Using Neighbor and Tag Estimations for Redundant Reader Eliminations in RFID Networks

Kashif Ali
School of Computing,
Queen's University, Canada
Email: kashif@cs.queensu.ca

Waleed Alsalih
Computer Science Department,
King Saud University, Saudi Arabia
Email: wsalih@ksu.edu.sa

Hossam S. Hassanein
School of Computing,
Queen's University, Canada
Email: hossam@cs.queensu.ca

***Abstract*—Deployments of Radio Frequency Identification (RFID) networks are anticipated to be dense and ad-hoc. These deployments usually involve redundant readers having overlapping interrogation zones and, hence, causing immense reader collisions. Elimination of redundant readers, from the network, is of utmost importance as otherwise they affect the lifetime and the operational capacity of the overall RFID network. In this paper, we propose a light-weight greedy algorithm that detects and eliminates redundant readers from the network. Our algorithm uses the ratio of tag counts to the number of neighboring readers of each reader to estimate the likelihood for that reader to be redundant. The proposed algorithm is highly scalable and poses a minimal communication overhead as compared with existing schemes in the literature.**

***Keywords*— redundant reader elimination, RFID tag estimation, greedy algorithm**

I. INTRODUCTION

Radio Frequency Identification (RFID) is an emerging automated identification technology that turns objects into a mobile network of nodes, which can then be used to track objects, trigger events and to take appropriate actions. With the emergence of small-size and low-power RFID readers, e.g., Skyetek M1-mini RFID reader, it is envisioned to integrate RFID networks with wireless sensor networks [1] (e.g., MICA2DOT) to achieve maximal event sensing and to use WiFi or cellular networks for the communication needs. Applications of such an architecture, e.g., Internet of things, demand dense and ad-hoc deployment of different components including RFID readers.

The nature of ad-hoc and dense deployment yields overlapped interrogation zones among RFID readers. This results into immense wireless interference without any coverage enhancements. Furthermore, the overlapping readers may also interrogate the same tags set, at the same time, resulting into data corruption. These readers collisions severely affects the performance of the overall system [2]. Furthermore, dense deployment may result in having redundant readers; a redundant reader is one whose interrogation area is fully covered by neighboring readers. An example of such a scenario is shown in Fig. 1 wherein the readers $R2$ and $R3$ are redundant. That is because all tags covered by $R2$ or $R3$ (i.e., $T2$ and $T3$) are also covered either by $R1$ or by $R4$. However, tag $T1$ is exclusively covered by $R1$ and tag $T4$ is exclusively covered by $R4$. A redundant reader is one that does not exclusively cover any tag. These redundant readers translate into unnecessary energy consumptions due to duplicate interrogations. Moreover, it results into extra processing and communication overheads and does not necessarily bring an enhancement to the tags coverage. Thereby, in order to extend the lifetime of the network and to improve its performance, there is a significant need for light-weight methods that efficiently detect and eliminate redundant readers.

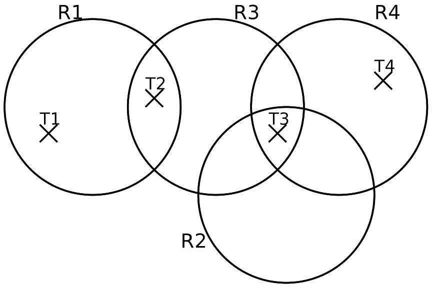


Fig. 1: An RFID network with redundant readers.

Many schemes exist in the literature for redundant readers elimination in RFID networks. Elimination of readers is mainly based on different markings, made by covering readers, into the tags memory. These markings include tags count of the covering readers [3], identity of the first interrogating reader [4], identity of the first singulating reader [5], [6], count of the neighboring readers [7] and others. Based on these markings, an algorithm decides on the redundancy of an RFID reader. These schemes, however, are not light-weight as they require considerable states to be maintained by the tags and involve frequent interrogations and singulations.

