This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2010 proceedings

# Energy-Efficient Parallel Singulation in RFID

Kashif Ali, Sharief M. A. Oteafy and Hossam S. Hassanein
School of Computing, Queen's University, Kingston, Canada
{kashif, oteafy, hossam}@cs.queensu.ca

***Abstract*—Tag collisions impose a significant hindrance to reading rates of Radio Frequency Identification systems. The parallel singulation approach, being a major milestone, clusters tags and autonomously interrogates each cluster in parallel. This technique reduces the number of tags being interrogated at a given time, reducing collisions, and achieves higher reading rates. However, such an approach faces two limitations as the number of clusters increase. The exponential increase in tag responses may hinder tag functionality due to energy spent on communication. Moreover, energy inefficiency is incurred at cluster-heads to process significantly more tag responses. These issues overshadow the promising benefits of employing parallel singulation. In this paper, we remedy such hindrances by proposing energy efficient enhancements to the parallel singulation technique. The essence of these enhancements lies in minimizing an important measure of communication overhead, referred to as *tags traffic rate*, which indicates the efficiency of interrogation cycles in communicating with all tags without incurring unnecessary overhead. Analyses carried out via simulation demonstrate significant improvements by the proposed schemes in reducing energy consumption of cluster-heads, without posing constraints on tag operations nor incurring significant degradation of reading rates.**

## I. INTRODUCTION

The functional limitation of passive Radio Frequency IDentification (RFID) tags in sensing its wireless medium translates into simultaneous medium access, hence resulting in collisions which undermine overall performance of RFID systems. Tag collisions are problematic as a tag has limited power and functionality. A passive tag can only transmit data by reflecting the reader's transmitted electromagnetic waves, hence cannot detect nor communicate with neighboring tags. Thus, efficient tag anti-collision mechanisms are required to sustain system operability. Several schemes have been proposed to deal with tag collisions in RFID systems. These schemes are broadly classified as either deterministic or probabilistic [1], [2]. Deterministic mechanisms and its variants [3] are natively tree-based algorithms, which split tags based on colliding bits in preceding interrogation cycles, in addition to their respective positions, into a more manageable set of tags. The reader, in probabilistic mechanisms and its variants [4], communicates the frame length and each tag picks a particular slot in the frame for its transmission. These deterministic and probabilistic schemes, however, interrogate the tags in a sequential manner. That is, it is only after the successful singulation of a tag that a subsequent singulation takes place for the remaining tags; presenting an inevitable significat delay especially as tags increase in number.

Recently, a parallel singulation (PS) algorithm [5] has been proposed which singulates multiple tags in parallel. The reader's interrogation zone is spatially partitioned into smaller clusters. Clusters are interrogated autonomously, in parallel, resulting in reduced collisions and enhanced reading rates. However, due to the broadcast nature of a wireless medium, tags listen and in some cases may respond to irrelevant requests, i.e., multiple times during a single interrogation cycle. The algorithm also requires sophisticated tag circuity, more of the received RF signal is directed for energy harvesting rather than communication, hence, affecting the tag's reading range. Furthermore, the reader needs to lengthen each query duration to accommodate a number of queries equal to the number of clusters within its interrogation range. In addition, the communication and processing overheads result in higher energy consumption in cluster-heads as well as the reader. As such, it is evident that parallel singulation achieves high reading rates, yet dictates specific requirements and imposes performance limitations.

This paper presents a remedy to these limitations and relaxes requirements without jeopardizing the high data rates of the parallel singulation algorithm. We investigate the communication overhead of tags in addition to the energy consumption of cluster-heads. Based on our findings, we then devise an enhancement for parallel singulation, namely Energy-Efficient Parallel Sigulation ($E^2PS$). The proposed algorithm keeps a low energy profile at the cluster-heads and maintains minimal communication by tags. The ns-2 simulator, extended to simulate RFID systems, is used to study the efficiency of our $E^2PS$ algorithm. The simulation-based evaluation demonstrates significant improvements in terms of energy consumption by tags during the singulation process while achieving high reading rates of the parallel singulation algorithm.

The remainder of this paper is organized as follows. Section II briefly explains the parallel singulation algorithm, presents a formulation of the employed model and an analysis of the energy consumption of tags. Our proposed energy efficient anti-collision algorithm is explained and detailed in III. Section IV elaborates on the simulations environment, performance metrics and evaluation methodology. Finally, section V concludes this work.

