CERIAS Tech Report 2005-63

REDUNDANT READER ELIMINATION IN RFID SYSTEMS

by Bogdan Carbunar, Murali Krishna Ramanathan, Mehmet Koyuturk, Christoph Hoffmann, Ananth Grama

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

Redundant Reader Elimination in RFID Systems

Bogdan Cărbunar*, Murali Krishna Ramanathan†, Mehmet Koyutürk†, Christoph Hoffmann† and Ananth Grama†
*Pervasive Platforms and Architectures Lab
Motorola Labs, Schaumburg, IL
Email: carbunar@motorola.com
†Department of Computer Science
Purdue University, West Lafayette, IN
Email:{rmk,koyuturk,cmh,ayg}@cs.purdue.edu

***Abstract*— While recent technological advances have motivated large-scale deployment of RFID systems, a number of critical design issues remain unresolved. In this paper we deal with detecting redundant RFID readers (the *redundant reader problem*). The underlying difficulty associated with this problem arises from the lack of collision detection mechanisms, the potential inability of RFID readers to relay packets generated by other readers, and severe resource constraints on RFID tags. We prove that an optimal solution to the redundant reader problem is NP-hard and propose a randomized, distributed, and localized approximation algorithm, RRE. We provide a detailed probabilistic analysis of the accuracy and time complexity of RRE and conduct elaborate simulations to demonstrate their correctness and efficiency.**

I. Introduction

Radio Frequency Identifier (RFID) systems consist of two types of components, RFID transponders (tags) and RFID transceivers (readers). RFID tags are comprised of a small integrated circuit for storing information and an antenna used for communication. Tags may be passive, i.e., they do not require batteries and instead use energy of the received signal to reveal its stored information. RFID readers are capable of reading the information stored at non line-of-sight RFID tags placed in their vicinity and communicate it through a wired or wireless interface to a central database. Supply chain automation, cold chain management (temperature logging), identification of products at check-out points, access control and security, are among common applications of RFID systems.

Significant investment by major retailers such as Wal-Mart and Tesco, mandating their manufacturers to place tags on cases and pallets provides a strong motivation for the large scale deployment of RFID systems. This investment is based on recent technological advances that have made possible, mass production of inexpensive RFID tags. Their cost is expected to drop below the five cents/tag threshold [1]. The main advantages of RFID systems are price efficiency (billions of dollars in anticipated savings for Wal-Mart alone [2]) and accuracy of stock management (GAP documented an increase of accuracy from 85% to 99.9% when using RFID technology [3]).

The miniaturization of RFID readers (SkyeRead M1-Mini [4]), coupled with their enhancement with Wi-Fi or cellular capabilities (SmartCode [5]), broadens the scope of applications of RFID systems. Wireless RFID systems, similar to wireless sensor networks, can be deployed in a ad-hoc fashion instead of being statically pre-installed. Unlike sensor networks, wireless RFID systems have the ability to decouple the sensing and communication functions. Since RFID tags interfaceable with external sensors, such as temperature and shock sensors or tamper indicators, have already been produced [6], wireless RFID systems can be easily extended with new sensing capabilities by deploying corresponding RFID tag types. Furthermore, the existing compatibility between recent RFID readers (SkyeRead M1-Mini [4]) and MICA2DOT motes motivates integration of wireless sensor and RFID networks. Such a hybrid infrastructure combines the affordability of deployment with the efficient and accurate identification and monitoring of objects.

The main problem addressed in this paper, of extending the lifetime of wireless RFID reader networks, stems from the limited battery life of wireless RFID readers and the need for accurate monitoring of areas of interest. This, in turn requires dense deployment of wireless RFID tags and readers. The solution proposed in this paper is based on the identification of redundant RFID readers, which we define in terms of the covered RFID tags. The temporary deactivation of such readers does not reduce the number of tags covered by the initial reader network. Our purpose is to detect the maximum number of redundant readers that can be safely turned off simultaneously. For example, in Figure 1, all RFID readers are redundant (i.e., each tag is covered by multiple readers), however, only a subset may be simultaneously deactivated.

While the problem of determining coverage redundancy has been extensively studied in wireless sensor networks [7], [8], [9], [10], it differs from the redundant RFID reader elimination problem in several aspects. First, coverage is defined in terms of contiguous circular areas associated with sensors, whereas in RFID systems coverage is defined in terms of discrete points (RFID tags). Second, solution to this problem for sensor networks relies on the existence of location information, or at least the ability to estimate distances between adjacent sensors. Due to the limited resources of RFID tags, in RFID systems such an assumption is not reasonable. Third, the limited resources of RFID tags coupled with the potential inability of RFID readers to act as packet routers, considerably restricts the solution space of the redundant reader problem.