In this paper, we propose a light-weight greedy algorithm to detect and eliminate redundant readers in RFID networks. The algorithm estimates the number of tags each reader covers and finds the number of neighboring readers it has. The motivation is that a reader covering less tags and one that has more neighboring readers have a higher chance of being redundant. Based on this observation, our algorithm finds the ratio of the number of tags a particular reader covers to the number of neighboring readers it has, and uses that ratio to predict whether that reader is redundant or not. The performance of our algorithm is evaluated and compared with the performance of existing schemes using simulations. While our scheme has less communication overhead, simulation results show that our scheme outperforms other schemes in terms of the amount of redundancy being detected and eliminated.

The remainder of the paper is organized as follows. Section II surveys the existing literature in the context of redundant reader elimination. Section III describes our proposed redundant reader elimination algorithm. Section IV presents simulation methodology and analyzes the results using various performance metrics. Finally, section V concludes our work.

II. BACKGROUND

Numerous schemes have been proposed in the literature for redundant reader elimination problem. The main motivation is to minimize readers collision and to reduce overlapping among readers coverage regions. The Redundant Reader Elimination (RRE) is a pioneering scheme in this direction [3]. The RRE scheme starts off with each reader broadcasting a query embedded with its identity and tags count within its interrogation zone. The tags count of each reader is to be stored by each tag receiving the broadcasted query. Access to a particular tag is granted to a reader whose broadcasted query was received by that tag and has the maximum tags count. Readers which are not granted access to any tag are marked redundant and, hence, eliminated. Tags in the RRE scheme incur significant communication overheads and they are required to perform frequent memory read and write operations. To overcome the situation, the Layered Elimination Optimization (LEO) and Layered Elimination Optimization with Redundant Reader Elimination (LEO+RRE) schemes [4] introduce the concept of ”first-read first-own” principle. An RFID reader tries to write its identity into the tag memory. The reader, which manages to singulate a tag, writes its own identity first is granted access to that tag. In the case of LEO+RRE algorithm, the RRE algorithm is run afterwards to eliminate any left-out redundant readers. In a similar approach, the authors in [4] introduced the concept of ”first-arrive first-serve” approach wherein the time delay readers encounter to read a tag is used to find a reader granted access to that tag. A reader queries and writes its identity to the tags it covers and is successful if and only if no other readers have done so. The reader without any tags is considered to be redundant and, hence, is eliminated. In [6], the authors have proposed a Two-step Redundant Reader Elimination (TRRE) based scheme, which is very much similar to the LEO scheme. In this scheme, a reader sends out its query packet, embedded with its own identity, to all tags it covers. The tag responds either with a NULL, which implies the reader has been granted the ownership, or with an identity, which is different from the querys embedded identity, which implies the tags ownership has already been assigned to another reader. A reader without any tag’s ownership is marked as redundant and, hence, eliminated. Although the scheme is light-weight, it is not effective in eliminating all redundancy as compared with the LEO+RRE scheme for example. In [7], the author introduced the idea that a reader with a smaller number of neighbors has a lower probability of interfering with other readers and, hence, should be selected. Using a cost function composed of the tags count and the number of neighbors, the chance that a particular reader is redundant is estimated.

All of the above mentioned schemes, and others in the literature, require significant number of iterative singulations and state preserving between tags and, therefore, they consume significant network resources to an extent that the gains from redundant readers elimination may be overshadowed. Light-weight schemes, on the other hand, do not suffer such a significant overhead, yet they tolerate some redundancy. We claim that our proposed scheme is both light-weight and very effective in detecting and eliminating redundant readers in both sparse and dense deployments.

III. NTE: NEIGHBORS AND TAGS ESTIMATION BASED ALGORITHM

In this section, we present our redundant reader elimination algorithm. We start with some assumptions and definitions we use in this work and that is followed by a detailed description of the algorithm we propose. We then show further details of the algorithm using an illustrative example. Finally, we discuss the algorithm’s computational and communication complexity.

