International Journal of Mathematical Archive-3(2), 2012, Page: 386-398
Available online through www.ijma.info ISSN 2229 – 5046

COLLISION RESOLUTION TECHNIQUE IN DENSE PASSIVE READ AND WRITE RFID TAGS

Katheeja Parveen M[a*], Sheik Abdul Khader P[a] and Munir Ahamed Rabbani M[b]

[a]Department of Computer Applications, B. S. Abdur Rahman University, Saudi Arabia
[b]College of Computer, Qassim University, Saudi Arabia
E-mail: katheeja.pareen@gmail.com

(Received on: 01-01-12; Accepted on: 20-01-12)

ABSTRACT

***R**FID technology allows information to be stored and read without requiring either contact or a line of sight between the tag and the reader. For this contact-less feature, it looked as an alternative solution to bar code in the distribution industry and supply chain, as it can hold more data. During heavy deployments, signal transmissions from multiple tags result in tag collisions or confusion causing increase in identification delay, communication overhead and unnecessary energy wastage that ultimately shortens the life span of the system. This paper has meticulously presented the existing anti – collision approaches for read and write passive tags and has also discussed the limitations in detail. Based on the comprehension gained, a new proposal is also illustrated for densely deployed tags which significantly outperforms the previously proposed algorithms.*

Keywords: *Passive Tags, Tag Anti – Collision, Deterministic, Probabilistic, Identification Efficiency of RFID systems*

1. INTRODUCTION

Radio Frequency Identification (RFID) is a technology that amalgamates the use of electromagnetic or electrostatic coupling in the radio frequency portion of the electromagnetic spectrum to uniquely identify an object, animal, or person with the help of an interrogator (reader), tag(s), and a host. RFID systems are implemented in time and data critical applications that include health care, supply chain management, asset tracking, aerospace, library and manufacturing. Out of the three types of tags namely passive, active and semi passive, passive tags have much demand due to their least system cost and longer life time. Passive tags are composed of an antenna coil and a silicon chip that includes basic modulation circuitry and non-volatile memory. Since they do not possess a battery on its own, the tag gets energized when a time-varying electromagnetic radio frequency (RF) wave passes through an antenna coil. An AC voltage which is generated across the coil, rectifies to supply power to the tag. The tag using the mechanism of backscattering transmits its ID to the reader. By detecting the backscattering signal, the reader demodulates the received signal to retrieve the tag's ID.

Passive tags are subdivided as Read only and Read / Write tags. Passive read-only RFID tags are designed for a broad range of applications like tracking, routing, and identifying vehicles, objects, animals, or people. Read/Write tags offer additional flexibility by allowing the data stored on the tag to be dynamically changed leaving the door opened for storing additional product data or sensor derived data for other applications.

RFID systems are a popular choice as the frequency band is not used by most of the electronic instruments or radiators deployed in industrial, scientific and medical fields. Deploying RFID systems in large scale industries faces myriad of challenges which are mostly related to multiple access problems, since communication between a reader and a tag is over a shared channel [1]. Moreover, due to financial prudence, users prefer to have a single reader to read multiple tags which causes tag collision. These collisions occur when multiple tags transmit their signals simultaneously. Besides collision, identification delay, tag communication overhead, idle responses, missed tag-reads, high energy consumption also needs to be addressed especially when the tag population is large.

Tag anti collision protocols have their root in TDMA, which is classified as probabilistic, deterministic and hybrid. Probabilistic based approaches are highly appreciated in deployments where time is critical while deterministic is a wanted choice in data critical RFID systems, where as hybrid algorithms are a combination of probabilistic and deterministic. All three approaches have their own limitations. Spread spectrum or similar techniques can be used to spectrally separate reader and tag transmissions, when permitted by local regulations.

***Corresponding author: Katheeja Parveen M[a]*, *E-mail: katheeja.pareen@gmail.com**

In spite of the tremendous amount of publications, the limitations of deterministic and probabilistic algorithms such as re-identification of staying tags, inefficiency in handling of arriving and departing tags, grouping tags whose ids are completely dissimilar causing collision at every bit still needs to be addressed in detail. Hence, it was felt that there is a need for the development of collision resolution algorithms that will blend in both time and data critical passive R/W RFID systems tuned to an environment with varying tag population.

The rest of the paper is organised as follows: Section 2 justifies the importance of the issue – Tag Collision and also throws light on the characteristics that an anti collision protocol should possess. Section 3 presents the existing algorithms from 30 years back to till date. In addition an analysis on the merits and deficits is also presented, as well as a consolidated report on the limitations and untaken care realistic environments that needs attention while implementing an algorithm is also emphasised. Section 4 exemplifies the proposed technique and the methodology behind is mentioned. Section 5evaluates the efficiency of the proposed algorithm based on the metrics namely number of collisions, identification time / delay, bit transmission by the reader, number of bits received by the reader, idle responses, energy consumption (while scanning and being idle), traffic intensity and possibility of occurrence of explosive states. Finally we conclude in Section 6.

2. PROBLEM DEFINITION

Tag collision is a major problem deferring tag identification of RFID system. Due to implementation cost constraints, it is often observed that vendors prefer to have a single reader to read accurately varied tag population which will also ensure to minimize number of collisions, identification time / delay, number of bits transmitted by the reader, number of bits transmitted by the tag, occurrence of idle responses, energy consumed by the reader while scanning, energy consumed by the reader while idle and algorithm execution time.

Tag anti-collision protocols that address this problem cannot be directly applied to the tag identification problem due to various constraints, which make this problem unique. These constraints are to be viewed as a requirement to keep the tags as cheap as possible. Many standard collision resolution protocols are for this reason non-applicable or difficult to implement in these systems. The characteristics of a good anti collision protocol are tabulated in Table 1.

