Parallel Singulation in RFID Systems

Kashif Ali and Hossam Hassanein
School of Computing, Queen's University
Kingston, Ontario, Canada, K7L 3N6
{kashif, hossam}@cs.queensu.ca

***Abstract*—Tag collisions can impose a major delay in Radio Frequency IDentification (RFID) systems. Such collisions are hard to overcome with passive tags due to their limited capabilities. In this paper, we look into the problem of minimizing the time required to read a set of passive tags. We propose the novel concept of parallel singulation which, aided by the multiple antennas configuration, interrogates tags in sets. Multiple instances of the anti-collision scheme are executed for each set of tags in parallel and in an autonomous manner. Simulation results show that our approach makes significant improvements in reducing collisions and interrogation delay and thus increasing the reading rate and throughput of RFID systems.**

I. Introduction

A typical RFID system consists of a number of radio frequency tags and an electromagnetic reader. The reader has an interrogation zone; the strength of the electromagnetic waves, generated by the reader, is able to power the tags in its interrogation zone, the reader can receive and decode the signal sent by any tag in its interrogation zone — a process known as singulation. At any arbitrary instant, a reader can only singulate one tag within its interrogation zone. Tags within the interrogation zone of a reader, and readers with overlapping interrogation zones, may simultaneously attempt to access the wireless medium for data communication.

The simultaneous wireless medium access, and hence collisions, undermines the overall performance of the RFID system. To sustain system operability, efficient mechanisms for medium access are required. Collisions in RFID can be of two types, reader collisions and tag collisions. Reader collision is solvable using the conventional medium access control techniques, as a reader can detect collision and communicate with other interfering readers. Tag collisions are problematic as a tag has limited power and functionalities. The passive tag can only transmit data by reflecting the reader transmitted electromagnetic waves and, hence, cannot detect nor communicate with the neighboring tags.

Several schemes have been proposed to deal with tags' collisions in RFID systems and are classified as either deterministic or probabilistic approaches [1] [2]. Deterministic mechanisms are essentially tree-based anti-collision algorithms [3]. The tree-based approach such as the binary search tree algorithm and its variants [1], [4], [5] splits the tags based on the previous interrogation cycle colliding bits, and their respective positions, into a more manageable set of tags. Furthermore, assorted aiding techniques, for example the statistical approach [6] which exploits the information stored in previous interrogation cycles to assist in forthcoming cycles and the energy-aware approach [7] which minimizes unnecessary processing and communication overhead by the MAC protocols, have been investigated to enhance the deterministic schemes. The reader, in the probabilistic mechanisms, communicates the frame length and the tag picks a particular slot in the frame for its transmission. The reader repeats this process until all tags have been successfully transmitted at least once. The frame size may be adjusted, based on the collision, idle and occupied frame information from a previous interrogation cycle, for the subsequent cycle [1]. The framed slotted Aloha, with its static and dynamic variants [8]–[10], employs the probabilistic mechanisms. However, the existing deterministic and probabilistic schemes interrogate the tags in a sequential manner. In other words, it is only after successful singulation of a tag that the subsequent singulation takes place for the remaining tags.

In this paper, we propose and evaluate a novel parallel singulation algorithm which singulates multiple tags in a non-sequential manner. The basic idea is that a significant amount of collisions can be avoided by spatially partitioning the interrogation zone, using multiple antennas configuration or smart relay devices [11], into micro-zones or clusters. Clusters are then autonomously interrogated in parallel. The proposed anti-collision algorithm reduces the overall tags' collisions and, hence, results in a faster singulation, owing to two underlying intuitions. First, with high probability, the number of tags within each cluster is smaller than the total number of tags within the reader's interrogation zone. And second, due to the autonomous nature of these clusters, only intra-cluster tag collisions are possible. The ns-2 simulator [12], extended to simulate the RFID systems, is used to study the efficacy of the parallel singulation algorithm. Our simulation results show significant improvements in terms of reading rates, collisions and delays for the parallel singulation algorithm over the existing anti-collision schemes.

The rest of the paper is organized as follows. Section II explains the parallel singulation algorithm and analyze its running complexity. Section III provides detail information about the simulations environment, performance metrics and evaluation methodology. Finally, section IV concludes this work.