A. Assumptions and definitions

We consider an RFID system of n passive tags and multiple readers. The transmission range of a passive tag t_i is modeled as a sphere of radius r_i, i.e., tag t_i can be interrogated by an RFID reader if and only if the distance between the tag and the reader is at most r_i. A reader can communicate with its neighboring readers and also some central middleware. We assume the existence of a central middleware as depicted by the EPC Gen2 standard [8], where our algorithm is to be executed.

Formally, the redundant reader problem is defined as follow. Given a set of RFID tags and a set of RFID readers covering all tags, find the minimum cardinality subset of RFID readers that cover all the tags [3]. To this extent we propose the Neighbor and Tag Estimation (NTE) algorithm.

B. The NTE algorithm

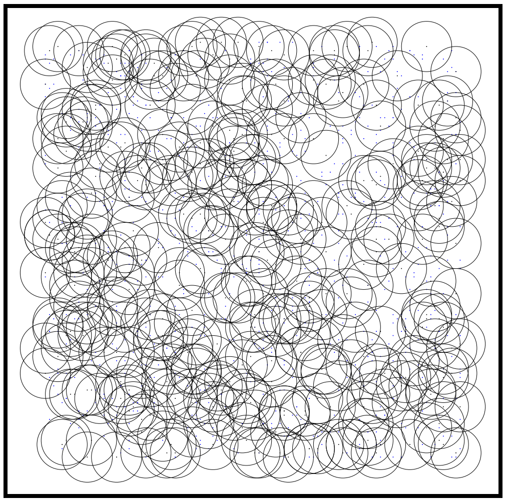
The redundant reader elimination problem is known to be NP-hard [3]. Therefore, we propose the NTE algorithm, which is a heuristic greedy algorithm, to find near optimal solutions in a reasonable time. The main idea is that when a reader R_i has a large number of neighboring readers, tags covered by R_i have a high probability of being also covered by other readers in the neighborhood of R_i. The algorithm uses this observation to identify redundant readers. For each active reader R_i, the algorithm assigns a weight based on the ratio of the number of active tags covered by R_i to the number of active neighboring readers of R_i. Initially, all tags are active and all readers are active. At the beginning of the each iteration, weights of active readers are calculated and then a reader with the maximum weight is deactivated and all tags covered by that reader are deactivated, too. Any reader with a weight of 0 is considered to be redundant and deactivated. This process continues until all tags are inactive, at which point all remaining active readers are considered to be redundant. The pseudocode for the NTE algorithm is shown in Algorithm 1.

In the presentation of our algorithm, we use the following notations:

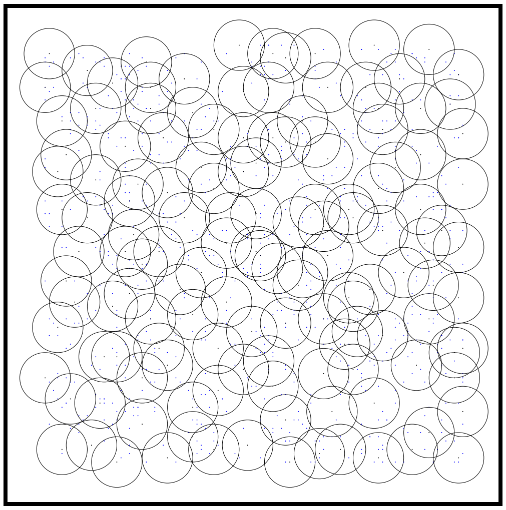
- $\mathcal{R} = \{R_1, R_2, \ldots, R_n\}$ is the set of active readers.
- $R^a = \{R_1^a, R_2^a, \ldots, R_m^a\}$ is the set of non-redundant readers.
- $R^r = \{R_1^r, R_2^r, \ldots, R_l^r\}$ is the set of redundant readers.
- $E(R_i)$ is the estimated number of active tags within the interrogation range of reader R_i.
- $N(R_i)$ is the number of active neighboring (interfering) readers of reader R_i.