Table 1 Characteristics of Anti Collision Protocol

#	*Characteristics*	*Description*
1	Minimal Delay	Time taken to identify all tags in the interrogation zone must be minimal as tagged objects are mobile.
2	Energy Consumption	Since passive tags do not carry any battery of their own, and are completely relies only on the energy generated by the reader, the consumption of power must be very minimal. The amount of power consumed is influenced by the total number of replies sent by each of the tags.
3	Reliability and Completeness	All the tags in the range of the reader should be identified. Tag starvation i.e., tags never getting the opportunity to interrogate with the reader should never be encountered. In other words, number of missed tag reads should always remain zero.
4	Robustness	The protocol should work irrespective of environmental conditions (dense / meager).
5	Scalability	Passive tags find their place in small and large scale systems. Hence the RFID reader must anticipate tags in any number in its readable range.
6	Readability	Passive tags can be read only at very short distances, may be from 0.1 m to 5m. When there is a delay in recognition time, due to mobility, the tags might leave the vicinity of the reader after the tag ID is half read.
7	Computational Complexity	Passive tags have limited computational capability, no ability to sense the channel, detect collisions, or communicate with each other. Any change in the tag circuitry is directly associated with cost.
8	Time Complexity	RFID is a time critical application oriented system, and hence delay caused during tag identification process directly degrades the efficiency of the system.
9	Message Complexity	Number of bit transactions between the sender (reader) and the receiver (tag/reader) should be limited.

3. RELATED WORK

Considering the limitations discussed, tag anti collision protocols are categorized into Space Division Multiple Access (SDMA), Code Division Multiple Access (CDMA), Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA) [2]. SDMA protocols are expensive and require complex antenna designs. While CDMA apart from being expensive, they are also in need of high energy source. On the other hand FDMA, requires a complex RFID reader as tags talk on one of the predefined frequency channels. The last of the four is TDMA, which constitute a larger group of anti collision protocols. TDMA protocols can either be Reader Driven (i.e., Reader Talk First) or Tag Driven (i.e., Tag Talk First). These protocols can further be classified as Probabilistic (Aloha based), Deterministic (Tree based) and Hybrid (combination of probabilistic and deterministic) protocols. Probabilistic requires tags to respond randomly in an asynchronous manner while most tree protocols operate by grouping responding tags into subsets.

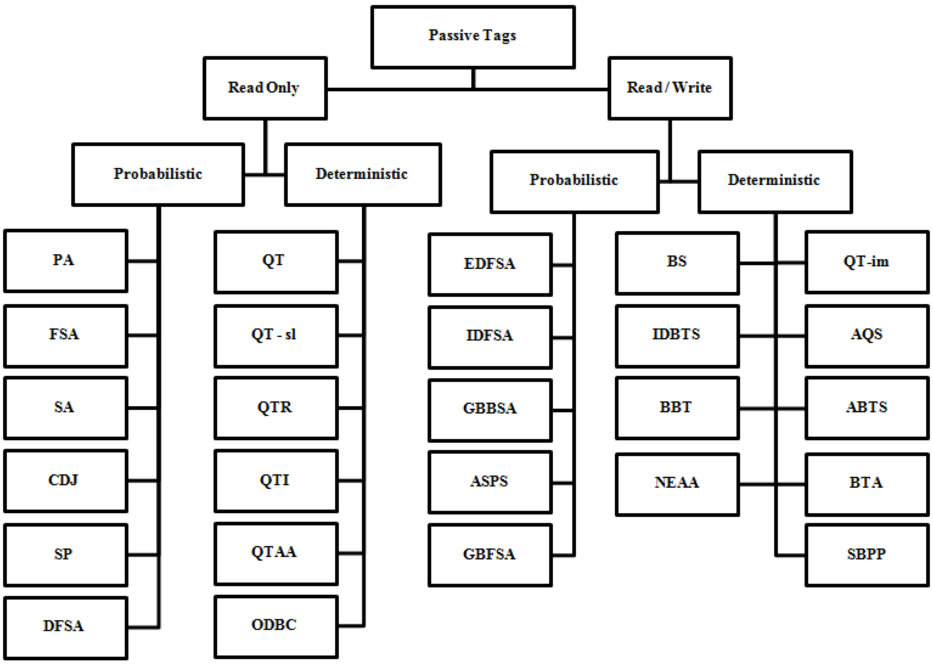


Fig 1: Tag Anti Collision Protocols

3.1. Probabilistic Protocols for R/W Tags

Enhanced Dynamic Frame Slotted Algorithm (EDFSA)
In EDFSA, when the number of unread tags is large, the algorithm divides the tags into groups based on tag estimation. However, in practical system, the thime taken to estimate the number of unread tag largely influences the number of slots per frame [3,4]. In addition the time spent on transmission prolongs the identification process [5].

Improved Dynamic Frame Slotted Aloha (IDFSA)
In IDFSA, the tags are designed to be divided into groups in different frequency channels to enhance the identification efficiency and to save the time of the command of the EDFSA. Geng Shu-qin et al [6] has also proven that the system identification is higher than the traditional method.

Grouping Based Bit Slot Aloha Protocol (GBBSA)
GBBSA begins the identification process by setting a bit slot count parameter "Q" in the query command. Upon receiving the query command, tags in the region group themselves by generating a random number between 0 and 2^Q – 1. Tags which have picked the random number as "0" immediately reply back to the reader and others wait for the next round. "Q" value is incremented or decremented for every round based on the number of collided bit slots. Even though GBBSA shows efficiency in identification, the delay that occurs due to the amendment of "Q" value for every round cannot be forgotten.

Adaptive Splitting and Pre – Signaling (ASPS)
ASPS [7] is a probabilistic counter-based tag anti-collision protocol, which uses two schemes, adaptive splitting and pre-signaling, to reduce tag collision and the number of messages sent between a reader and tags. In the first phase, the adaptive splitting scheme obeys ISO/IEC 18000-6B protocol until the first tag is identified successfully. It then enters

the second phase, in which the reader estimates the number of colliding tags of a specific counter value. The idea of pre-signaling solely relies under the assumption that the reader can distinguish three cases of identical responses from multiple tags by signal strength inspection.

Frame Slotted Aloha with Grouping Tactic and Binary Selection (GBFSA)
In GBFSA, each tag generates a random number between 0 and N-1, which represents the group number. Tags selecting the same number are elected to be in a sub group. The reader identifies the first group of tags and records the number of the group of tags based on tag estimation. Assuming uniform random distribution of tags, after estimating the number of tags in the first group, the algorithm multiplies it by N to obtain the total number of tags. If the number of tags in the first group is too small, the value of N is adjusted appropriately and the grouping procedure is repeated [8].

Spread Partial- Q Slot Count (SPSC)
SPSC algorithm is based on a slotted aloha CDMA technique along with tag set partitioning. The algorithm partitions the tags into a certain number of groups by taking advantage of different power levels received from the continuous wave form of the reader to power-up passive tags, which then backscatter accordingly with varied signal strengths.