We prove that even with centralized knowledge of the RFID

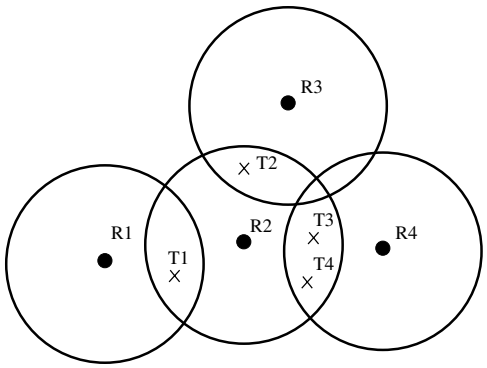

Fig. 1. Redundant reader example: readers $\mathtt{R}_1$, $\mathtt{R}_2$, $\mathtt{R}_3$ and $\mathtt{R}_4$ are redundant since the tags covered by each is covered by at least one other reader. This redundancy information would not be detected by a sensor redundancy detection algorithm, since the coverage areas of any of the readers are not subsumed by the others. The optimal solution requires only $\mathtt{R}_2$ to be active, while the other readers may be turned off.

system topology, an optimal solution for the redundant reader elimination problem is NP-hard. We introduce a randomized, decentralized, and localized approximation algorithm for the redundant reader elimination problem, called RRE. For each reader, the first step of RRE detects the set of RFID tags placed in the vicinity of a reader. The difficulty associated with this step rests on the potential occurrence of reader collisions at tags. Reader collisions occur at tags situated in the vicinity of two or more readers that are simultaneously sending queries. Such tags may be unable to correctly decode the queries, potentially leading to unexpected behavior. The absence of global topology information, where readers might not be aware of generated collisions, makes the task of accurate query scheduling difficult. For completeness reasons, we shortly describe a randomized, distributed, and localized algorithm, RCA [11], for avoiding reader collisions and allowing RFID readers to accurately detect the tags in their vicinity.

In the second step of RRE, each RFID reader attempts to write its tag count (number of covered tags) on to all its covered tags. A tag placed in the vicinity of several readers will overwrite the count stored on behalf of a reader only if the new value is larger. The reader that issued the highest count for a tag, *locks* the tag. In the final step of RRE, each reader sequentially queries all its covered tags to discover the ones it has locked. A reader that has not locked any of its covered tags is declared redundant.

RCA and the subsequent steps of RRE rely on a randomized querying technique, for avoiding reader collisions. Section III presents this technique in the context of RCA. Section IV defines the redundant reader problem and proves its NP-hardness and Section V presents our solution, RRE. Section VI presents our simulation results and Section VIII draws conclusions.

II. Network Model

Our algorithms are designed under the following conservative assumptions.

- Our algorithms are applicable to any number of RFID readers and tags and we make no assumptions on the underlying reader or tag topology. We do not assume the presence of a centralized entity capable of collecting the topology of the reader network or controlling the behavior of individual readers. Thus, our algorithms do not rely on the ability of RFID readers to communicate.
- We assume the presence of passive tags only, as opposed to active tags (the latter are more powerful and expensive). Therefore, RFID tags use the energy of the received signal in order to answer queries from readers.
- Tags have limited memory. Part of it is read-only, used to store unique identifiers, and part of it is writable. Tags are capable of doing prefix matching.
- RFID readers are able to detect RFID tag collisions, occurring when multiple RFID tags reply to the same query.

III. Reader Collision Avoidance

We first examine the impact of collisions, an essential aspect of RFID systems. More precisely, we look at a popular solution for tag collisions and then propose an efficient solution for avoiding reader collisions.

Tag Collisions: The area around an RFID reader, where RFID tags can receive the reader's signal and their replies can be correctly decoded by the reader, is called the *interrogation zone* of the reader. The main functionality of an RFID reader is to detect the unique identifiers of all the RFID tags in its interrogation zone. Simultaneous replies from RFID tags situated in the interrogation zone of a reader make accurate decoding of signals impossible. This problem, known as the tag-collision problem, prevents an RFID reader from simultaneously reading all its covered RFID tags.

Several techniques have been proposed to solve the tag-collision problem. A popular solution, known as the tree walking algorithm(TWA) [12], is based on a recursive traversal of the binary name tree of RFID tag identifiers. The reader initially sends a broadcast query containing the "0" string. All RFID tags in its interrogation zone whose id prefix is "0" must reply. If a reply is received, or a tag-collision is detected, the reader recurses on the left and then the right subtree of "0", rooted at "00" and "01". However, if no reply is received, the reader concludes the absence of "0"-prefixed tags in its interrogation zone and subsequently sends a "1" query. For a reader, the complexity of TWA is proportional to the number of tags present in its interrogation zone and to the length of the binary representation of RFID tag identifiers.

Reader Collisions: TWA [12] does not solve the following related problem. When two RFID readers are placed close enough for their interrogation zones to overlap but far enough to prevent direct communication, RFID tags placed within the intersection area of the interrogation zones may receive queries from both readers simultaneously. Such queries, potentially part of the TWA protocol, will interfere, preventing the corresponding RFID tags from correctly interpreting the queries. These tags may escape detection by any reader in the system.

Outline of RCA: We propose a randomized, distributed and localized solution to the reader collision problem in [11]. We present the details in this paper for clarity. Our algorithm, named RCA (Reader Collision Avoidance), is presented in the context of TWA. However, a similar approach can be extended to any scenario where a reader needs to communicate with a tag. Similar to TWA, in RCA an RFID reader sends a broadcast query containing a certain prefix expected to match the identifiers of RFID tags in its interrogation zone. However, unlike TWA, where the lack of an answer is considered to denote absence of matching RFID tags, the reader backs-off for a random number of time frames and repeats the query. The purpose of the random back-off and query repetition is to ensure w.h.p. the choice of a time frame not picked by another RFID reader, thus avoiding reader collisions.