Algorithm 1: The greedy NTE algorithm

```
1  NTE-Reader($\mathcal{R}$):
2  // Initially all tags are active;
3  $\mathcal{R}^a = \phi$;
4  $\mathcal{R}^r = \phi$;
5  while $\mathcal{R} \neq \phi$ do
6      $max \leftarrow 0$ ;
7      foreach $R_i \in \mathcal{R}$ do
8          $E(R_i)$= An estimate to the number of active tags covered by $R_i$;
9          $N(R_i)$ = The number of active neighboring readers for $R_i$;
10         if $E(R_i) = 0$ then
11             $\mathcal{R}^r \leftarrow \mathcal{R}^r \cup \{R_i\}$;
12             $\mathcal{R} \leftarrow \mathcal{R} - \{R_i\}$;
13         end
14         else
15             if $N(R_i) > 0$ then
16                 $Ratio[i] \leftarrow E(R_i)/N(R_i)$;
17             end
18             else
19                 $Ratio[i] = \infty$;
20             end
21             if $max$ < Ratio[i] then
22                 $max \leftarrow Ratio[i]$;
23                 $idx \leftarrow i$;
24             end
25         end
26     end
27     $\mathcal{R}^a \leftarrow \mathcal{R}^a \cup \{R_{idx}\}$;
28     $\mathcal{R} \leftarrow \{\mathcal{R}\} - \{R_{idx}\}$;
29     All tags covered by $R_{idx}$ are deactivated;
30     // using a broadcasted request.
31 end
32 return $\mathcal{R}^r$;
```



(a) Network before the execution of NTE algorithm



(b) Network after the execution of NTE algorithm

Fig. 2: Snapshot of random RFID network from the simulator

The algorithm begins with basic housekeeping (lines 3 and 4). At the beginning of each iteration, each active reader R_i estimates the number of tags it covers (i.e., $E(R_i)$) and determines the number of active neighboring readers it has (i.e., $N(R_i)$) (lines 8 and 9). To estimate the number of tags, the scheme presented in [9] can be used as it can estimate the number of tags within 99% accuracy ratio. The active neighboring readers of a reader R_i are those readers which are within the communication range of R_i and, hence, can be easily counted. Readers that do not cover any active tags (i.e., $E(R_i) = 0$) are considered to be redundant (lines 10-13). Amongst active readers that still cover at least one active tag, the one with the maximum weight (i.e., $Ratio[i]$) is selected and excluded from the redundancy list (lines 27 and 28), and active tags covered by that reader are deactivated (line 29). Any tie is broken arbitrarily. This can be done by a single message broadcasted by the selected reader. The selected reader itself is deactivated and not counted in the number of neighboring readers during subsequent iterations. The algorithm terminates when the set of readers R is empty (i.e., all readers have been classified as redundant or not).

A snapshot of an example of an RFID network before and after applying the NTE algorithm is shown in Fig. 2, which shows how the NTE algorithm described above is successful in eliminating a significant number of redundant readers.

C. Example

We show in this subsection how our algorithm finds the redundant readers in the scenario shown in Fig. 1. The set of readers $\mathcal{R} = \{R_1, R_2, R_3, R_4\}$ is given, as input, to the NTE algorithm. Tracing the NTE algorithm on this instance is shown in Table I.

In the first step, all readers in $\mathcal{R}$ estimate the number of active tags they cover and the number of active neighboring readers they have. Readers R_1, R_2, R_3 and R_4 have 2 tags, 1 tag, 2 tags and 2 tags, respectively. The numbers of active neighboring readers R_1, R_2, R_3 and R_4 have are 1, 2, 3 and 2, respectively. Accordingly, reader R_1 will have the maximum weight and, hence, will be excluded from the list of redundant readers and deactivated be removing it from the set $\mathcal{R}$. Active tags covered by R_1 (i.e., T_1 and T_2) are deactivated.