Progressive Scanning (PS)
PS proposed by Weilian Su is similar to ODBC. The reader starts transmitting from a minimum EIR power level $P_{r,min}$ until the maximum $P_{r,max}$ that is permitted by the regulations. In each retransmission, the reader increases the transmitted power by an increment "*k*" and the tags that are farther in distance reply. The tags inside the range are identified by using Frame Slotted Aloha (FSA) protocol. After each scan, the frame size changes according to the frame size estimation algorithm. The first cycle starts by choosing a minimum frame size from the available set of frame sizes (64, 128, 256 and 512). This procedure is repeated until the transmitted power reaches $P_{r,max}$ [13].

3.2. Deterministic Protocols for Read and Write Passive Tags

QT- Incremental Matching (QTim)
QTim which is similar to QTsl, was proposed to reduce the number of query bits transmitted by making a tag to remember the bit position of the prefix it has matched so far. The tag communication complexity is same for both the protocols, however the number of bits transmitted is still reduced.

Binary Search Tree (BST)
In BST protocol, the reader performs identification by recursively splitting the set of answering tags. Each tag has a counter initially set to zero. Only tags with counter set to zero are allowed to answer at the reader's queries. After each tag transmission, the reader notifies the outcome of the query namely collision, identification, or no answer. During the event of collision, each tag with counter set to zero adds a random binary number to its counter and non-zero counter tags increase by one their counters. In such a way, the set of responding tags is split into two subsets. After a no-collision transmission, all tags decrease their counters by one [9].

Enhanced Binary Search Tree (EBST)
EBST is presented as a variant by Yu et al [16]. The key difference to BS is that EBSA does not restart the reading process after a tag is identified. In EBST with Cut Through, a tag is represented by the labels in the edges along the tree from the root to the leaf node. The value in the leaf node represents the tag of the RFID node. On the other hand, the values in the intermediate nodes represent the number of RFID nodes with the same prefix from the root to this node [17].

Adaptive Binary Tree Splitting (ABTS)
Myung et al [10] proposed ABTS, an enhancement over BTS algorithm introduced by Hush and Wood [11]. ABTS achieves fast identification by reducing not only collisions but also unnecessary idle slots. Similar to the BTS algorithm, tags can either be in transmit or wait state. However, unlike BTS, tags have two counters, Progressed Slot Counter (PSC) and Allocated Slot Counter (ASC). The detailed identification process can be found in [10].

Enhanced Binary Tree Splitting (EBTS)
Chen et al [12] present a variant of the ABTS algorithm called EBTS. It uses Manchester coding to identify the location of collided bits. If a collided bit is detected, the reader stops tags from transmitting the remaining bits of their ID. Each tag maintains a pointer that stores the location of the first collided bit. If the pointer has a value k, it indicates that the k^{th} bit suffered a collision. In other words, all bits prior to the k^{th} bits have been received correctly. Thus, in future read requests, tags only need to transmit those bits from their ID that occur after the k^{th} bit. These bits are then identified using ABS.

Adaptive Query Tree (AQT)

Myung et al [10] proposed a protocol called AQT, where the reader is required to maintain a queue that operates similar to the stack in QT algorithm. In addition, the reader is required to maintain a candidate queue (CQ) to store queries sent in past identification rounds. AQT reduces collisions, but it generates some idle cycles. To guarantee the recognition of all (staying and arriving) tags, the reader uses not only queries of readable cycles but also queries of idle cycles of the last frame. Though the reader eliminates unnecessary idle cycles by leaving tags, some idle cycles cannot be avoided in order to cover all possible ranges of the tag ID.

Scanning Based Pre Processing (SBPP)

Choi et al [18] proposed SBPP technique that uses Manchester coding to locate collided bits in tag responses. The reader notifies tags the whereabouts of these collided bits, and uses a QT algorithm to identify them. The performances of SBPP aided anti-collision protocols are in the middle of between the worst case and the best case.

Bit by Bit (BBT) and its variants

Jacomet et al [24] presented a BBT arbitration method where a separate channel is used for binary zero and one. When requested, each tag transmits the specified bit in one of these channels. If the reader receives a different response from both channels, it sends a control bit silencing the subset of tags that replied with 0 (or 1). On the other hand, if the reader receives a response in only one of the two channels, a bit is identified successfully. Similar to ID-BTS, the reader has a stack and each tag has a counter to store its tree position.

Modified bit by bit binary tree (MBBT) algorithm proposed by Choi et al (2004), operates in a similar manner to the BBT algorithm. The key difference is that MBBT does not use multiple timeslots to receive binary 0s and 1s. He also proposed Enhanced bit by bit binary tree (EBBT) algorithm. In EBBT, a reader first requests tags to respond with their complete ID. The assumption here is that tags responses are synchronized. From these responses, the reader identifies collided and collision-free ID bits.

Tree Slotted Aloha (TSA)

Like in FSA, TSA is not memoryless, since each tag has to remember the random number generated in the previous cycle, and the level of the tree [15]. However the amount of memory needed by each tag is very small, namely few bits to remember the tree level and the slot number of the previous reading cycle. The tree built by TSA protocol is identical to such random hash tree, if we would know the exact number of tags. So, if this is the case, we can assert that the average number of slots needed to identify n tags is 2.3020238n.

New Enhanced Anti Collision Algorithm (NEAA)

In NEAA [20], M-readable and counters are used to identify more than one tag in a single timeslot. Tags are classified into a unique set by grouping those tags whose sum of the bits is the same. Like ABS, tag maintains two counters (ASC and PSC). Apart from these counters, the tag also maintains another counter named "pointer" to record the first reply bit of a tag ID. Reader also maintains a counter named TSC and a stack to store the bits received. During collision, the reader informs tags where the collided bit is and the counters namely ASC, PSC and TSC works similar to ABS.

Optimal Distance Based Clustering (ODBC)

ODBC [21] works by dividing the interrogation zone into "k" sized clusters based on the distance to the reader. Partitioning the interrogation zone can be achieved by controlling the reader's antenna power level. The tags in different clusters are interrogated separately. To read tags in one cluster, the RFID reader and tags in that cluster can adopt any one of the anti collision resolution protocols and get identified by surpassing through several cycles. After identification, the tags are marked as read and made to be muted by sending a sleep command. The identification process continues further until the reader reaches its maximum interrogation range.