II. Parallel Singulation Algorithm

Collisions happen when there is more than one tag within the reader's interrogation zone and the reflected signal from a tag is strong enough to interfere with others. To alleviate this

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

situation, the concept of distributed receiving-based communication architecture is proposed in [11]. In such architecture, an additional system component is introduced, termed as fielders or cluster-head, as depicted in Fig. 1. The RFID reader and cluster-heads dynamically adjust the tags' reflectivity coefficient and receiving signal threshold, respectively, to form micro-interrogation zones or clusters within the reader interrogation zone. The basic philosophy is to cluster the tags into micro-zones and then interrogate these clusters as it would reduce the number of tags that needs to be singulated at one time and hence, improve system throughput.

Existing anti-collision schemes are based on certain assumptions about the underlying system. First, tags within the interrogation zone are interrogated by the reader only in a sequential manner. In other words, it is only after successful singulation of a tag that the subsequent singulation takes place for the remaining tags. And second, all tags that lie within the interrogation zone will cause collisions at the reader. On the contrary, RFID systems based on distributed receiving have the following important characteristics. First, the interrogation zones are divided into multiple non-interfering micro-zones. Second, the reader only knows and is concerned with the intra-zone collisions. Finally, and most importantly, due to the micro-zones, the sequential singulation is not the only option. To elaborate, since only the intra-zone collisions are possible, each micro-zone can be interrogated independently.

To this end, we propose a parallel singulation algorithm where multiple instances of an existing anti-collision scheme are executed, for each micro-zone, in parallel and autonomous manner. We use a simple binary search tree anti-collision algorithm [1] [13] yet any other anti-collision algorithm can be used. In this section, we describe the system model, explain the parallel singulation algorithm and analyze its running complexity.

A. System Model

Consider an interrogation area A_R enclosed 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 micro-zone is managed by a cluster-head. The cluster head, a physical device [11], selectively relays the received tags' serial numbers to the reader. Assuming the disk model, the reader 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 the 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 micro-zone.

B. The algorithm

The main goal of RFID anti-collision algorithms is to increase the singulation rate by reducing the tags' collisions. In the case of parallel singulation algorithm, this is achieved in two ways. First, the execution of an instance of the singulation algorithm (conventional binary search scheme is used in this paper), for each micro-zone, in an autonomous manner. And second, synchronization between the reader and the cluster-heads. Pseudocode for the parallel singulation algorithm is shown in Algorithm 1. The parallel singulation algorithm is composed for of four logical components: initialization, concurrent singulation, communication and timeout calculations.

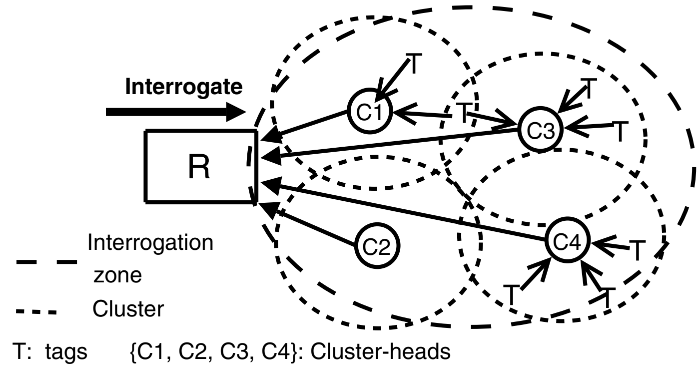


Fig. 1. Clustering concept of the parallel singulation scheme

1) Initialization: The algorithm begins with housekeeping tasks (lines 1-3) which includes setting all clusters states to *FORWARD*, broadcasting the RESET and the initial serial request (REQA) command. Clusters in the forward state always relay tags' serial numbers to the reader whereas converse holds in silent state. The reader, after resetting tags internal state (RESET), broadcasts the initial serial request (REQA) command using the highest possible 32-bit serial number as its parameter. This initial interrogation request serves as the root of the singulation tree.

2) Parallelism: For each cluster, the reader maintains an independent instance of the singulation tree and its associated data structures, i.e., singulation tree and queue. An instance of the anti-collision algorithm (listed in lines 16-28) is iteratively executed during each cluster's 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 (line 12). The most significant collision bit is also replaced by 1, followed by trailing 1's, and is queued on $Queue_k$ (line 13) which is de-queued under two scenarios. First, when no response from the tags was received for the last request (line 16) and second, after a successful tag singulation (line 19).