The premise of the algorithm is as follows. An RFID reader divides time into disjoint epochs and each epoch is further divided into multiple disjoint time frames. In each epoch, an RFID reader picks a frame uniformly at random and sends its query in that frame. If no tag answer is received, the RFID reader repeats the query in a randomly chosen time frame of the next epoch. Even if a reader collision at matching RFID tags has occurred during the query, the query duplication correlated with the random backoff decreases the chances of repeated reader collisions. Theorem 1 proves that if a query is not answered $\mathtt{O}(\log \psi)$ times, then w.h.p., there are no RFID tags matching the query in the interrogation zone of the RFID reader. Here, ψ is the total number of RFID readers. If, however, an answer is received, either as a clear tag response or by detecting a tag collision, the RFID reader recursively moves to the next query, as in the TWA algorithm.

Implementation: Algorithm 1 presents the pseudocode for RCA using an Orca [13] like syntax. Orca is a parallel programming language for distributed systems, that provides elegant constructs for expressing reactive behavior, such as *guards*. Operations consist of one or more guards with syntax

guard `expression` **do** `statementSeq` **od**,

where `expression` is a boolean expression and `statementSeq` is a sequence of statements. The operation containing guards blocks until one or more guards are true. Then one of the satisfied guards is randomly chosen and its statements are executed atomically.

The operation of a tag is shown in Algorithm 1, lines 1-10. A tag replies only to queries containing strings whose prefixes match its own identifier (lines 5-9). inQ.first is used to denote the packet currently received by the tag. The operation of a reader is shown in Algorithm 1, lines 11-31. Time is divided into epochs, with each epoch containing a fixed number, n, of time frames. The duration of a time frame is equal to the time necessary for a query to propagate from a reader to a tag. For each prefix queried, the reader waits for a maximum of `e` epochs (line 18) and in each epoch sends exactly one broadcast message containing the prefix. During each epoch, the broadcast message is sent in a randomly chosen time frame (lines 19-22).

Algorithm 1 The generic reader and tag behavior. `getRandom(v1, v2)` returns a random integer value between `v1` and `v2` and `bCast(packet)` is used to broadcast `packet`.

```
1. Object implementation RFIDTag;
2.   T_id : integer; #tag identifier
3.   inQ : queue; #queue of incoming packets
4.   Operation run()
5.       guard inQ.first.type = query do
6.           if prefixMatch(inQ.first.tid, T_id) then
7.               bCast(new packet(TAG));
8.           fi
9.       od
10. end

11. Object implementation RFIDReader;
12. count, e : integer; #epochs per bit read
13. frame, n : integer; #time frames in each epoch
14. T, T_out : integer; #time out value
15. inQ : queue; #queue of incoming packets
16. Operation treeWalk(prefix : integer)
17.       count := 0;
18.       while count++ < e do
19.           frame := getRandom(0, n);
20.           sleep(frame);
21.           T = getTime();
22.           bCast(new packet(query, prefix));
23.           guard inQ.first.type = TAG_COL || TAG do
24.               treeWalk(prefix + "0");
25.               treeWalk(prefix + "1");
26.           od
27.           guard getTime() - T ≥ T_out do
28.               sleep(n - frame - 1);
29.           od
30.       od
31. end
```

The lack of a reply may denote either the absence of a tag matching the queried prefix in the interrogation zone, or the occurrence of reader collisions at such tags. If less than `e` queries with the current prefix have been sent, the reader waits until the beginning of the next epoch to repeat the above process (lines 27-29). If no reply or collision is detected after `e` rounds, the reader ignores the subtree rooted at the queried prefix. However, the receipt of an individual reply or the detection of a tag collision stops this process, since the reader can now safely recurse on the two children of the employed prefix (lines 23-26).

Analysis: Let ψ be the total number of readers and γ the total number of RFID tags in the system, τ be the number of time frames per epoch and β be the bit length of RFID tag identifiers. In our analysis, we assume a star topology in which interrogation zones of all ψ RFID readers share all γ RFID tags. Note that this is a worst case assumption.

The following theorem, whose proof can be found in [11], provides an upper bound on the number of query repetitions in RCA.

Theorem 1: When $\tau = \gamma$, repeating each query $\mathtt{O}(\log \psi)$ times ensures w.h.p. at least one correct receival of a reader's query by all the RFID tags in its interrogation zone. This statement holds even when all ψ readers simultaneously

attempt to query their tags.

This theorem leads to the worst case time complexity of RCA, whose proof is in [11].

Complexity of RCA: The time complexity of RCA, $\mathtt{T_{RCA}}$ is $\mathtt{O}(\gamma\beta\log\psi)$ time epochs.

IV. The Redundant Reader Problem

We now define the redundant RFID reader problem and prove that finding the optimal solution is NP-hard. We define a redundant reader as follows:

Definition 1: An RFID reader that covers a set of RFID tags that are also covered by other RFID readers is referred to as a redundant reader.