Divide and Conquer Technique

Divide and Conquer Technique algorithm blindly partitions the tags and probes the first subset of tags to estimate the tag population. The partioning algorithm consists of three stages i.e., an initial estimation of tag population is made, repartition the tags into optimal number of subgroups to yield maximum throughput and finally identify the remaining subgroups in sequence. The rationale behind this method is that higher throughput can be obtained by performing identification for multiple smaller subsets of equal size instead of doing so for all tags at once. In particular, the estimate of the number of tags in a subgroup becomes more accurate as more subgroups have been identified. By taking the average of the tag populations in subgroups that have already been processed, an accurate tag estimate of the size of the next subgroup can be obtained [22].

3.3. Discussions

It is often observed that most probabilistic algorithms compensate the time taken to avoid collisions with tag estimation. In these approaches, most algorithms (Divide and Conquer Technique, EDFSA, etc.) adopt the idea of randomly backing off. Nevertheless, randomly backing-off or waiting without responding for a pre-determined delay period should not be the only choice for reducing the number of collisions as there may arise chances that a tag may not respond, i.e., remain idle even when the channel is free. This situation can frequently be experienced when tags leave the interrogation zone during the reading cycle or before it is identified by the reader. In addition, how long should the tag remain asleep is not mentioned in the algorithms. In GB-FSA, random number grouping is suggested to avoid collisions. Tags are expected to generate a random number based on the maximum frame size. Implementation of this technique requires a change in the tag circuitry as well as additional responsibility to generate a random number. Moreover, the electing a particular random slot can also provide chances to increase collision at every bit inside the group [19].

Implementation of QT-im can be considered as a rescuer from QT and its variants for reducing the number of bits transmitted. It has provided new channels of enhancement which has lead to the introduction of ABTS and AQT by Myung et al [10]. Even though both algorithms have taken efforts to reduce collisions, production of idle cycles, however could not be avoided. NEAA is an improvement of AQT which proposes the idea of counting the number of bit 1's (or) 0's present in a tag id. Those tags who match the total count of bit 1's or bit 0's form a group, and follow up their identification process group by group. Consider a scenario where there are two tags having tag Id's 1001 and 0110 joining hands together to form a group, as their number of bit 1's match. It is observed from the tag Id's that collision occurs at every bit, and hence uniting these two tags may not provide solution to reduce the number of collisions. Moreover, NEAA needs an extra pointer when compared to ABS, to record the first reply bit during a collision time slot. Even though NEAA shows better result than QTim, it requires change in tag circuitry. In GBBSA, reader sends a query, by which the tags randomly generates a reservation sequence based on the number of bits. Clustering of tags based on the distance (ODBC) and power levels (PS and SPSC) are also proposed by researchers in order to resolve collision. However, they may face the same issue as discussed in the earlier example for NEAA.

To summarize, the limitation of the existing algorithms can be consolidated as,

(1) re-identification of staying tag
(2) inefficiency in handling of arriving and departing tags
(3) grouping tags whose tag ids are partially or completely dissimilar, thus causing collision at every bit
(4) possibility of tag starvation
(5) change in tag hardware design, leading to increased system cost
(6) inapt assumptions like tags can be feigned to detect collisions, broadcast collision occurrence to neighbouring tags
(7) introducing innumerable tag pointers and reader counters
(8) readers can make use of one of the tag estimation functions to estimate the number of unread tags to evaluate the frame size

Also, the tag collision algorithm designed must be devoid of the following unrealistic assumptions,

(1) One of the tags can act as a cluster head to initiate communication.
(2) Tags can be designed to react to collision by gathering information from other tags
(3) Tags can inform its own near neighbours when collision occurs or when it begins transmission
(4) Tags can be programmed to calculate complex modulo or matrix operation
(5) Tags can calculate its distance from the reader at every instance and based on that it may decide to reveal its identity
(6) Uniform distribution of tags in a sub groups, i.e., from the knowledge obtained from identifying one sub group, the estimation of total number of tags in the range can be assessed.

4. PROPOSED WORK

Keeping in the background, the contributions discussed in the previous section, the main aim of this research is to develop a collision resolution algorithm, Least and Most Significant Bit Arbitration Algorithm for Read and Write (LaMSBA_RW) passive tags that account for the following scenarios,

(1) Similarity in the distribution of Tag ID.
(2) Tags staying for more than one reading cycle.
(3) Arrival of New Tags before / after the completion of a reading cycle
(4) Departure of Tags before / after the completion of a reading cycle

In [23], the author has presented the efficiency of LaMSBA algorithm for WORM (Write Once Read Many) tags. The proposed algorithm is an enhancement of LaMSBA for reducing the number of collisions, idle responses, bit transmission and identification time.

Reader begins its session (reading cycle) by broadcasting a SEND_ID command. Upon receiving the command, tags in the given range backscatter their ID bit by bit. Manchester code is constructed at the reader's end to identify the collided and non-collided bit positions. Reader maintains 2 variables to store the least bit position (first collided bit from extreme left to right without intermediate collided bits) and most bit position (first collided from extreme right to left without intermediate collided bits) based on the Manchester code. A stack is constructed where in each member of the stack stores the non-collided bit values in the respective non-collided bit positions in consultation with Manchester code. If least and most bit positions correspond to collided bits, then push the bit values – 1---1, -1---0, 0---1, -0---1-, -0---0- to the members of the stack. If in case collision occurs at least bit alone, then to the stack push the probable queries -0---k-, -1---k-, where "k" corresponds to the non-collided most bit position. Similar method is adopted when collision occurs at most bit alone. Finally if collision has not existed in both positions, then to the member of the stack fill in the received least bit position value and most bit position value. The reader broadcasts the bit position and its corresponding values to the tags. Tags whose bit positions and bit values matches, increment and decrement their counters accordingly and backscatter the next least bit position and most bit position and its corresponding values. In case the received bits collided at both positions or at one, then the reader repeats the procedure as discussed earlier. The reader repeats the cycle by popping the first member and broadcasting the bit positions. Simultaneously, the reader updates its counters before every transmission. After successful identification of a tag, the reader decrements / increments its most / least bit counters. The reader requests the identified tag to store the broadcasted Reader_ID and random number. The reader records this random number against the identified tag ID. This procedure is repeated until stack is empty. Since the readers' stack is constructed only on the basis of Manchester code and tags' response, the algorithm provides numerous advantages which include reducing collision, idle response, bit transmission and energy consumption. The step by step procedure is illustrated in Table 4.1.