In the second step, active readers (i.e., the set $\mathcal{R}$) estimate the number of active tags they cover and the number of active neighboring readers they have. At this stage, reader R_1 and tags T_1 and T_2 are not counted. Therefore, readers R_2, R_3 and R_4 cover 1 tag, 1 tag and 2 tags, respectively. The numbers of active neighboring readers they have are 2, 2 and 2, respectively. R_4 is picked as it has the maximum weight.

In the last step, readers R_2 and R_3 will be added to the list of redundant readers as they do not cover any active tag. As a result, the readers R_1 and R_4 can be used to interrogate all tags and readers R_2 and R_3 will be turned off as they are redundant.

TABLE I: Tracing the NTE algorithm for the scenario of Fig. 1

(a) Step-I

Input:	$\mathcal{R} = \{R_1, R_2, R_3, R_4\}$	$\mathcal{R}^a = \phi$
Reader	$E(R_i)$	$N(R_i)$
R_1	2	1
R_2	1	2
R_3	2	3
R_4	2	2
Output:	$\mathcal{R} = \{R_2, R_3, R_4\}$	$\mathcal{R}^a = \{R_1\}$

(b) Step-II

Input:	$\mathcal{R} = \{R_2, R_3, R_4\}$	$\mathcal{R}^a = \{R_1\}$
Reader	$E(R_i)$	$N(R_i)$
R_1	-	-
R_2	1	2
R_3	1	2
R_4	2	2
Output:	$\mathcal{R} = \{R_2, R_3\}$	$\mathcal{R}^a = \{R_1, R_4\}$

(c) Step-III

Input:	$\mathcal{R} = \{R_2, R_3\}$	$\mathcal{R}^a = \{R_1, R_4\}$
Reader	$E(R_i)$	$N(R_i)$
R_1	-	-
R_2	0	0
R_3	0	0
R_4	-	-
Output:	$\mathcal{R}^r = \{R_2, R_3\}$	$\mathcal{R}^a = \{R_1, R_4\}$

D. Time complexity

Estimating the number of tags covered by a reader can be done in a constant time [9]. Deactivating tags covered by a particular reader can be also done in a constant time, as it just requires a single broadcasted message. When a reader is deactivated, it can broadcast a message to its neighboring readers telling them that it is going into an inactive mode; upon receiving such a message, reader decrements its number of active neighboring readers. Thereby, readers manipulate the number of tags they cover and the number of neighboring readers they have in a constant time. Now, we are left with the *while loop* that have, in the worst case, $O(N)$ iterations, where N is the number of readers. The time complexity of each iteration is dominated by the process of finding the reader with the maximum weight which is done in a centralized fashion and, hence, takes $O(M)$ time, where M is the number of active readers. Therefore, the overall time complexity of the algorithm is $O(N^2)$.

IV. PERFORMANCE EVALUATION AND RESULTS

In this section, we evaluate the performance of the proposed NTE redundant reader elimination algorithm using various performance metrics to compare it with other recently proposed schemes.

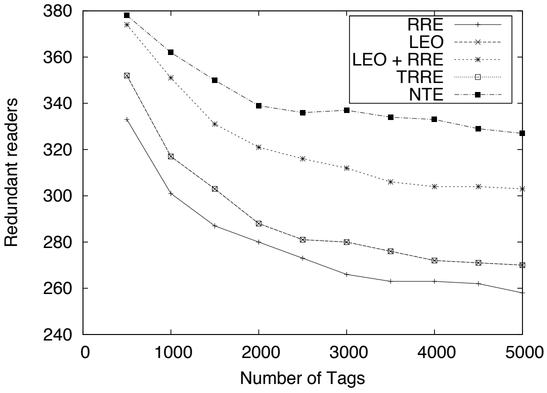
A. Evaluation methodology

We have implemented the RFID system using an in-house simulator. For comparison, we use recently proposed algorithms such as RRE [3], LEO+RRE [4] and TREE [6]. These schemes were selected based on the their relative performance improvements over other schemes in the literature. For comparative analysis, our evaluation metrics are the number of redundant readers, the number of tag reads and the number of tag writes. A high number of redundant readers imply the scheme is effective in detecting and eliminating redundant readers. A low number of tag reads and writes implies that the scheme is light-weight and does not require state maintenance on the tags. Furthermore, we evaluated the NTE scheme in terms of resilience to the reader's range.