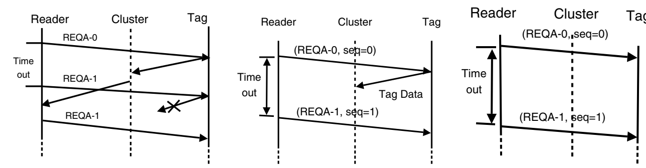
3) Reader and cluster-heads communication: The tag replies to a reader request if and only if the requested serial number is greater than its own. Due to broadcasting nature, the tag listens and in some cases may respond to an irrelevant request, i.e., is not intended for its cluster. For instance, consider two clusters a and b with their respective request

Algorithm 1 Parallel singulation algorithm

1: Set all cluster's state s_x to **FORWARD**
2: Send (broadcast, RESET)
3: Send (broadcast, $REQA_{0xffffffff}$)
4:
5: // Singulation process for the i^{th} interrogation cycle:
6: **repeat**
7:   **wait** until $REQA_{timeout}$
8:
9:   **for all** cluster c_k **do**
10:     // Let j be the most significant collision bit, if any.
11:     **if** $(Collision)_{c_k}$ **then**
12:       $REQA_{(i,k)} \leftarrow b_{n-1}...b_j = 0b_{j-1} = 1...b_0 = 1$
13:       $Queue_k \leftarrow b_{n-1}...b_j = 1b_{j-1} = 1...b_0 = 1$
14:     **else if** $(no - reply)_{c_k}$ **then**
15:       // NULL is returned, if the $Queue_k$ is empty
16:       $REQA_{(i,k)} \leftarrow$ Pop from $Queue_k$
17:     **else if** $(no - collision)_{c_k}$ **then**
18:       // Tag is successfully singulated
19:       // Read data, Write data, UnSelect, etc.
20:       $REQA_{(i,k)} \leftarrow$ Pop from $Queue_k$
21:     **end if**
22:   **end for**
23:   **for all** cluster c_k **do**
24:     **if** $REQA_{(i,k)} \neq$ NULL and $REQA_{(i,k)}$ is Unique for i^{th} cycle **then**
25:       $Send(c_k, REQA_{(i,k)})$
26:     **else if** $REQA_{(i,k)}$ = NULL **then**
27:       $Send(c_k, SILENT)$
28:       $s_k \leftarrow$ **SILENT**
29:     **end if**
30:   **end for**
31: **until** $\exists k | s_k$ = **FORWARD**

as $REQA_{(i,k)}$ and $REQA_{(i,j)}$. If $REQA_{(i,k)} \leq REQA_{(i,j)}$ then a subset of tags from the cluster a may response even if their serial number is greater than $REQA_{(i,j)}$. Such a situation potentially inhabits the cluster singulation tree with unwanted tree nodes and, hence, an unpredictable outcome. To alleviate such a data structure corruption, the cluster-heads are kept informed, by the reader, of their successive interrogation requests. This facilitates in effective filtering (listed in lines 22-29) of the irrelevant tags' replies.

4) Timeout: Various scenario trigger timeouts and retransmissions and are outlined in Fig. 2. The $REQA_{timeout}$ time (line 7) is the duration of an interrogation cycle, i.e., the maximum time the reader waits for a tag response. The timeout can be non-legitimate or legitimate. The non-legitimate case is when the reader, upon time expiration, broadcasts the queued request (line 16). However, shortly after that, it receives the delayed tag response (Fig. 2-a). In such a case, the reader cancels the current interrogation cycle, queues the transmitted request and continues with the singulation process based on the delayed responses. The handling of the delayed transmission



(a) Redundant cycle (b) Filtering of tag data (c) No Tag data

Fig. 2. Various timeout scenarios in parallel singulation

is critical as, by neglecting such cases, we may potentially overlook some tags. The legitimate case happens when first, the cluster-head filters out all of the tags' responses (Fig. 2-b) and second, when an empty request exists, i.e., no tag matches with the request (Fig. 2-c). Both scenarios lead to a graceful timeout followed by broadcasting of the successive interrogation request (lines 17-19).