Table 4.1 Reader and Tag operations of LaMSBA_RW

#	**Description**	
1	Reader	Broadcast the beginning of a reading cycle. Send command to tags to reply bit by bit
2	Tags	Respond ID bit by bit
3	Reader	Identify the collided and non-collided bit positions. Push the non-collided bit values in their respective position to the reader stack
4	Reader	Pop the first element of the reader stack and initialize the following variables: nlb = highest non-collided bit position from the extreme left to right without intermediate collided bits least_bit_pos = nlb most_bit_pos = least non-collided bit position from the extreme right to left without intermediate collided bits
5	Reader	Broadcast "least_bit_pos" and "most_bit_pos" to Tags
6	Tags	Store least bit position, most bit position
7	Reader	a=least_bit_pos_val ; b=most_bit_pos_val If collision occurs at "Least Bit" and "Most Bit" position, Reader constructs a stack with 4 probable partial ID (PPID) -0 - - - - - - - - - 0- -0 - - - - - - - - - 1- -1 - - - - - - - - - 0- -1 - - - - - - - - - 1-
		Else if Collision occurs at "most bit" alone then Reader constructs a stack with 2 PPID -a - - - - - - - - - 0- -a - - - - - - - - - 1-
		Else if Collision occurs at "least bit" alone then Reader constructs a stack with 2 PPID -0 - - - - - - - - - k- -1 - - - - - - - - - k-
		Else // *no collision* Reader constructs a stack with 1 PPID -a - - - - - - - - - k-
8	Reader	Pop the first element from the stack
9		Find for matching tags
10	Tags	Un matching tags mute themselves
11	Tags	if matching_tags_count does not exist then // idle response if reader stack is not empty then

		goto step 8 end if end if
12	Reader	Broadcasts a command to ask matching tags to reply least_bit_pos +1 and most_bit_pos -1
13	Tags	Matching Tags respond their next least_bit_pos and most_bit_pos bit values and increment / decrement their stored least and most bit position
14	Reader	If tag identified, then decrement / increment in least and most bit position
		Goto step 7
		Repeat the steps until stack is empty

5. PERFORMANCE ANALYSIS

A simulator is designed in C++ to evaluate the performance of the proposed algorithms. For simulation, one RFID reader with 10,000 tags in its interrogation zone is considered. The reader is modelled based on the design features of SkyeTek's M1 – Mini RFID Reader. This reader operates from a Lithium rechargeable battery which has 0.48 KJ of energy. The tag to reader data rate is taken as 26 Kbps as per ISO 15693.
Since in mobile environments, tags may arrive and (or) depart continually which results in indefinite change in tag population size, the environment follows single channel, Poisson arrivals, exponential service, infinite population, service in random order (M/M/1 : SIRO/∞/∞) as

- Tag arrivals are according to Poisson process a rate of λ_T arrivals per second
- Service time of a reader is exponentially distributed
- The environment has only one reader
- The length of the queue in which arriving tags wait before identified is infinite
- The tag population available to join the system is infinite

Incoming traffic to the RFID system is subjected to Erlang's queuing theory under the assumption of pure chance traffic i.e., tags arrivals and departures are random and independent events.

To analyse the efficiency of LaMSBA_RW algorithm, the following evaluations are made

1. Number of tags identified in a second can be calculated as,

$$\text{Identification Rate per second}\,(\mu) = \frac{\sum_{i=0}^{i=n-1} Tags_i \;\; (No. of\ Tags\ in\ the\ interrogation\ range)}{Time\ taken\ to\ identify\ all\ tags}$$

2. Since the environment subjected for study is mobile, tags are modelled to arrive in a random interval. Hence the number of tags arriving on an average per second can be estimated as,

$$\text{Tag arrival Rate per second}\,(\lambda) = \frac{No. of\ Tags\ arriving\ in\ the\ interrogation\ range}{inter - arrival\ time}$$

3. To find if the environment is in steady state or explosive state

$$\text{Traffic Intensity}\,(\rho) = {\lambda (mean\ arrival\ rate)}\big/{\mu\ (mean\ service\ rate)}$$

   The system is said to be stable only if $\lambda < \mu$. If the death rate is less than the birth rate, the average number of tags in the environment will become infinite.

4. To measure the idle time of a reader in percentage

$$Idle\ time = 1 - \rho$$

5. To deduce the average duration of a tag before it is being identified can be represented as,

$$\text{Expected waiting time of a tag to be identified} = {\lambda}\big/{\mu(\mu - \lambda)}$$

6. $\text{Probability that an arrival tag has to wait for "t" seconds} = \int_t^{\infty} (\mu - \lambda) e^{-(\mu-\lambda)w}\, dw$

The following assumptions are made,

- Noise free channel is considered, i.e., packet losses occur only due to collision

- Tag's antenna is never at 90 degrees
- RFID Readers are allowed to transmit energy until all tags are read
- Communication from Tag to Reader is modelled as Poisson process
- All tags in the reader's interrogation zone are of same length
- Reader has the knowledge on the number of bits present in a tag ID
- Reader is unaware of the number of tags
- Although tags are energized at the same time, the energy consumption is estimated only after the reading process has started.
- The delay associated with energizing tags, propagation and processing delays are omitted.