According to this definition, all the RFID readers in Figure 1 are redundant. A simple solution to detect the redundant RFID readers is to have all RFID readers simultaneously broadcast a query containing the empty string. Since all the RFID tags that receive such a query must answer, an RFID reader that receives no reply is redundant. This is either because the RFID reader covers no RFID tag, or because interference occurred at all its covered RFID tags. Such a solution has two important drawbacks. First, it requires time synchronization between all RFID readers. Second, turning off all the redundant RFID readers may leave uncovered tags that were previously covered by at least two redundant readers (blind tags). For example, in Figure 1, the simultaneous deactivation of $\mathtt{R_1}$ and $\mathtt{R_2}$ leaves RFID tag $\mathtt{T_1}$ uncovered.

In order to maximize the number of RFID readers that can be simultaneously deactivated, the minimum number of readers that cover all RFID tags needs to be discovered. We define the redundant reader problem as follows:

Redundant Reader Problem: Given a set of RFID tags and a set of RFID readers covering all the RFID tags, find the minimum cardinality subset of RFID readers, covering all the tags.

For example, in Figure 1, $\mathtt{R_2}$ is the only reader that needs to be active. In order to prove that the redundant reader problem is NP-hard, we first prove the following lemma, illustrated in Figure 2.

Lemma 1: Given a set of n points, $p_1, p_2, .., p_n$, placed inside a circle of radius $\mathtt{R}$, there exists a subset of 3 of the n points, p_i, p_j, p_k, such that all the n points are placed inside $C(O_{ijk}, R)$. O_{ijk} is the mass center of p_i, p_j, p_k and $C(x, R)$ denotes the circle centered at x with radius R.

Proof: We provide a constructive proof. If all the points are covered by a circle of radius $\mathtt{R}$, then a circle of radius $\mathtt{R}$ going through 2 of the points and covering all the other points exists (see Figure 2). If the circle has a third of the n points on its perimeter, then we have completed the proof. Otherwise, shrink the circle until its perimeter touches a third point. The resulting circle has radius less than or equal to $\mathtt{R}$, is the circumcircle of three of the n points, and covers all the other points.

■

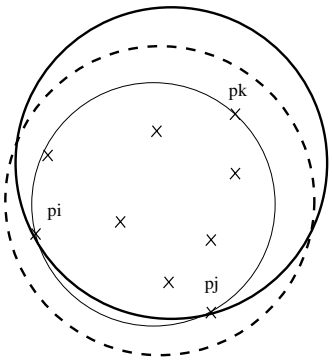

Fig. 2. Set of points covered by a circle of radius $\mathtt{R}$, shown with an interrupted perimeter. There is a circle of radius $\mathtt{R}$ going through points $\mathtt{p_i}$ and $\mathtt{p_j}$ and covering all the other points. Shrink this circle until it first touches one more point, $\mathtt{p_k}$. The resulting circle, has radius less than or equal to $\mathtt{R}$.

We can now prove the following important result.

Theorem 2: The redundant reader problem is NP-hard.

Proof: We prove its inclusion in the NP-hard set by providing a reduction from the geometric disk cover (DC) problem, known to be NP-hard [14]. The input for the DC problem consists of a set of $\mathtt{m}$ points and a value $\mathtt{R}$. The output consists of the minimum number of disks of radius $\mathtt{R}$ that cover all the points. We use the following polynomial-time reduction from DC to the redundant reader problem. Add a disk of radius $\mathtt{R}$ centered at each point in the input set of DC. Then, for all combinations of 3 points of the input set of DC, add a disk of radius $\mathtt{R}$, centered at the mass center of the 3 points. Let $\mathtt{S}$ denote the set of all disks created. It is clear that the disks in $\mathtt{S}$ cover all the input points of DC. Moreover, as a direct consequence of Lemma 1, the disks that form the solution for the DC problem are contained in $\mathtt{S}$. The reduction has $\mathtt{O}(\mathtt{m}^3)$ complexity. If a polynomial time algorithm for the redundant reader problem would exist, we could find the minimum number of disks needed to cover the points, which cannot be worse than the solution for the DC problem, in polynomial time. ■

V. Redundant Reader Elimination Algorithm

We propose a distributed approximation algorithm for the redundant reader problem. As specified in Section II, we make no assumption on the topology of the RFID reader network, effectively claiming no direct communication between RFID readers. We assume, however, the existence of writable tags that are able to store information upon requests from in-range RFID readers. We assume initially that RCA (see Section III) has been previously executed by all readers to identify RFID

tags in their vicinity. Later in this section we discuss a simple modification to our algorithm to remove this assumption.

Outline of RRE: RRE (Redundant Reader Elimination) consists of two steps. In the first step, each RFID reader attempts to write its tag count (number of covered tags) to all its covered RFID tags. An RFID tag only stores the highest value seen, along with the identity of the corresponding reader. For this, each reader issues a write command containing its reader identifier and tag count. Similar to RCA, the write operation is performed during $\mathtt{O}(\log \psi)$ consecutive epochs, once per epoch. During each epoch, the time frame for sending the write request is randomly chosen. As shown in Theorem 1, this process ensures w.h.p. that at least one write command issued by each RFID reader will be correctly received by all its covered RFID tags. Thus, after $\mathtt{O}(\log \psi)$ epochs, each RFID tag stores the largest number of tags covered by an RFID reader situated in its vicinity, along with the identity of that reader, called *holder* of the tag.