C. Delay Analysis

In this section, we analyze the running complexity, i.e., average, best and worst singulation delay, of the proposed parallel singulation algorithm. 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 $f(k)$. For the binary tree protocol [1], in general:

$$f(n) = c\, n - 1, \tag{1}$$

where c is a constant and n is total number of tags.
In the parallel singulation algorithm, the reader interrogates the micro-zones' tags in parallel. Let $E[t_1]$,...,$E[t_k]$ be the estimated number of tags within the clusters c_1,...,c_k, respectively. Then, the overall number of cycles is the maximum number of cycles, amongst all micro-zones, the singulation process takes

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

To determine the best, average and worst case scenarios, based on (2), it suffices to find an estimate of minimum, maximum and average of the number of tags in the micro-zones.

1) Best Case: The best case scenario is when all the tags are evenly distributed amongst the clusters

$$E[t_i] = \frac{A_i * k}{A_R}; for\ any\ given\ cluster\ i. \tag{3}$$

2) Worst Case: The worst case scenario is when all the tags are within a single cluster i

$$E[t_i] = k \;\; and \;\; t_j = 0,\ (\forall i | i \neq j) \tag{4}$$

This corresponds to the case of the existing approaches, i.e., sequential singulation algorithms. Hence, the parallel singulation algorithm cannot perform any worse than existing algorithms.

3) Average Case: The cluster, in an attempt to cover the reader's interrogation zone, may overlap. In other words, there is a possibility that a tag may belong to multiple clusters and

[1]The scheme [13] utilized in this paper has $f(n) = 2\, n - 1$

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

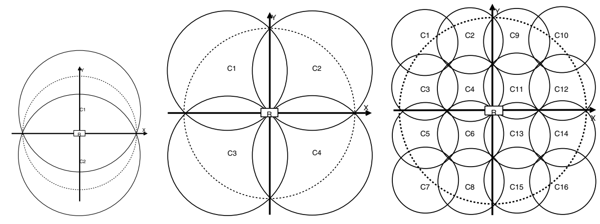
hence, is counted by than one cluster. The area covered by an overlapping region of the two clusters $A_{overlap}$, each with equal radius, i.e., $R_{c1} = R_{c2} = R_c$, is formulated using simple geometry as

$$A_{overlap} = 2R_c^2 \cos^{-1}\left[\frac{d}{2R_c}\right] - \frac{1}{2}d\sqrt{4R_c^2 - d^2} \qquad (5)$$

and therefore, an estimate of the number of tags is

$$E[t_i] = \frac{N}{A_R}\left[1 - \frac{A_{overlap}}{O}\right] P_i \qquad (6)$$

Here O is the count of the cluster overlapping regions and P_i is the probability with which the cluster c_i singulate the overlapping tags. A common value of P_i, assuming two overlapping cluster, is 0.5 as each cluster has equal probability of tags' singulations within the overlapping regions. That is, each cluster has equal probability to singulate the overlapping tags. In a typical grid-based deployment (Fig. 3), since each entity has at-most four neighbors, the cluster is at-most overlapped by four others, i.e., $O = 4$.



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

Fig. 3. Grid-based cluster formation

III. Performance evaluation of the parallel singulation scheme

In this section, we analyze the performance of the conventional tree-based and parallel singulation schemes. As evaluation, we use total cycles, request collisions, bit collisions, idle singulation cycles and communication overhead as performance metrics.

A. Simulation Methodology

We have extended the ns-2 network simulator to support RFID. The major extension involves modifying the underlying simulator architecture to support the basic RFID functionality, the EPC class1-gen2 MAC protocol, and non-EPC singulation protocols from the literature. Minor changes to the simulator include inheriting of the network node to serve as an RFID tag, a reader, and 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 parameters, unless otherwise mentioned. In the setup, tags are uniformly distributed in a grid of $20 \times 20m$. A single reader is located at the center of the grid with an interrogation range of $15m$. The cluster-heads are deployed on a grid, as depicted in Fig. 3. The tags randomly choose a 96-bit serial number and a 32-bit serial number, the latter of which is used in singulation. Simulations are run until all the tags are successfully identified. The performance metrics are averaged over twenty different topology runs generated using different random seeds.