## II. PARALLEL SINGULATION

To facilitate a rigorous study and efficient improvement of the parallel singulation algorithm employed in RFID systems, an in-depth study of its main components and inner workings is imperative. As such, this section provides an overview of the parallel singulation algorithm and presents an analysis of

978-1-4244-6404-3/10/$26.00 ©2010 IEEE

its hurdles in the context of communication overhead, energy consumption induced by cluster-heads and querying latency.

### A. Model

Consider an interrogation area $A_R$ covered by a single RFID reader $R$ with maximum interrogation range of $R_r$. The interrogation area is divided into $n$ clusters $c_1, ..., c_n$ of area $A_{c_1}$,..., $A_{c_n}$, respectively. Each cluster is controlled by a cluster-head. The cluster head, a physical device [6], selectively relays the received tags' serial numbers to the reader. Assuming an omni-directional disk model, the reader's interrogation area is completely covered by the clusters, i.e., $A_R \leq (A_{c_1} \bigcup ... \bigcup A_{c_n})$. Assuming a uniform distribution, there are $k$ tags within the interrogation area $A_R$, i.e., $t_1$,..., $t_k$, where $k >> n$. Each cluster $i$ receives responses only from tags within its area $A_{c_i}$. The reader during each interrogation cycle $i$ broadcasts singulation requests $REQA_{(i,j)}$ intended for the cluster $c_j$. For an interrogation cycle, the maximum number of singulation requests transmitted by the reader equals the number of clusters.

### B. Singulation Algorithm

In RFID systems operating with parallel singulation, the reader's interrogation zone is divided into multiple non-interfering clusters. This division reduces the collision problem into intra-cluster collisions, to be resolved autonomously and in parallel. The reader maintains, for each cluster, an independent instance of the collision tree and its associated data structures. An instance of the anti-collision algorithm runs for every cluster, and is iteratively executed for each interrogation cycle. In a collision scenario between the tags of the $k^{th}$ cluster, a new broadcast request $REQA_{(i,k)}$ is formed by replacing the most significant collision bit by $0$ followed by trailing 1's. The broadcast request for the $k^{th}$ cluster is then used for the subsequent interrogation cycle. The collision bit is also replaced by 1, followed by trailing 1's, and is queued on $Queue_k$, which is de-queued under two scenarios. First, when no response from the tags was received for the last request and second, after successful tag singulation. The algorithm terminates when all the cluster-heads seize to receive tag responses.

### C. Analysis of Communication overhead

The parallel singulation algorithm regulates a query to be sent for each cluster simultaneously. However, a broadcasted reader request intended for its cluster reaches all tags within its interrogation range; resulting in responses from tags of other clusters. For instance, consider two clusters $a$ and $b$ with their respective request as $REQA_{(i,k)}$ and $REQA_{(i,j)}$. If $REQA_{(i,k)} \leq REQA_{(i,j)}$ then a subset of tags from cluster $a$ may respond even if their serial number is greater than $REQA_{(i,j)}$, thus resulting in significant communication overhead. The number of clusters within the interrogation zone of the reader dictates the number of tag responses. The higher the number of clusters, the more the queries sent by the reader, thus the larger the number of tag responses. In a typical configuration, e.g. four or sixteen clusters, the average number of tag responses can double or quadruple, respectively, when compared to a conventional sequential algorithm such as the binary search algorithm [1].

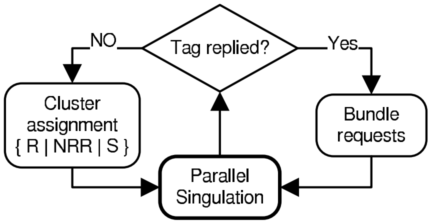


Fig. 1. Flow chart of the $E^2PS$ algorithm

### D. Analysis of Energy Consumption

Cluster-heads, in the parallel singulation approach, are responsible for receiving, processing and relaying valid data (serial numbers) to the reader. The cluster-heads, akin to wireless sensor networks, are battery operated devices [6], thus it is important to reduce their energy consumption. However, because of the exceptionally large number of tag responses, the cluster-head batteries are depleted much sooner, hence limiting system operational capability.