In the second step, an RFID reader queries each of its covered RFID tags and reads the identity of the tag's holder. A reader that locked at least one tag is responsible for monitoring the tag and will have to remain active. However, a reader that has locked no tag can be safely turned off. This is because all the tags covered by that reader are already covered by other readers that will stay active. The read queries issued by a reader for each of its tags are similarly repeated during random time frames for $\mathtt{O}(\log \psi)$ time epochs to avoid reader collisions occurring at queried tags.

In conclusion, each tag is locked by the reader in its vicinity that covers most tags. A reader that locks at least one tag is required to remain active. This strategy provides a distributed greedy heuristic for the redundant reader problem. Moreover, ties are broken arbitrarily. Ties occur at tags that can be locked by two or more contending readers, that is, readers that cover equal numbers of tags. In the current implementation, such a tag is locked by the first contending reader whose query reaches the tag.

Implementation: Algorithm 2 presents the pseudocode for RRE. The solution assumes writable RFID tags. The functionality of a writable tag is shown in operation `run` of `WritableRFIDTag` (lines 4-13). The RFID reader and tag objects inherit the corresponding variables defined in Algorithm 1. When a writable tag receives a `write` command containing the identifier of the reader issuing the command and its tag count, it saves the values locally only if the tag count is larger than the value currently stored. When the command received is a `read`, the tag returns a packet containing its identifier followed by the reader's identifier and count value stored locally.

The detection of redundant RFID readers is exhibited in operation `isRedundant` of `RFIDReader` (lines 18-39). First, a reader selects a random time frame during `e` consecutive epochs, and broadcasts a `write` command containing its identifier and tag count (lines 19-24). Subsequently, it queries each of its covered tags, using a `read` command, for `e` consecutive time epochs in order to find the tag's holder (lines 25-37). Note

Algorithm 2 The generic RFID reader and writable tag behavior for detecting redundant readers.

```
1.Object implementation WritableRFIDTag;
2.  R_id : integer; #identifier of locking reader
3.  count = 0 : integer; #count of highest bidder
4.  Operation run()
5.      guard inQ.first.type = write do
6.          if inQ.first.c > count then
7.              R_id := inQ.first.rid;
8.              count := inQ.first.c;
9.          fi;
10.     guard inQ.first.type = read do
11.         bCast(new packet(T_id, R_id, count));
12.     od
13. end

14.Object implementation RFIDReader;
15. R_id : integer]; #reader identifier
16. tags : array[integer] of integer; #covered tags
17. redundant = true : boolean;
18. Operation isRedundant(prefix : integer)
19.     while count + + < e do
20.         frame := getRandom(0, n);
21.         sleep(frame);
22.         bCast(new packet(write, R_id, tags.size));
23.         sleep(n − frame − 1);
24.     od
25.     for i in 1..tags.size do
26.         while count + + < e do
27.             T = getTime();
28.             frame := getRandom(0, n);
29.             sleep(frame);
30.             bCast(new packet(read, tags[i]));
31.             guard inQ.first.tid = tags[i] do
32.                 if inQ.rid ! = R_id then
33.                   redundant := false;
34.                 od
35.             guard getTime() − T > n do od
36.         od
37.     od
38.     if redundant = true do turnOff(); fi
39. end
```

that after sending a `read` command, at the chosen time frame, the reader waits either to receive a reply from the queried tag or for the epoch to end (lines 31-35).

A. Discussion

Synchronization: We have assumed until now that all RFID readers have already executed RCA, detecting all the RFID tags in their interrogation zone. This assumption ensures that on completion of the first step of RRE, tags placed in the vicinity of at least two readers store the highest number of tags covered by the readers. For example, in Figure 1, the count of tag $\mathtt{T}_3$ is 4, from reader $\mathtt{R}_2$. However, if we assume that initially RFID readers are not aware of the identity of adjacent tags and RCA needs to be executed just before RRE, the following scenario may occur (see Figure 1 for illustration): since $\mathtt{R}_4$ only covers two RFID tags, whereas $\mathtt{R}_2$ covers four, $\mathtt{R}_4$ will complete RCA before $\mathtt{R}_2$ and also the first step of RRE. Then, $\mathtt{R}_4$, upon discovering to be the holder of $\mathtt{T}_3$ and $\mathtt{T}_4$, will also decide to stay active, despite its redundancy.