B. Reading Rate

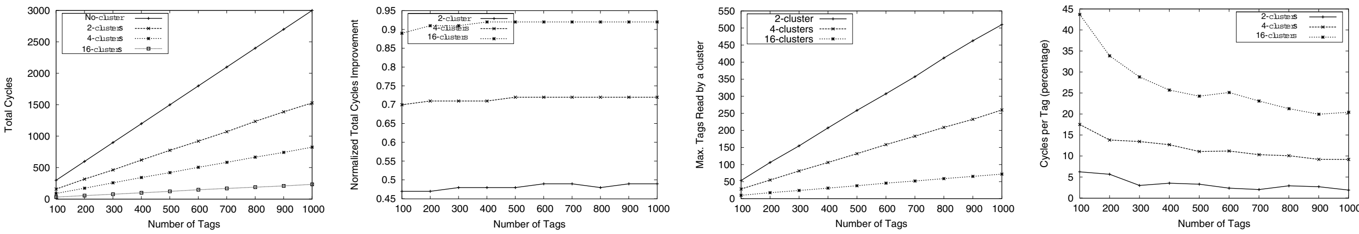
The tag reading rate is a product of the total number of interrogation cycles and the individual cycle intervals. High reading rates imply that more tags can be singulated by a reader. The total singulation (reading) cycles, normalized cycle improvements and the average singulation cycles per tag for various tag enumerations, using the conventional and the parallel singulation schemes, with configurations of 2, 4 and 16 clusters, are shown in Fig. 4. The number of singulation cycles for the proposed approach is many-fold lower in comparison to the number of cycles required by the conventional approach. For instance, using 2, 4 and 16 clusters, our approach decreases the total reading cycles by 45%, 70% and well over 90%, respectively (Fig 4-b). Furthermore, the total reading cycle, when using the 16-clusters configuration, is almost linear as is shown in Fig 4-a. Hence, the RFID system based on the parallel singulation has the potential to support higher volumes and dense tag distributions.

Due to the nature of parallel singulation, the cluster with the highest tag enumeration determines the total number of singulation cycles. As an illustration, Fig. 4-c shows the maximum number of tags singulated, among all clusters, using 2, 4 and 16 clusters configuration. The results show, as expected, that more clusters yield high reading rates since each interrogated set of tags, within each cluster, are obviously fewer than those within the whole interrogation zone. For instance, with a total of 1000 tags in the reader's interrogation zone, using a configuration of 16-clusters, the maximum tags read by any given cluster are always less then 100. However, under a similar setting, using a configuration of 2-clusters, the numbers are almost five to six times higher. In short, the 2-cluster configuration requires more cycles thus low reading rates in comparison with the 16-clusters configuration. On the other hand, an increase in the number of clusters also increases the probability of the overlapping regions. Overlapping imbalances the tag distribution and creates idle cycles. Fig. 4-d shows the average cycles per tag normalized to the best case (eq. 3). As the ratio of the overlapping area of the 16-clusters configuration is high, so is its divergence, e.g., by 25% from its best case. However, the proposed scheme diverges less for the fewer cluster configuration. For instance, divergence of only 3% is observed for the 2-clusters configuration. To summarize, there exists a trade off between the reading rate and the cluster configurations.

C. Collision

A collision is defined at the bit level and/or at the response level. The bit level measures the number of colliding bits at the reader and the response level measures number of colliding responses. The importance of each type depends on the anti-collision scheme being used. For instance, it is more important

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.



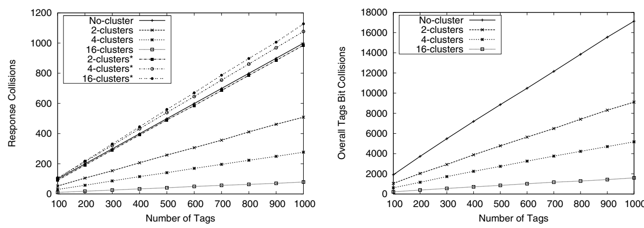
(a) Total Cycles

(b) Normalized Cycles

(c) Maximum tags read amongst the clusters

(d) Normalized Cycles

Fig. 4. Total cycles, average cycles per tag and normalized cycles comparison between conventional and distributed RFID system



(a) Request Collisions

(b) Bit Collisions

Fig. 5. Improvements in the request collisions and the bit collisions (accumulative collisions, in the key box are shown with trailing '$*$')