## III. ENERGY-EFFICIENT PARALLEL SINGULATION ($E^2PS$)

As previously highlighted, the clusters of the parallel singulation scheme result in extra communication overhead induced by tags and increased energy consumption at cluster-heads. The communication overhead, and hence significant energy consumption, is attributed to three main reasons. First, the impossibility of associating tags to a particular cluster *a priori*. Second, the tags inability to identify, amongst all the broadcast queries, which correspond to the clusters it physically lies within. Furthermore, the limited functionality available to passive tags constrains the employment of any sophisticated cluster prediction and assignment algorithms.

To mitigate the situation, we propose an enhancement to the parallel singulation [5], namely the Energy-Efficient Parallel Singulation ($E^2PS$) algorithm. The extension includes novel variations of the cluster assignment algorithm, bundling of reader queries and the algorithm termination process. The overall flowchart of the $E^2PS$ algorithm is shown in Fig. 1 and detailed hereafter.

### A. Modified Singulation Algorithm

The pseudocode of the proposed $E^2PS$ algorithm is presented as Algorithm 1. The algorithm starts with the cluster assignment mechanism (lines 4-9). The reader broadcasts a special command *SELECT-CLUSTER*, which contains the total number of clusters within its interrogation zone, triggering the tags to reset their cluster affiliation (line 4). The cluster affiliation is used by the tag to ignore all queries except the one targeted for its chosen cluster. Afterwards, the reader sets the clusters state to *FORWARD* (line 5) as it broadcasts the initial serial request (REQA) command using the highest possible

32-bit serial number as its parameter (line 6). This initial interrogation request serves as the root of the singulation tree. The algorithm terminates if there is no response to the initial request (lines 7-9).

**Algorithm 1** Energy-efficient parallel singulation algorithm

```
1:  Send (broadcast, RESET)
2:  repeat
3:     // ClusterID assignment:
4:     Send (broadcast, SELECT-CLUSTER)
5:     Set all cluster's state $s_x$ to FORWARD
6:     Send (broadcast, $REQA_{0xFFFFFFFF}$)
7:     if for all clusters $c_k$ $\neg\exists(reply)_{c_k}$ then
8:        terminate
9:     end if
10:
11:    repeat
12:       // Parallel singulation:
13:       for all cluster $c_k$ do
14:          if $(collision)_{c_k} || \neg\exists(reply)_{c_k}$ then
15:             $REQA_{(i,k)} \leftarrow$ next query, if any
16:          else if $\neg(collision)_{c_k}$ then
17:             Tag is singulated, read/write data to/from tag
18:             $REQA_{(i,k)} \leftarrow$ next query, if any
19:          end if
20:       end for
21:       // Bundle requests:
22:       for all cluster $c_k$ do
23:          if $REQA_{(i,k)}$ = NULL then
24:             $Send(c_k, SILENT)$
25:             $s_k \leftarrow$ SILENT
26:          end if
27:       end for
28:       Send(broadcast, $REQA_{(i,k)}$) // for REQAs $\neq$ NULL
29:    until $\exists k | s_k$ = FORWARD
30: until false //forever
```

The reader maintains, for each cluster, an independent instance of the singulation tree and its associated data structures, e.g., singulation tree, query queue, etc. An instance of an anti-collision algorithm is iteratively executed during each cluster's interrogation cycle (lines 12-20). That algorithm is used to formulate new queries (lines 14-19) in the event of successful tag singulation, empty interrogation cycle and tag collisions. Subsequent interrogation queries are both broadcasted and sent to the targeted cluster-heads. Meanwhile, the clusters with empty (NULL) queries are put to sleep, i.e., *SILENT* state (lines 23-26). The parallel singulation process of the $E^2PS$ algorithm is terminated when a cluster switches to the *FORWARD* state (line 29).

The cluster assignment process is repeated to handle cases of tags miscalculating their cluster association. The algorithm, i.e., the cluster assignment procedure, parallel singulation and bundle request, is repeated indefinitely (line 30) until no response is made to the reader's initial request. This handles the likely scenario of tags associated with incorrect clusters. This is inherent from the lack of technical sophistication confining the cluster assignment schemes, as explained next.

### *B. Cluster Assignment Schemes*

Due to the limitations of functionality in passive RFID tags, we propose three efficient and light-weight cluster assignment schemes, for the $E^2PS$ algorithm; namely, random, sequential and memory-based.

*1) Random:* In the random-based cluster assignment approach, namely $E^2PS\text{-}R$, the tag picks randomly, using its native RNG, a number in [1,$n$] . The total number of clusters $n$ is broadcasted by the reader (Algorithm 1 line 4). The random-based approach is a simple memoryless scheme. However, if large numbers of clusters are supported in the system, during every iteration, each has an equal probability of being picked regardless of previous history. This may results in extra interrogation cycles hence reading delay.