In order to solve this problem, we require active RFID

readers to maintain a list of locked tags and to passively listen for RFID tag responses to queries initiated by other readers. When an RFID reader, $\mathtt{R}$, receives such a message, of format $\mathtt{R_x}, \mathtt{T_y}, \mathtt{c}$ (see Algorithm 2 line 11), indicating that the holder of tag $\mathtt{T_y}$ is $\mathtt{R_x}$ with a tag count $\mathtt{c}$, if $\mathtt{c}$ is larger than its own tag count, the reader $\mathtt{R}$ removes tag $\mathtt{T_y}$ from its list of locked tags. When the list is empty, the reader becomes redundant and can be safely turned off. Theorem 1 (see Theorem 1) proves that if such a scenario occurs, a reply of content $\mathtt{R_x}, \mathtt{T_y}, \mathtt{c}$ *will* be received by $\mathtt{R}$ for all tags $\mathtt{T_y}$ covered by readers with a larger tag count. Using the example in Figure 1, if $\mathtt{R_4}$ has $\mathtt{T_3}$ and $\mathtt{T_4}$ in its list of locked tags on completion of its first step of RRE, during $\mathtt{R_2}$'s execution of the first step of RRE, $\mathtt{R_2}$ will choose at least one time frame during $\mathtt{e}$ epochs, both for $\mathtt{T_3}$ and $\mathtt{T_4}$, when no other RFID reader is transmitting. Thus, $\mathtt{R_4}$ will overhear the replies of $\mathtt{T_3}$ and $\mathtt{T_4}$. Note that their replies will not generate a tag collision at $\mathtt{R_4}$, since the tags are queried sequentially by $\mathtt{R_2}$ (Algorithm 2 line 30).

System Adaptivity: The current description of RRE assumes a static environment. However, in reality, RFID tags and readers may fail and new components may be randomly deployed. Scenarios where new RFID tags are deployed in areas covered only by inactive readers, or when active RFID readers fail, leaving tags covered only by inactive readers, are particularly important. We present a simple extension of RRE that maintains the invariant of having at least one active RFID reader for each covered tag, in these two scenarios. Our solution periodically re-activates inactive readers and executes RRE on all the readers. Then, the following problem, illustrated in Figure 1, may occur. If the only active reader, $\mathtt{R_2}$, fails when $\mathtt{R_1}$, $\mathtt{R_3}$, and $\mathtt{R_4}$ are re-activated, tags $\mathtt{T_1}, .., \mathtt{T_4}$ have the associated count value set to 4. The re-activated readers discover, this time inaccurately, their redundancy and switch off, leaving all the tags uncovered.

One solution to this problem executes RCA periodically every $\mathtt{T}$ time units, to identify all its covered tags, including newly deployed ones. Subsequently, the readers reset the count value of each of their covered tags and re-execute RRE. An RFID tag will agree to set its counter to a smaller value, 0, since 0 is a control value (an RFID reader covering no tags will not issue a `write` command containing a 0 tag count field). Of course, this can lead to a situation in which even though no reader has failed, $\mathtt{R_2}$ sets the counter of its tags to 0 and then to 4, followed by the activation of $\mathtt{R_4}$, $\mathtt{R_4}$'s setting the counter of its tags to 0 and then to 2. Then, $\mathtt{R_4}$ and $\mathtt{R_2}$ will both decide to stay active even though $\mathtt{R_4}$ is redundant. A solution for this scenario is to set the period $\mathtt{T}$ of each reader to be inversely proportional to the tag count of the reader. Then, $\mathtt{R_2}$ will execute this procedure more often than $\mathtt{R_4}$, eventually causing $\mathtt{R_4}$ to discover its redundancy. Another solution, requiring more complex tags, is to have timers on tags. A tag may store a tag count only for a limited time, until the expiration of its timer. The timer is set when a new tag count value is stored by the tag.

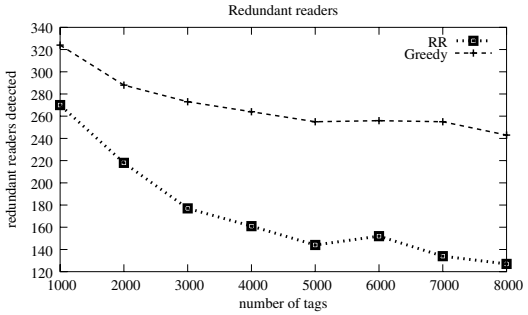

Fig. 3. Number of redundant readers discovered by RRE and Greedy when the number of RFID tags randomly deployed increases from 1000 to 8000. The number of RFID readers is constant, 500, throughout this experiment.

B. Analysis

Since the number of RFID tags covered by a reader is not known before running RCA, accurately evaluating the time necessary for RCA to complete is difficult. Even though the duration of the first step of RRE is fixed, $\log \psi$ time epochs, the second step of RRE may start at different times even for readers that have started RCA simultaneously. The question is then if, due to the lack of synchronization among RFID readers, RRE can leave uncovered tags. We define the following safety property which should hold for any distributed algorithm for the redundant reader elimination problem and prove that RRE satisfies it.

Safety: An algorithm for the redundant reader elimination problem is said to be safe, if it will not turn off RFID readers that cover RFID tags not covered by active readers.

Claim: RRE is safe.