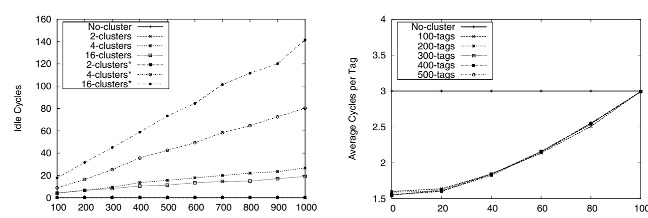
to reduce the bit-level collisions in a query tree [14], which is a bit-by-bit approach, than it is to the binary search tree [1]. The response level collisions and the bit-level collisions, for the conventional and the proposed algorithms, are shown in Fig. 5. In the case of the parallel singulation, the overall collisions is the maximum number of collisions (bit-level or request-level) that were seen among the clusters. The cluster facilitates spatial isolation of the tags and only has the intra-cluster collisions. For this reason, the response level collisions and the bit-level collisions, for the parallel singulation, show immense improvements. For instance, using the 4-clusters configuration reduces the bit-level and response-level collisions by more than 70%. In short, more the clusters fewer are the collisions.

The reader, during each interrogation cycle, has to process the serial number requests for each cluster. The fewer the request collisions the fewer the reader requests that need to be sent. The accumulated collisions for the parallel approach are close to the conventional singulation approach (Fig. 5-a). With fewer clusters, the number of accumulated response collisions is marginally better than the conventional approach. However, it gets worse with an increase in the number of clusters. For instance, for the 4-clusters and the 16-clusters cases, the number of accumulated response collisions for the parallel singulation is respectively, on average, 7% and 10% higher than the conventional singulation. This happens because the percentage of the tags within the overlapping regions increases with an increase in number of clusters. The reader needs to transmit multiple requests for the overlapping tags thus causing more collisions.

D. Idle cycles and effect of clusters overlapping

An idle cycle is an interrogation cycle in which no response from the tags is received at the reader for a legitimate request (scenario of Fig. 2-b and Fig. 2-c). An idle cycle is a result of the empty branches of the singulation tree. The empty branch exists when the tag is put to sleep mode or it moves out of the cluster boundary. Two scenarios may result from such situation. First, the tag is located in the overlapping region of the neighboring clusters and thus was singulated by the reader using the neighboring cluster-heads and second, the mobility of the tag, i.e., it moves in-out of the reader's interrogation zone.

The idle cycles, assuming stationary tags, for the conventional and the parallel singulation techniques are shown in Fig. 6-a. As expected, the idle cycles are not found in the conventional singulation approach but they are noticeably higher in the parallel approach. For instance, 10% of the total cycles, in 16-clusters configuration, are idle cycles. Moreover, the idle cycles increase with the number of clusters and tags as both situations translate to a higher probability of finding the tags within the overlapping regions of the clusters.



(a) Idle cycles

(b) Average cycles per tag (overlapping)

Fig. 6. Idle cycles and overlapping zone effects for conventional and parallel singulation algorithm (accumulative idle cycles, i.e., sum from all clusters, is shown with trailing '$*$' in the key box)

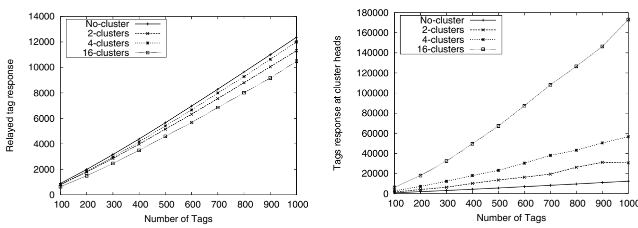
Cluster overlapping introduces redundancy by creating multiple instances of the tag interrogation nodes in the singulation tree of respective clusters. Fig. 6-b shows the effect of two overlapping clusters on the average singulation cycles. With an increase in the overlapping ratio, shown as the percentage of the total cluster area, the average singulation cycle increases exponentially. This increase is independent of the tags enumeration. After successful singulation of a tag, the redundant

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.

nodes at other cluster's singulation trees will translate into idle cycles. The overlapping effect, however, can be reduced by an optimal placement of the cluster-heads.

E. Communication overheads

The parallel singulation algorithm broadcasts singulation requests based on responses from the clusters. As explained, in section II-B3, the filtering mechanism is required at the clusters to drop the non-related replies. Data transfer in the parallel singulation scheme is a two-legged communication first, between the tags and the cluster-heads and second, between the cluster-heads and the reader.



