1 and size(tag_matrix_curr)>1
Step 6: Set i:=1
Step 7: Repeat through Step 10 till i<size(tag_matrix_curr)
Step 8: if tag_matrix_curr(i) = selected_tag then goto Step 10
Step 9: temp(size(temp)+1) := tag_matrix_curr(i),
temp_depth(size(temp_depth)+1) := tag_matrix_depth(i) – 1
If temp_depth(size(temp_depth)+1) = 0 then set is_found:=1
Step 10: Increment I by 1
Step 11: Set tag_matrix_depth := temp_depth, tag_matrix_curr := temp
Step 12: Stop

d. GetResolvedTag algorithm iteratively calls another algorithm to generate depth counter values on the basis of collision till there is only one tag left with depth counter value 0

Input
tag_matrix_curr- 1D array of all the tags out of which one is to be selected
tag_matrix_depth- corresponding current depth counter for each tag
Output:
tag_matrix_temp – all the selected tags on the basis of binary search algorithm
tag_matrix_depth- corresponding updated depth counter for each tag

Step 1: Start
Step 2: tag_matrix_temp := generateDepthMatrix(tag_matrix_temp)
Step 3: Repeat through step 4 till size(tag_matrix_temp) >1
Step 4: tag_matrix_temp := generateDepthMatrix(tag_matrix_temp)
Step 5: Stop

e. GenerateDepthMatrix is an algorithm that takes the most significant collided bit from another algorithm and on basis of it selects the tags with bit value '1', and increments the depth counter for the tags with bit value '0'

Input
tag_matrix_temp- 1D array of all the tags out of which one is to be selected
Output:
tag_matrix_temp– all the selected tags on the basis of binary search

Step 1: Start
Step 2: Initialize bit_pos := getMostSignificantActiveBitPosition(tag_matrix_temp)
Step 3: Initialize i:=1, temp := zeros(0,0)
Step 4: Repeat through Step 5 till i< size(tag_matrix_temp)
Step 5: if bitget(tag_matrix_temp(i), bit_pos) = 1 then
temp(size(temp)+1) := tag_matrix_temp(i)
else
tag_matrix_depth(i) := obj.tag_matrix_depth(i) + 1
Step 6: tag_matrix_temp := temp
Step 7: Stop

f. GetMostSignificantActiveBitPosition is an algorithm that computes the position of most significant collided bit

Input
tag_matrix_temp- 1D array of all the tags for which most significant collided bit is to be calculated
Output:
bit_pos– position of most significant collided bit

Step 1: Start
Step 2: initialize and_output with ANDed result of all tag IDs
Step 3: initialize or_output with ORed result of all tag IDs
Step 4: initialize comp_and_output with complement of and_output
Step 5: temp := dec2bin(bitand(comp_and_output, or_output))
Step 6: bit_pos := length(temp)
Step 7: Stop

V. RESULTS

The configuration of the system used for obtaining the results is – Windows 7, 2 GB RAM, 1.6 Ghz Core 2 Duo Processor. At the time of program execution some essential background processes were also running in the operating system, the total CPU consumption prior to execution was 40%, and, that of memory was 70%. An excel file with random binary numbers was taken as input to the program.

No Of Tags	No of Collisions	Time Taken (in Sec)
8	8	0.858
16	16	0.874
32	33	0.889
64	65	1.123
128	130	4.009
256	258	11.138
512	515	36.712
1024	1027	108.445
2048	2053	413.011

Table 1: Results for different no of tags

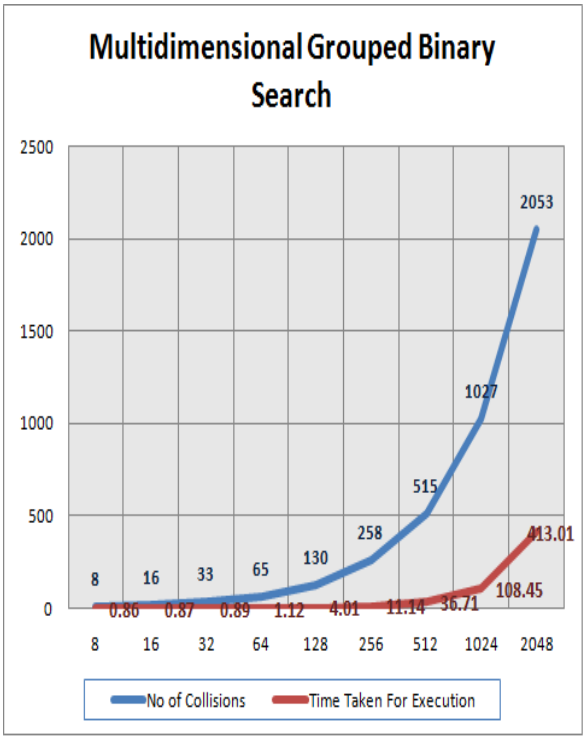

Figure 5: Graph showing the trend for no of collisions and time taken while reading tags

VI. Conclusion

In this paper we have proposed and implemented a multidimensional grouped backtracking binary search algorithm for reduced RFID tag collisions. In this paper we have proposed the algorithms required for simulation / implementation of the same. It is also proposed that by using the methodology of multidimensional grouping we can use a multiple reader based system, which will further reduce the collisions, and, at the same time will enable parallel processing. However, a separate research may be conducted for the exact hardware requirement for multiple frequency support of the proposed system.