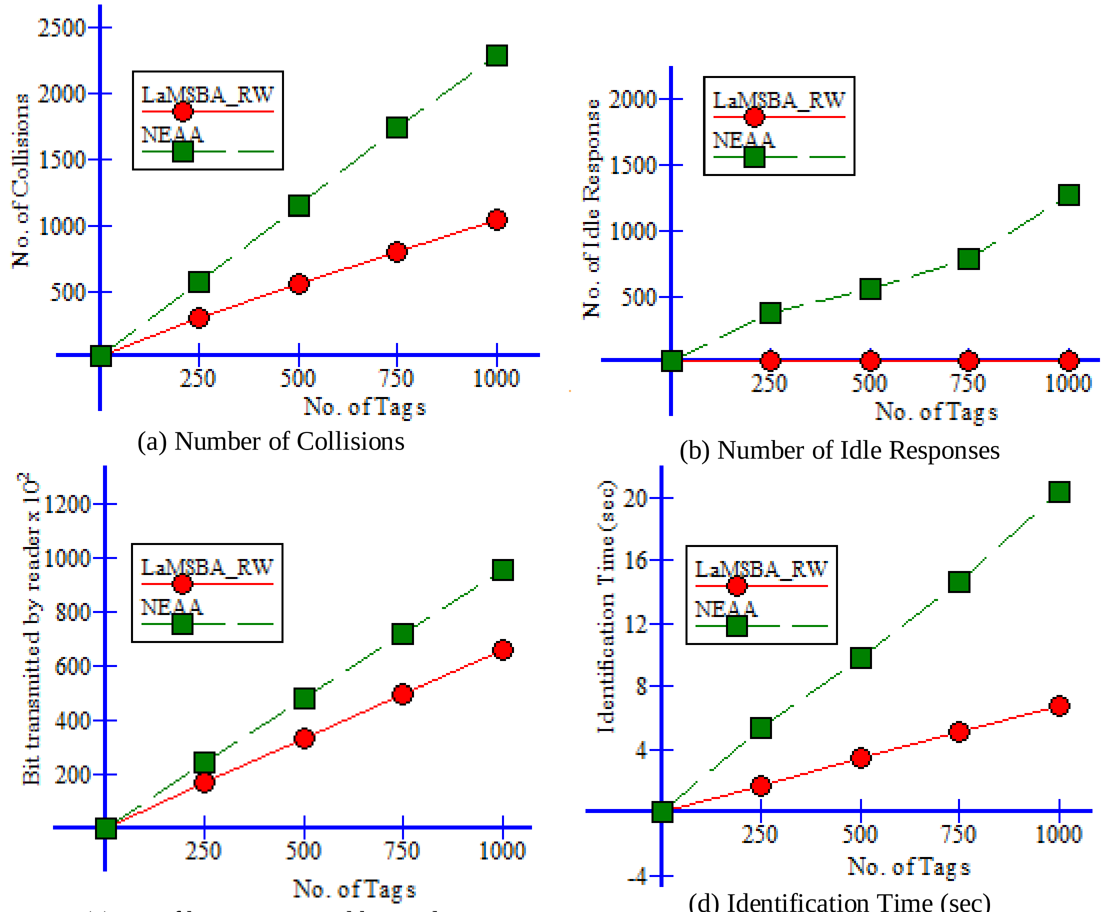
The consolidated simulation parameters are described in Table 2.

Table 2: Simulation Parameters

Parameter	**Value**
Terrain	20 m x 20 m
Number of UHF Readers	1
Reader's Identification Range	3 m
Number of Tags	5,000
Tag ID	Randomly selected 96 Bit ID
Tag Mobility	Random Walk Model
Maximum Speed of a Mobile Tag	2 m/sec
Stationary Probability	0.5 sec
Time spent for a reader to determine a Idle Response	0.5 ms
Duration of a silence command	0.19 ms
No of Bits transmitted to issue a silence command	5 bits
Power consumed by the Reader during Scanning	180 milli watts
Power consumed by the Reader during Idle Time	30 milli watts
Power consumed by the Reader during Sleep	50 milli watts
Data Rate	26 kbps / 40 kbps / 80 kbps

5.1. Impact of Similarity in the Distribution of Tag IDs

The distribution of Tag ID bits is one of the factors that influence the performance of tree based algorithms. To evaluate the distribution, an identical bit, γ is defined. For simulation, tag ID's are generated such that each tag ID has $(x,\ x_2, \ldots\ x_\gamma,\ x_{\gamma+1}, \ldots.\ x_{96})$ wherein the first few bits $(x_1,\ x_2, \ldots\ x_\gamma)$ remain the same for all tags. Simulation results have shown that NEAA has a high communication overhead as tree splitting is based on collided bits and M-readable slots. On the contrary, LaMSBA_RW performs much better since it does not completely rely on the pattern of IDs. Besides as the identical bit increases, the performance also rises proportionately (figure 2(d)). This proportionate increase can be attributed towards the consideration of Manchester encoding as a first step which helps in the differentiation of collided and non – colliding bits. As observed in the graph (figure 2(a)), number of collisions increases very gradually when compared with NEAA whose collisions increases steadfastly. Notably, idle responses (figure 2(b)) remain null as the members of the reader stack are built on the basis of Manchester code and tags' response.



(a) Number of Collisions

(b) Number of Idle Responses

(c) No of bits transmitted by reader

(d) Identification Time (sec)

Figure 2 : Impact of Similarity in the Distribution of Tag IDs

5.2. Impact of Staying Tags

To evaluate the impact of staying tags, tags in this scenario are neither allowed to move away from the reading range or new tags are allowed to enter inside the range during / after the reading cycle. In addition, the tag id bit values are also randomly distributed, i.e., there is a probability that collision may occur at every bit. Let a_x denote tag *x*. Let $A_{(r,i)}$ constitute the set of tags which reside inside reader's range, *r* in the i^{th} reading cycle of a reader. For the given i^{th} cycle of a reader, tag is said to be a staying tag if it was recognized in the last reading cycle $(A_{(r,i-1)})$, and stays within reader's range in the current reading cycle $(A_{(r,i)})$ i.e. a staying tag is one of $\{a_x : a_x \in A_{(r,i-1)} \cap A_{(r,i)}\}$. Since staying tags have been recognized in the last reading cycle, the reader need not re-recognize staying tags in the current identification cycle. Nevertheless, most existing tree based algorithms cause collision between staying tags as they do not take account of the staying tag information. As the number of staying tags increases, collisions occur more frequently in NEAA.