*2) Non-Repetitive Random :* The non-repetitive random based approach, namely $E^2PS\text{-}NRR$, is similar to the random approach, i.e., the cluster is picked randomly. However, using historical data, a cluster is discarded if it was picked at a previous cycle. The non-repetitive random cluster assignment scheme proves effective as the system faces increased number of clusters, at the expense of history-keeping. At a small scale, however, the non-repetitive random scheme would perform no better, in terms of correct cluster assignment than the random-based scheme.

*3) Sequential :* In the sequential approach ($E^2PS\text{-}S$) the tag selects a cluster starting from 1 till $n$ in sequence. The sequential scheme is useful in a small scale setting, i.e., minimal clusters and sparse tag environments.

### *C. Delay Analysis*

As part of our evaluation, a detailed running time analysis is presented. This analysis presents a caliber to the performance of the parallel singulation scheme as an analysis of the average case scenario. Accordingly, previous algorithms and future improvements could be held against this caliber and scaled to determine their efficiency.

Let $f(k)$ be the number of cycles required to read a set of $k$ tags, and let $E[f(k)]$ be the expected value for $f(k)$. The scheme [7] utilized in this paper has $f(n) = 2n - 1$ cycles. The main factor at play is the number of tags in a given cluster. Intuitively, as the number of tags increase in $A_{c_i}$, the potential for collisions increases as more tags attempt to access the medium simultaneously. In fact, the relation is on average superlinear. We initially assume a uniform distribution of tags over the reader's interrogation region $A_R$. That is, the number of tags $f(t_i)$ in a given a cluster $A_{c_i}$ is estimated by $E[f(t_i)]$:

$$E[f(t_i)] = \frac{A_{c_i} * k}{A_R} \tag{1}$$

Nevertheless, as we investigate the worst case scenario, this assumption will be waived to accommodate for cases where nodes clutter up a smaller set of clusters, or present an un-even distribution.

Denoting the estimated number of tags in clusters $c_1$,...,$c_n$ as $E[t_1]$,...,$E[t_k]$, we obtain a measure for the overall number of cycles exectued by the algoirthm as the maximum number of cycles amongst all clusters. Thus, the singulation process takes $C(n)$ cycles:

$$C(n) = Max\left[C(E[t_1]), C(E[t_2]), ..., C(E[t_k])\right] \tag{2}$$

Since this is only a measure of the number of cycles, we still need to take into account the number of iterations executed by each tag, on average, to finally select it's correct cluster-head. This highly depends on the scheme used for cluster assignment, and is detailed in the following subsections.

*1) Random cluster assignment:* The probability of a tag selecting an incorrect cluster at any given iteration is $(n-1)/n$. Hence, the expected number of tags incorrectly choosing their clusters thus needing to iterate again, $E[t_{iterate}]$ is given by:

$$E[t_{iterate}] = \frac{k*(n-1)}{n} \tag{3}$$

Summing over all iterations, and deducting those tags which correctly choose a cluster, we obtain the expected number of repeats as:

$$E[repeats_R] = \sum_{j=1}^{n} k*\frac{(n-1)}{n} - j*\frac{k}{n} \tag{4}$$

*2) Memory-based cluster assignment:* Since this cluster assignment scheme assumes prior knowledge of incorrect cluster numbers from previous iterations, the expected number of repeats, not allowing repetitions, is dervied as:

$$E[repeats_{NRR}] = \sum_{j=1}^{n} k*\frac{(n-j-1)}{n-j} - j*\frac{k}{n} \tag{5}$$

*3) Sequential cluster assignment:* Given the sequential nature of this approach, and its exhaustive parsing through all possible clusters in order, it is straight forward to derive the expected number of repeats as:

$$E[repeats_S] = \sum_{j=1}^{n} k - j*f(t) \tag{6}$$

In the worst case scenario, where all the tags reside in a single cluster, only $E[repeats_S]$ of the $E^2PS$-$S$ algorithm would converge to $n$ / 2. We cocnlude our analysis to derive a measure for the total number of cycles, i.e., $C(n)$ in (2) is given by (3), (4) and (5) for the $E^2PS$-$R$, $E^2PS$-$NRR$ and $E^2PS$-$S$ schemes, respectively.