REFERENCES

[1] Mingsheng Hu, Zhijuan Jia, Xiaoyu Ji and Liu Hong, "RFID Anti-Collision Algorithm Based on Occurrence Position", Research Journal of Applied Sciences, Engineering and Technology, vol. 5, pp. 23- 29, January 2013.
[2] Klaus Finkenzeller [2010]. "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification", 3rd edition, John Wiley & Sons.
[3] Geng Shu-qin, Wu Wu-chen, Hou Li-gang and Zhang Wang (2010). "Anti-Collision Algorithms for Multi-Tag RFID", Radio Frequency Identification Fundamentals and Applications Bringing Research to Practice, Cristina Turcu (Ed.), ISBN: 978- 953-7619-73-2,InTech, Available from: http://www.intechopen.com/books/radio-frequency-identification-fundamentals-and-applications-bringing-research-to-practice/anti-collision-algorithms-for-multi-tag-rfid
[4] N. Abramson, "The Aloha System—another Alternative for Computer Communications", Proc. Fall Joint Computer Conf., Am. Federation of Information Processing Soc. Conf., vol. 37, pp. 281- 285, Nov. 1970.
[5] J. Capetanakis, "Tree Algorithms for Packet Broadcast Channels", IEEE Trans. Information Theory, vol. 25, no. 5, pp. 505-515, Sept. 1979.
[6] Jihoon Myung, Wonjun Lee, Jaideep Srivastava, "Adaptive Binary Splitting for Efficient RFID Tag Anti-Collision", IEEE Communications Letters, Vol. 10, No. 3, pp. 144-146, March 2006
[7] Wang Jianfang, "A Novel Anti-collision Backtracking Algorithm Based on Binary-tree Search in UHF", Proceedings of the Third International Symposium on Computer Science and Computational Technology, ISCSCT 2010, pp. 212-214, August 2010
[8] Lvqing Yang, Caili Wang, Wenhua Zeng, "Research of Matrix-based Grouping Method on Anti-collision Algorithm for RFID Tag Identification", Advances in information Sciences and Service Sciences (AISS), Volume5, Number13, pp. 126- 123, August 2013
[9] Shilpa Bhatt,Deepal Dhadhwal,Ayush Bhatt , " RFID: Review & Comparison of Tree Based Tag Collision Resolution Algorithms" SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE) volume 2 Issue 3 March 2015
[10] Haosong Gou, Younghwan Yoo, "Bit Collision Detection Based Query Tree Protocol For Anti Collision In RFID System", International Journal of Innovation Computing, Information & Control , vol. 8, Number 5(A) , pp. 3081- 3102, May 2012.
[11] Jihoon Myung, Student Member, IEEE, Wonjun Lee, Senior Member, IEEE, and Jaideep Srivastava, Fellow, IEEE, "Adaptive Binary Splitting for Efficient RFID Tag Anti- Collision", IEEE COMMUNICATIONS LETTERS, VOL. 10, NO. 3, pp 144-146, MARCH 2006.
[12] Parveen Kumar, "Quadratic Search: A New and Fast Searching Algorithm (An extension of classical Binary search strategy)", International Journal of Computer Applications (0975 – 8887), Volume 65– No.14, March 2013
[13] Ajit Singh and Dr. Deepak Garg, "Implementation and Performance Analysis of Exponential Tree Sorting", International Journal of Computer Applications (0975 – 8887), Volume 24– No.3, pp. 34-38, June 2011.
[14] Ankit R. Chadha, Rishikesh Misal, and Tanaya Mokashi, "Modified Binary Search Algorithm", International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Volume 7– No. 2, April 2014.
[15] X.-L. Shi, X.-W. Shi, Q.-L. Huang, and F. Wei, "An enhanced Binary Anti-Collision algorithm of backtracking in RFID System", Progress In Electromagnetics Research B, Vol. 4, pp. 263–271, 2008
[16] http://www.slideshare.net/KALPNA06/rfid-presentation-37643386
[17] http://lbi.ro/teachinginnovatively/store/soft.php?soft=47
[18] http://wikipedia.unicefuganda.org/latest/A/Chemical%20kinetics.html