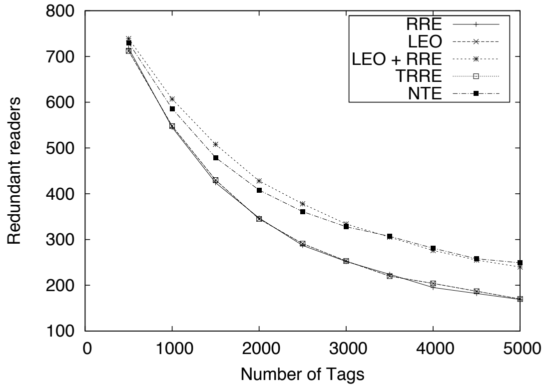
Unless otherwise mentioned, simulations are performed using the following parameters. The RFID reader has an interrogation range of 5m and an interrogation area of 250x250m^2. Both tags and readers are assumed to be randomly place within the interrogation area. The results are averaged over twenty different runs generated using distinct random seeds.

B. Simulation Results

1) Redundant readers: The overall network performance, mainly in terms of energy consumption and reading delay, is influenced by interference among readers [10]. Hence, an effective scheme is one that detects more redundant readers. Comparisons of different schemes based on the number of redundant readers they detect for both dense (50x50m^2) and sparse (250x250m^2) networks are depicted in Fig. 3. In dense environments, the NTE algorithm detects redundant readers