(a) Clusterheads-to-Reader (b) Tags-to-clusterheads

Fig. 7. Communication Overhead

Fig. 7-a shows that the number of accumulated tag responses, as they are relayed by the cluster-heads to the reader, is in harmony with the conventional singulation algorithm. In other words, despite broadcasting multiple requests within an interrogation cycle, the cluster-heads are effective in filtering the unrelated tags' responses. In a system configured with a large number of clusters, the amount of data transfer between the heads and the reader is lower, by a margin of 15%, than the conventional singulation algorithm. The lower data volume is a side-effect of the localized and distributed nature of the proposed solution. However, converse results hold, shown in Fig. 7-b, for the communication between the tags and cluster-heads. For instance, an exponential increase is observed with an increase in the number of clusters and tags. The high volume of data transfer between the tags and the cluster-heads translates into a much longer burst of the carrier wave signal, i.e., longer reading cycles, more processing at the cluster-heads and more bandwidth consumption. Sophisticated functionality at the tag, e.g., enhanced matching techniques of the serial number to the reader requests, although may reduce these overheads. However, these overheads do not impose any major threat to the overall performance. This is evident from the simulation analysis of the reading rate as they include these communication overheads.

IV. CONCLUSIONS

Anti-collision protocols are of great importance for the RFID systems. An efficient anti-collision algorithm results in lower collisions, increased data rates and lower delays under various tag distributions. In this paper, we proposed the parallel singulation algorithm for the distributed receiving architecture. In such architecture, the interrogation zone of the reader is divided into a number of micro-zones or clusters. The parallel singulation algorithm singulates tags by interrogating each cluster autonomously and in parallel. The simulation results have been presented to show the superiority of our proposed anti-collision algorithm. Moreover, any existing or forthcoming anti-collision can be integrated with our approach for clusters interrogation.

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. I. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Transactions on Information Theory IT-25*, no. 5, pp. 505–515, 1979.

[4] N. Zhang and B. Vojcic, "Binary search algorithms with interference cancellation rfid systems," in *MILCOM 2005: IEEE Military Communications Conference*, 2005, pp. 950–955.

[5] J. Choi, D. Lee, H. Jeon, J. Cha, and H. Lee, "Enhanced binary search with time-divided responses for efficient rfid tag anti-collision," in *ICC '07: IEEE International Conference on Communications*, 2007, pp. 3853–3858.

[6] J. Myung, W. Lee, and J. Srivastava, "Adaptive binary splitting for efficient rfid tag anti-collision," *IEEE Communications Letters*, vol. 10, pp. 144–146, 2006.

[7] V. Namboodiri and L. Gao, "Energy-aware tag anti-collision protocols for rfid systems," in *PerCom '07: Fifth Annual IEEE International Conference on Pervasive Computing and Communications*, March 2007, pp. 23–36.

[8] S. Lee, S. Joo, and C. Lee, "An enhanced dynamic framed slotted aloha algorithm for rfid tag identification," in *MobiQuitous 2005: the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, 2005, pp. 166–172.

[9] J. Cha and J. Kim, "Dynamic framed slotted aloha algorithms using fast tag estimation method for rfid system," in *CCNC 2006: 3rd IEEE Consumer Communications and Networking Conference*, 2006, pp. 768–772.

[10] G. Khandelwal, K. L. A. Yener, and S. Serbetli, "Asap : A mac protocol for dense and time constrained rfid systems," in *ICC '06: IEEE International Conference on Communications*, 2006, pp. 4028–4033.

[11] K. Ali and H. Hassanein, "Distributed receiving in rfid systems," in *LCN '09: Proceedings of the 34th IEEE Conference on Local Computer Networks*, 2009, to appear.

[12] "The network simulator ns-2." [Online]. Available: http://www.isi.edu/nsnam/ns/

[13] 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.

[14] C. Law, K. Lee, and K.-Y. Siu, "Efficient memoryless protocol for tag identification," in *DIALM '00: Proceedings of the 4^{th} International Workshop on Discret e Algorithms and Methods for Mobile Computing and Communications*. New York, NY, USA: ACM Press, 2000, pp. 75–84.

978-1-4244-4148-8/09/$25.00 ©2009
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE "GLOBECOM" 2009 proceedings.