Proof: Let us assume that a tag $\mathtt{T_1}$ is situated inside the interrogation zones of two readers, $\mathtt{R_1}$ and $\mathtt{R_2}$. Furthermore, $\mathtt{R_1}$ covers fewer tags than $\mathtt{R_2}$. Then, it is likely for $\mathtt{R_1}$ to start the second step of RRE before $\mathtt{R_2}$ has succeeded writing its tag count on its covered tags. Then, both $\mathtt{R_1}$ and $\mathtt{R_2}$ will believe to be the locker of $\mathtt{T_1}$. However, $\mathtt{T_1}$ will not be left uncovered, since both $\mathtt{R_1}$ and $\mathtt{R_2}$ are required to stay active. This will only decrease the number of redundant readers able to be simultaneously deactivated. ■

Complexity of RRE: $\mathtt{T_{RRE}} = \mathtt{O}(\gamma\beta \log \psi)$.

Proof: The complexity of RCA, is $\mathtt{O}(\gamma\beta \log \psi)$ (see Section III). The first step of RRE, where each RFID reader sends a write command to all its tags, takes $e \log \psi$ epochs. The second step, where RFID readers send queries to each of their tags, takes $\gamma \mathtt{e} \log \psi$ epochs. Thus, $\mathtt{T_{RRE}} = \mathtt{O}(\gamma\beta \log \psi)$. ■

VI. Simulation Results

All of our experiments are performed by randomly (uniformly) deploying RFID tags and readers in a $1000 \times 1000 m^2$ domain. In this section we analyze the efficiency of RRE in terms of the number of redundant readers detected. We compare the performance of RRE with a centralized greedy approximation algorithm for the redundant reader problem. The comparison is done in terms of the number of RFID readers able to be turned off simultaneously. The centralized greedy algorithm, GREEDY, sequentially selects the unvisited RFID reader with the highest density of covered, unvisited RFID tags. It then marks the selected RFID reader and its covered RFID tags as visited. GREEDY stops when there are no more unvisited tags. The set of visited RFID readers remain active and the others can be safely deactivated. GREEDY is safe, in the sense that deactivated RFID readers will not leave tags uncovered (see Section V-B). The GREEDY algorithm is however difficult to implement, since it requires centralized knowledge of the reader network.

In the first experiment we randomly place 500 RFID readers and between 1000 and 8000 RFID tags in the $1000 \times 1000\ m^2$ domain. Figure 3 shows the number of redundant RFID readers discovered by RRE and GREEDY. For fewer RFID tags deployed, RRE is reasonably close to GREEDY, by discovering 83% of the redundant readers discovered by GREEDY. As the number of RFID tags increases, the performance of RRE relative to GREEDY degrades, but it always discovers over 50% of the redundant readers of GREEDY. Both GREEDY and RRE discover less redundant readers as the number of deployed RFID tags increases. Both algorithms base their decision on the number of RFID tags covered by readers. By increasing the RFID tag density, the distribution of RFID tags per reader becomes more uniform, making it more difficult to choose good, active RFID readers. However, the decrease is more acute for RRE, since in scenarios where readers whose interrogation zones overlap cover equal numbers of tags, consistently breaking ties becomes a difficult problem. We illustrate such a scenario in Figure 4, where each of readers $\mathtt{R}_2$, $\mathtt{R}_3$ and $\mathtt{R}_4$ covers four tags. While the optimal solution requires only $\mathtt{R}_2$ and $\mathtt{R}_4$ to be active, we can imagine a run of RRE where $\mathtt{R}_4$ locks $\mathtt{T}_5, .., \mathtt{T}_7$, $\mathtt{R}_3$ locks $\mathtt{T}_3$ and $\mathtt{T}_4$ and $\mathtt{R}_2$ locks $\mathtt{T}_1$ and $\mathtt{T}_2$, effectively requiring all three readers to be active. The example can be easily extended, and one can see that in the worst case RRE can require $2\mathtt{r} - 1$ active readers,

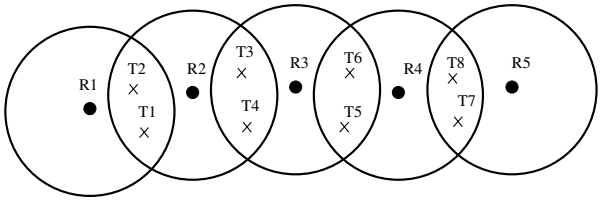

Fig. 4. Difficulty of consistently breaking ties. The optimal solution keeps only $\mathtt{R}_2$ and $\mathtt{R}_4$ active. However, in a scenario where $\mathtt{R}_2$, $\mathtt{R}_3$ and $\mathtt{R}_4$, each covering 4 tags, lock a different set of tags, all of them will have to be active.

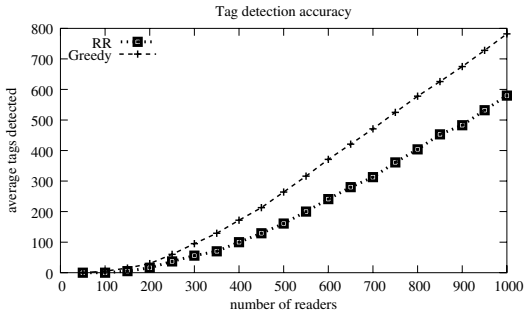

Fig. 5. Number of redundant readers discovered by RRE and Greedy when the number of RFID readers randomly deployed increases from 50 to 1000, for a total of 4000 RFID tags.

where $\mathtt{r}$ would be sufficient. This degenerate worst case is, however, rare. Moreover, as noted before, the performance of GREEDY comes with the high cost of collecting all reader network information at a central point.