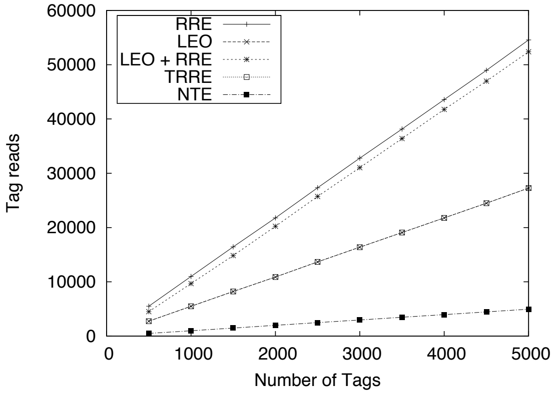


(a) Simulation area of 50x50m^2

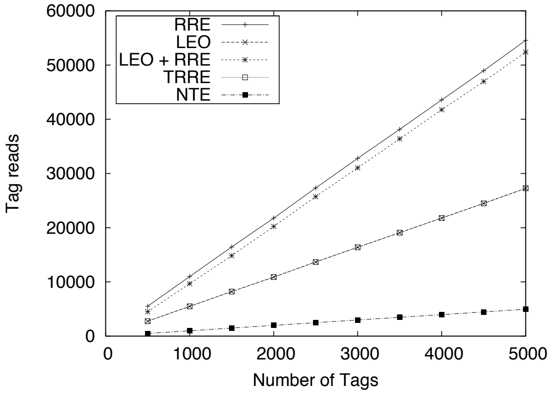


(b) Simulation area of 250x250m^2

Fig. 3: Redundant reader detected in sparse and dense network



(a) Simulation area of 50x50m^2



(b) Simulation area of 250x250m^2

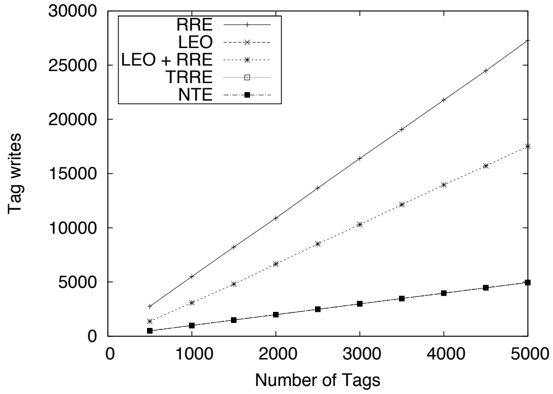
Fig. 4: Number of tag reads in sparse and dense environments

far beyond other schemes whereas in a sparse environment the improvement obtained is not significant. For instance, in dense environments, the NTE schemes can detect 15% more redundant readers than the best of the other schemes, i.e., LEO+RRE. However, in sparse environments only 2% improvement is observed over LEO+RRE. There are two reasons behind this behavior. First, in a dense environment a reader has a good chance of having a large number of neighbors and picking a reader with the maximum tags to neighbors ratio as a non-redundant reader has the potential to render many neighboring readers redundant, which is not the case in a sparse environment. Second, in a sparse environment, readers are expected to have a limited variation in the tags to neighbors ratio they have, and accordingly this ratio is not as effective as it is in a dense environment.

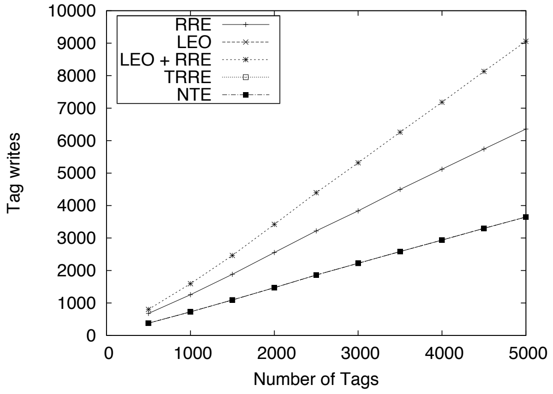
2) Tags reads and writes: Most of the existing schemes, e.g., RRE and LEO, in the literature have high communication complexities associated with them. Communication complexity is defined as the number of reads and writes operation made from and to tags. Tag singulation is required in order to read-from a tag. Tag singulation is an expensive process as it is determined by the number of tags within a reader's interrogation range and the overlapping among readers. Fig. 4 and Fig. 5 show comparisons between different schemes in terms of the number of read and write operations they incur. The number of reads and writes is obviously lower in sparse environments than it is in dense environments for all schemes. However, the number of reads and writes in the NTE scheme is many folds less than those in other schemes. This is because the reader, in the NTE scheme, writes only once to the tags within its range. Furthermore, each tag receives a single write operation.

Operations in the tags count estimation are not counted as read operations. This is because the estimation scheme does not require tags singulation. Read operations that make a difference are ones that require singulation and no such an operation is needed in the NTE scheme. Therefore, although the number of redundant readers detected by the LEO+RRE scheme is similar to that in the NTE scheme, in the sparse environment, the NTE scheme is much more efficient in terms of energy consumption and delay. In fact the NTE algorithm is at least six times more efficient in accessing the tags memory.

3) Effect of interrogation range: Increasing the interrogation range of readers increases overlapping among them. Fig. 6 shows the normalized percentage improvement achieved by the NTE scheme over the LEO+RRE scheme in an area