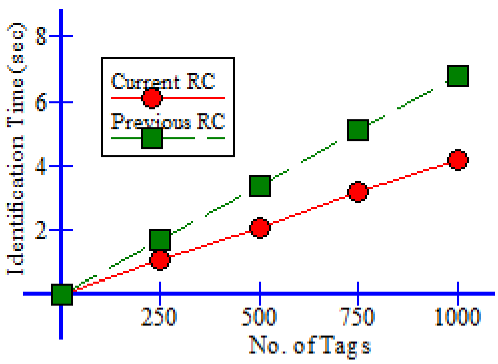


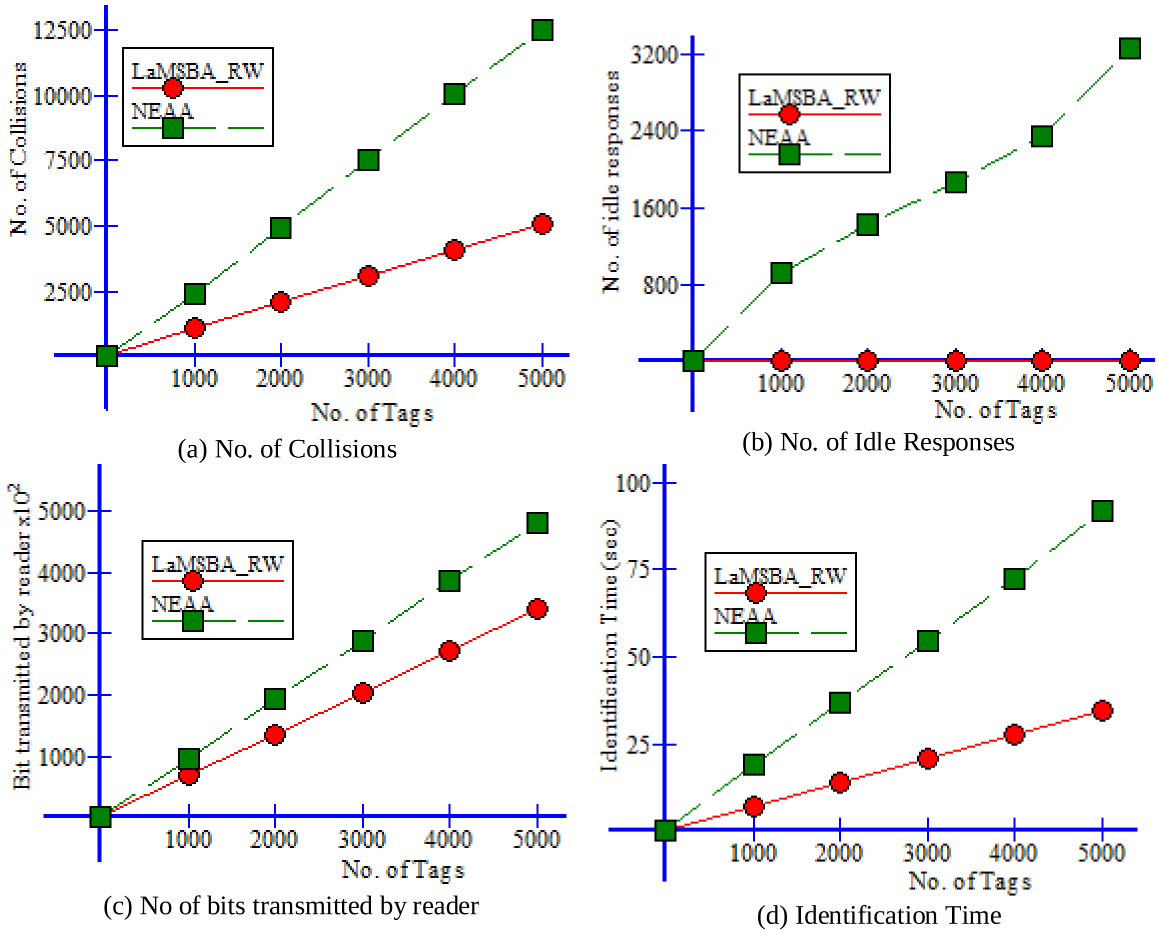
Figure 3: Identification Time of LaMSBA in Current and Previous RC *RC – Reading Cyle

Consequently, NEAA produces a similar identification delay as observed in the previous cycle. LaMSBA_RW avoids collisions between staying tags by asking the tags to record the reader ID and random number when it was identified. Hence the algorithm begins it session by broadcasting the reader ID and instructs those matching tags to respond their random number.

Upon receiving, matching tags backscatter the stored random number. When number of tags respond resulting in collision, the algorithm follows the procedure of PA with fast mode and muting to identify the staying tags. The choice of PA with fast mode and muting was made due to its least consumption of energy. QT-sl can also be implemented, if the reader is designed to store the identified tag Ids like in AQS. However, if the number of staying tags is huge, it is preferable to use LaMSBA_RW. The motivation behind our proposal to store the Reader's ID in the tag is that in most RFID deployed environments, the tag readings do not defer drastically in successive scanning sessions.

5.3. Impact of Newly Arriving Tags

After having recognized 1000 tags in the first identification round, 5000 new tags are allowed to enter for recognition. A tag is set to be an arriving tag if it has just arrived in the readers' range before the start or during the course of a cycle i.e. $\{a_x : a_x \in A_{(r,i)} - A_{(r,}$. Generally, as the number of arriving tags increases in the interrogation zone, the identification delay also increases substantially.



(a) No. of Collisions

(b) No. of Idle Responses

(c) No of bits transmitted by reader

(d) Identification Time

Figure 4: Impact of Arriving Tags

In LaMSBA_RW, the reader summons the staying tags to broadcast their random number as discussed in earlier section. After identifying the staying tags, the reader begins its session to identify the arriving tags by using LaMSBA_RW algorithm. With reference to NEAA, number of collisions (figure 4(a)) and identification delay (figure 4(d)) increases as new tags arrive. This could be owed to the fact as NEAA has not differentiated between a staying tag and an arriving tag in the current cycle. As observed in Table 3, RFID systems with NEAA enter into an explosive state very earlier (λ = 60 tags/ sec), when compared to LaMSBA_RW (λ = 160 tags/ sec).

Table 3: No of tags identified per second

Data Rate (kbps)	Identification Rate (μ)	
	LaMSBA_RW	NEAA
26	142	52
40	148	53
80	154	54

Table 4: Traffic Intensity at Reader's end

λ = Arrival Rate (No. of tags arriving per second)	Traffic Intensity ($\rho = \lambda/\mu$)					
	LaMSBA_RW			NEAA		
	μ = 142	μ = 148	μ = 154	μ = 52	μ = 53	μ = 54
20	0.14	0.13	0.12	0.38	0.37	0.37
40	0.28	0.27	0.25	0.77	0.75	0.74
60	0.42	0.40	0.38	***1.15***	***1.13***	***1.11***
80	0.56	0.54	0.51			
100	0.70	0.67	0.64			
120	0.84	0.81	0.78			
140	0.98	0.94	0.90			
160	***1.12***	***1.08***	***1.03***			

Impact on Departing Tags

Tags may leave a reader's interrogation zone and then cause idle slots, unnecessarily prolonging the monitoring process. Leaving tag is a tag which was recognized in the last frame by reader *r*, and has left from the reader's range before the starting of the current frame, i.e. $\{a_x : a_x \in A_{(r,i-1)} - A$. NEAA produces a number of idle slots which increases the identification delay while LaMSBA reduces these slots by removing all probable partial queries from the stack which does not have a tag response instantaneously.