## IV. Performance evaluation

In this section we analyze the performance of the $E^2PS$ algorithm, with the proposed cluster assignment schemes, against the parallel singulation ($PS$) algorithm. In our evaluation, we caliber the communication overhead (number of tag responses to cluster-heads) , energy consumption at the cluster-heads named as the tag traffic rate, total cycles based on their tag reading rates, and the devised singulation efficiency metric.

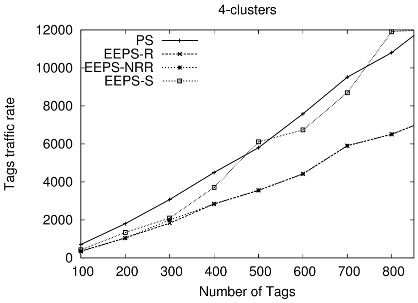


Fig. 2. Average number of tag responses in a 4 cluster configuration

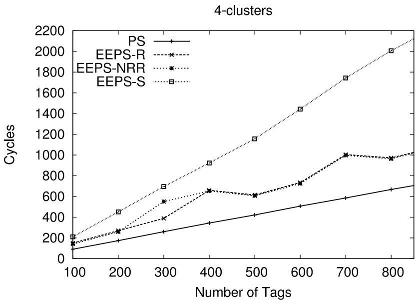


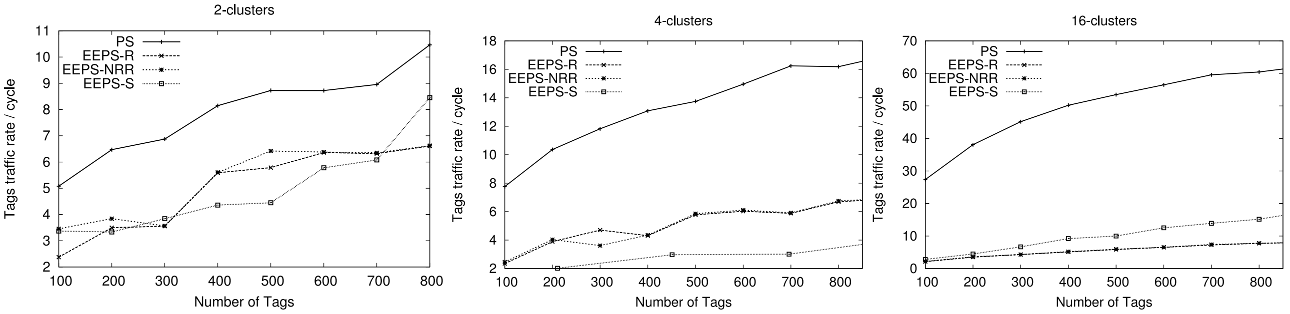
Fig. 3. Total reading cycles for various tags enumeration

### A. Simulation methodology

We have extended the ns-2 network simulator to implement RFID system. The major extension involves modifying the underlying simulator architecture to support core RFID functionality, the EPC class1-gen2 MAC protocol, and non-EPC singulation protocols from the literature. Minor changes to the simulator include tailoring the network node to serve as an RFID tag, a reader, and a cluster-head, the single-hop communication model (backscattering modulation) between the tag, cluster-header and reader, and so forth. Simulations are performed using the following common parameters, and variations are stated in their respective sections. In the setup, tags are uniformly distributed on a $20 \times 20m$ grid. A single reader is located at the centre of the grid with an interrogation range of $15m$. The cluster-heads are deployed in a grid manner. The tags randomly choose a 96-bit serial number and a 32-bit serial number, the latter used in singulation. Simulations are terminated when all the tags are successfully identified. The performance metrics are averaged over twenty different topology runs generated using different random seeds.

### B. Tags traffic rate

Tags traffic rate is defined as the number of times tags transmitting their serial number (singulation ID) in response to received queries. The average, for various set of tags and four cluster configuration, is shown in Fig. 2. In the case of the $PS$ algorithm, the tag may reply to queries of other clusters, therefore, the tags traffic rate for $PS$ is significantly high and increases as the number of clusters increase. However, in a configuration with few clusters (e.g. two) the average number