(a) Simulation area of 50x50m^2



(b) Simulation area of 250x250m^2

Fig. 5: Number of tags write in sparse and dense environments

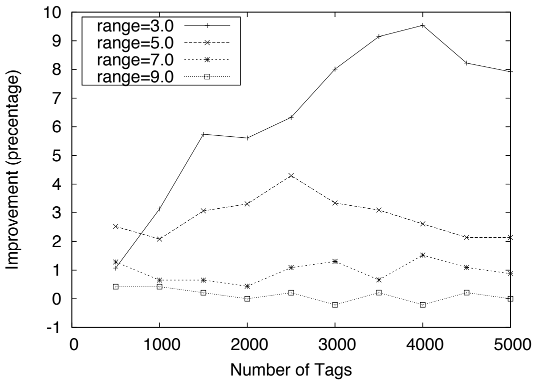


Fig. 6: Effect of reader's range on NTE algorithm

of $150\text{x}150m^2$. The normalized percentage improvement decreases with an increase of reader's range. The reason for this behavior is that the number of tags and neighboring readers increases with the range and, hence, the overall tags to neighbors ratio decreases. This will results in lesser redundant readers as per definition of the NTE scheme. However, even for the worst case scenario, the proposed NTE scheme not only performs, in terms of redundant readers elimination, as good as the next best scheme but surpass, by many folds, all existing schemes in terms of lower tag access.

To summarize, the proposed NTE scheme is more effective in detecting redundant readers than other schemes in both dense and sparse environments. However, the improvement is higher in dense environments. Furthermore, in both dense and sparse environments, the numbers of read and write operations in the NTE scheme is many folds smaller than that of other schemes and, hence, has very lower time and energy overheads.

V. CONCLUSION

Deployments of Radio Frequency Identification (RFID) networks are anticipated to be dense and ad-hoc. These deployments result in significant overlapping and coverage redundancy among readers. Detecting and eliminating redundant readers are very important to save the time and energy consumed by an RFID network to interrogate a set of tags. In this paper, we propose a light-weight greedy algorithm for the redundant readers elimination problem. An empirical study is presented to evaluate the performance of our algorithm as compared with existing ones. Not only does our algorithm require less communication between readers and tags, but also is more effective in detecting redundant readers than other schemes as validated by comprehensive experiments.

REFERENCES

[1] L. Zhang and Z. Wang, "Integration of rfid into wireless sensor networks: Architectures, opportunities and challenging problems," in *GCCW '06: Proceedings of the Fifth International Conference on Grid and Cooperative Computing Workshops*, 2006, pp. 463–469.

[2] W. Yoon and N. H. Vaidya, "Rfid reader collision problem: performance analysis and medium access," *Wireless Communications and Mobile Computing*, pp. 1–24, 2010.

[3] C. Bogdan, K. R. Murali, K. Mehmet, H. Christoph, and G. Ananth, "Redundant-reader elimination in rfid systems," in *SECON'05:Proceedings of the Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*. IEEE Computer Society, 2005, pp. 176–184.

[4] C.-H. Hsu, Y.-M. Chen, and C.-T. Yang, "A layered optimization approach for redundant reader elimination in wireless rfid networks," in *APSCC'07: Proceedings of the 2nd IEEE Asia-Pacific Service Computing Conference*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 138–145.

[5] K.-M. Yu, C. W. Yu, and Z.-Y. Lin, "A density-based algorithm for redundant reader elimination in a rfid network," in *Proceedings of Second International Conference on Future Generation Communication and Networking*, 2008, pp. 89–92.

[6] Z.-Y. Yang and J.-L. Chen, "The simulation and analysis of algorithms for redundant reader elimination in rfid system," in *Third UKSim European Symposium on Computer Modeling and Simulation, 2009. EMS '09.*, nov. 2009, pp. 494 –498.

[7] N. Irfan and M. C. Yagoub, "Efficient algorithm for redundant reader elimination in wireless rfid networks," *International journal of computer science issues*, vol. 3, no. 11, pp. 1–8, May 2010.

[8] EPCglobal, "Epc radio-frequency identity protocols class-1 generation-2 uhf rfid protocol for communications at 860 mhz .960 mhz version 1.0.9," 2005.

[9] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in rfid systems," in *MobiCom '06: Proceedings of the 12th Annual International Conference on Mobile Computing and Networking*, 2006, pp. 322–333.

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