6. CONCLUSION

Deploying an RFID system in large scale industries has myriad of challenges, many of which are related to the multiple access problems. Since the communication between a reader and a tag is over a shared channel, these systems are prone to collisions often. Moreover, due to the short radio range, the replies arrive almost simultaneously at the interrogator causing tag collision. Currently, this problem is handled by establishing a two way communication link between the tags and the interrogator. Least and Most Significant Bit Algorithm (LaMSBA) is implemented to suit to erratic environments. The construction of probable queries by considering the collided bits from the extreme left and extreme right instead of scanning from left to right as in Query Tree or from right to left as performed QT – Reversed has considerably reduced the number of collisions, identification delay, tag communication overhead and energy consumption. The strategies used in this algorithm can be considered as one of the better alternatives for reducing the number of collisions in time and data critical passive RFID systems.

REFERENCES:

[1] Seshargiri K.V., Pavel V. Nikitin, Sander F. Lam (2005), "Antenna Design for UHF RFID Tags: A Review and a Practical Application", IEEE Transactions on Antennas and Propagation, Vol 53. No 12.

[2] Rohatgi A. and Durgin G. D. (2005), "RFID anti-collision system using the spread spectrum technique," in Technical Report, pp. 1–19.

[3] Katheeja Parveen M, Sheik Abdul Khader P, Munir Ahamed Rabbani M (2011), "Cluster bit collision identification for recognizing passive tags in RFID", Advanced Computing Communications in Computer and Information Science, Vol. 133 Part 2, pp. 190-199.

[4] Vogt H. (2002), "Multiple object identification with passive RFID tags," in The IEEE Intl. Conf. on Man and Cybernetics, (Tunisia), Vol. 3, pp 6–13.

[5] Su-Ryun Lee, Sung-Don Joo and Chae Woo Lee (2005), "An Enhanced Dynamic Frame Slotted Aloha Algorithm for RFID Tag Identification", in Proc. of the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous '05), pp 1-6.

[6] 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, pp 103-112.

[7] Ming – Kuei Yeh, Jehn – Ruey Jiang and Shing – Tsaan Huang (2009), "Adaptive Splitting and pre-signalling for RFID tag anti-collision", Computer Communications 32(2009), pp 1862-1870.

[8] Wang Ya-qi, Jiang Guo-ping and Wang Jing (2009), "Frame Slotted Aloha with Grouping Tactic and binary selection for anti-collision in RFID systems", The Journal of China Universities of Posts and Telecommunications, Vol. 16, No. 4, pp 47-52.

[9] Finkenzeller K. (2003), "RFID Handbook Fundamentals and Applications in Contactless Smart Cards and Identification", John Wiley and Sons Ltd.

[10] Myung J., Lee W., and Srivastava J. (2006), "Adaptive binary splitting for efficient RFID tag anti-collision", IEEE Personal Communication Mag., vol. 10, no. 3, pp. 144–146.

[11] Hush D. R. and Wood C. (1998), "Analysis of tree algorithms for RFID arbitration," in The IEEE Intl. Symposium on Information Theory, (Mexico City, USA), pp. 107–107.

[12] Chen W.C., Horng S.J. (2007), and Fan P. , "An enhanced anti-collision algorithm in RFID based on counter and stack," in Second International Conference on in Systems and Networks Communications, pp. 21–24.

[13] Weilian Su, Nikolas V Alchazidis and Tri. T. Ha (2010), "Multiple RFID Tags Access Algorithms", IEEE Transactions on Mobile Computing, Vol. 9, No. 2, pp. 174-187.

[14] Thomas La Porta F., Gaia Maselli and Chiara Petrioli (2011), "Anticollision Protocols for Single-Reader RFID Systems: Temporal Analysis and Optimization", IEEE Transactions on Mobile Computing, Vol. 10, No. 2, pp 267- 279.

[15] Bonuccelli M. A., Lonetti F., and Martelli F. (2005), "Randomized hashing for tag identification in RFID networks", in Technical Report: Computer Communications and Networks.

[16] Yu S., Zhan Y., Wang Z., and Tang Z. (2005), "Anti-collision algorithm based on jumping and dynamic searching and its analysis", Computer Engineering, vol. 31, pp. 19–20.

[17] Chen W.C., Horng S.J. (2007), and Fan P. , "An enhanced anti-collision algorithm in RFID based on counter and stack," in Second International Conference on in Systems and Networks Communications, pp. 21–24.

[18] Choi J. H., Lee D., Youn Y., Jeon H., and Lee H. (2006), "Scanning based pre-processing for enhanced RFID tag anti collision protocols," in International Symposium on Communications and Information Technologies (ISCIT).

[19] Katheeja Parveen M, Sheik Abdul Khader P, Munir Ahamed Rabbani M (2010), "B-bit pooling memoryless algorithm for anti-collision resolution in passive RFID Tags", 2010 IEEE International Conference on Communication Control and Computing Technologies, pp. 357-365.

[20] Yuan-Hsin Chen, Shi-Jinn Horng, Ray-Shine Run, Jui-Lin Lai, Rong-Jiang Chen, Wei-Chih Chen, Yi Pan and Terano Takao (2010), "A Novel Anti-Collision Algorithm in RFID Systems for Identifying Passive Tags", IEEE Transactions on Industrial Informatics, Vol 6, No. 1, pp 105-120.

[21] Waleed Alsalih, Kashif Ali and Hossam Hassanein (2008), "Optimal distance based clustering for tag anti collision in RFID systems", in IEEE Conference on Local Computer Networks, pp 266-273.

[22] Jeong Guen Kim (2008), "A Divide and Conquer Technique for Throughput Enhancement of RFID Anti-Collision Protocol", IEEE Communications Letters, Vol. 12, No. 6, pp 474-476.

[23] Katheeja Parveen M, Sheik Abdul Khader P, Munir Ahamed Rabbani M (2010), "Least and Most significant Bit Arbitration algorithm in uniform distributed passive RFID tags", Proc. of 2010 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1-4

[24] Jacomet M., Ehrsam A., and Gehrig U. (1999), "Contact-less identification device with anti-collision algorithm," in Conference on Circuits Systems, Computers and Communications, (Athens, Greece).

[25] Lee S. R., Joo S. D and Lee C. W. (2005), "An enhanced dynamic framed slotted Aloha algorithm for RFID tag identification," in The 2nd Intl. Annual Conference on Mobile and Ubiquitous Systems: Networking and Services, (San Diego, USA), pp. 166–172.

****