(a) Two clusters (b) Four clusters (c) Sixteen clusters

Fig. 4. Overall system efficiency for numerous set of tags

of tag responses roughly equal that of the $E^2PS$ scheme. The $E^2PS$ algorithm, with various cluster assignment schemes, are labelled as $EEPS$-$R$, $EEPS$-$NRR$ and $EEPS$-$S$ in the same figures. Both $E^2PS$-$R$ and $E^2PS$-$NRR$ algorithms have the lowest number of tag responses and remain stable regardless of the number of clusters or the tags. On the contrary, the $E^2PS$-$S$ scheme performance noticeably degrades with the increase in number of clusters. This is caused by, in the worst case, the tags of the highest numbered cluster. For example, the tags of the sixteenth cluster iterate fifteen times before they pick the right cluster assignment. The $E^2PS$-$S$ algorithm is simple and effective, when the system supports few clusters, whereas $E^2PS$-$R$ algorithm is more effective in large cluster configurations.

### *C. Reading rate*

The tag reading rate is a product of the total number of interrogation cycles and the individual cycle intervals. High reading rates imply that more tags could be singulated by a reader. The total singulation (reading) cycles for various tag enumerations, using the $PS$ algorithm and the variations of the $E^2PS$ algorithm, with configurations of four clusters, are plotted in Fig. 3. The $E^2PS$ schemes demonstrate relatively high reading cycles when compared to the conventional approach. However, with an exception of the $E^2PS$-$S$ scheme, both $E^2PS$-$NRR$ and $E^2PS$-$R$ schemes are within marginal difference, less then 10% of the $PS$ approach. Hence, the energy-efficient parallel singulation algorithm, using either the random or non-repetitive random cluster assignment schemes, maintain high reading rates of parallel singulation. However, they do so without incurring communication overhead leading to high energy consumption by cluster-heads.

### *D. Singulation efficiency*

We define the *singulation efficiency* metric to caliber the overall improvement of the $PS$ and $E^2PS$ schemes - the latter using different cluster assignment schemes - as the number of tag responses per singulation cycle (tag traffic rate per singulation cycle). It measures the improvement in communication overhead and energy consumption with respect to reading rates. The overall measured efficiency for various proposed algorithms and parallel singulation schemes are depicetd in Fig. 4. All of the proposed algorithms outperform $PS$. Significant improvements are observed under dense tag environment and large number of clusters (Fig. 4-c), where the overall efficiency of both $E^2PS$-$R$ and $E^2PS$-$NRR$ are five times higher than that of the conventional $PS$ algorithm.

## V. CONCLUSIONS

The realm of RFID systems has great potential in expanding into major applications benefiting all aspects of our lives. RFID protocols witnessed a major milestone as the parallel singulation scheme was proposed to introduce parallelism in interrogation and extended operation through designated cluster-heads. However, this scheme faces limitations as the number of clusters supported by the system increases. This is mainly due to elevated overhead in communication and many redundant messages exhausting the power of cluster-heads. This paper presented an enhancement to the parallel singulation scheme, namely $E^2PS$, with varying methods of cluster selection, to reduce redundancy in communication and hence conserve on power spent. The simulation results demonstrated the significant gains in system efficiency without affecting tags operations nor interrogation range.

## REFERENCES

[1] K. Finkenzeller, *RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification*. New York, NY, USA: John Wiley & Sons, Inc., 2003.

[2] D.-H. Shih, P.-L. Sun, D. C. Yen, and S.-M. Huang, “Taxonomy and survey of rfid anti-collision protocols,” *Computer Communications*, vol. 11, no. 29, pp. 2150–2166, Jan. 2006.

[3] J. Myung, W. Lee, J. Srivastava, and T. K. Shih, “Tag-splitting: Adaptive collision arbitration protocols for rfid tag identification,” *IEEE Transaction on Parallel and Distributed System*, vol. 18, no. 6, pp. 763–775, 2007.

[4] C. Law, K. Lee, and K.-Y. Siu, “Efficient memoryless protocol for tag identification,” in *Proceedings of the $4^{th}$ International Workshop on Discret e Algorithms and Methods for Mobile Computing and Communications*, 2000, pp. 75–84.

[5] K. Ali and H. Hassanein, “Parallel singulation in rfid systems,” in *Proceedings of IEEE Global Communication (Globecom)*, Dec. 2009, to appear.

[6] K. Ali and H. S. Hassanein, “Distributed receiving in rfid systems,” in *Proceedings of the 34th IEEE Conference on Local Computer Networks*, Oct. 2009, pp. 69–76.

[7] 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 Electromagnetic Research*, vol. 4, pp. 263–271, 2008.