The second experiment compares the performance of RRE and GREEDY when the number of randomly deployed RFID readers increases from 50 to 1000, when the total number of RFID tags is 4000. Figure 5 shows the results of this experiment. For scarce deployment of RFID readers, very few of the readers are redundant. As their density increases, however, so does the number of redundant readers. For example, for 1000 RFID readers, GREEDY discovers almost 800 to be redundant. While initially RRE is very accurate, as the number of RFID readers increases, RRE discovers fewer redundant readers. However when between 500 to 1000 readers are deployed, RRE consistently discovers more than 80% of the redundant readers of GREEDY. The difference is again due to the difficulty in breaking ties in RRE. As the number of deployed RFID readers increases, the number of readers whose interrogation zones overlap, also increases, generating more contentions.

The final experiment measures the dependency between the number of redundant readers discovered by RRE and GREEDY and the interrogation zones of RFID readers. We randomly deploy 500 RFID readers and 4000 RFID tags, and increase the interrogation radius of readers from 40 to 100m.

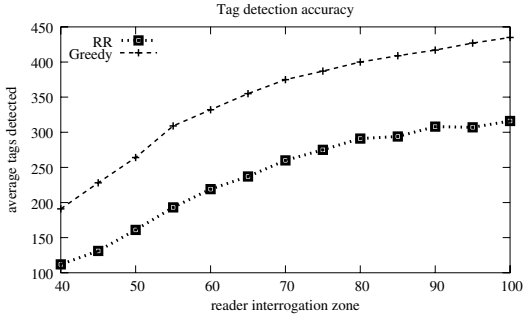

Fig. 6. Number of redundant readers discovered by RRE and Greedy when the interrogation radius of RFID readers increases from 40 to 100m. The number of RFID readers is 500 and the number of RFID tags is 4000, for the entire duration of the experiment.

Figure 6 shows that as expected, with the increase in the interrogation radius of RFID readers, both RRE and GREEDY discover an increasing number of redundant readers. This is because active readers cover larger areas, effectively necessitating fewer active readers to cover all the tags. Note that while RRE discovers fewer redundant readers than GREEDY, the difference is almost constant for smaller interrogation zones. Due to an increase in the number of interrogation zone overlaps, leading to an increased difficulty of breaking ties, the difference between GREEDY and RRE increases slightly for large interrogation zones.

VII. Related Work

The reader-collision problem in RFID systems was first documented in [15]. The solution proposed, of allocating different frequencies to interfering RFID readers, is centralized. A simple decentralized version, where readers listen for collisions and use randomized backoff when detecting one, is discussed. In contrast, our work assigns different time slots for transmitting RFID readers. Moreover, our solution guarantees w.h.p. that each RFID reader is able to correctly read all the RFID tags placed in its interrogation zone.

Perhaps closest to our goal of correctly reading covered RFID tags is the work of Waldrop et. al [16]. They propose Colorwave, a decentralized MAC protocol for RFID reader networks whose purpose is to allocate disjoint time slots for reader transmissions. The protocol is based on the presence of an interference graph whose links denote interference between the end-points corresponding to RFID readers. Hence, an interesting extension to this work would be a description of the interference graph construction. As shown in Figure 4, interference at certain RFID tags is difficult to detect, since even the presence of such tags may not be known.

A related problem occurs in ad-hoc networks, when nodes attempt reduce their energy consumption during the process of discovering their adjacent neighbors. McGlynn and Borbash [17] introduce a family of "birthday protocols" using random independent transmissions to discover neighbors. Tseng et. al [18] propose three protocols that employ the power saving mode (PSM) of 802.11 cards in order to distributively allow nodes to switch to a low-power sleep mode and still discover their neighbors, even in highly mobile environments. Zheng et. al [19] formulate the problem of asynchronous wakeup of nodes as a block design problem in combinatorics. Asynchronous wakeup allows nodes to alternate between effectively relay packets and sleep, without using synchronized clocks. We emphasize that the main difference between the problem of reducing energy consumption while keeping track and communicating with adjacent neighbors in ad-hoc networks and the problem of accurately discovering tags in the vicinity of readers resides in the balance of resources of nodes in the classic ad-hoc networks. In our approach tags are assumed to be passive and severely resource constrained.

The problem of coverage of a set of entities has been studied in a variety of contexts. In the area of wireless sensor networks, Tian and Georganas [7] present an algorithm for detecting sensors whose coverage area is completely covered by other sensors. A sensor turns itself off only when each sector of its coverage disk is covered by another sensor. Zhang and Hou [8] provide a distributed algorithm for extending the network lifetime by turning off redundant sensors. Their mechanism for deciding a sensor to be redundant requires a sensor to divide its coverage area into small grids and then using a bitmap to indicate whether the center of each square of the grid is covered by some other sensor. Ye et al. [9] present an algorithm that extends the network lifetime by maintaining a necessary set of working sensors and turning off redundant ones. A sensor is alternatively sleeping or active. When a sensor wakes up, if it has an active sensor inside its transmission range, it turns off again. Slijepcevic and Potkonjak [20] introduce a centralized algorithm for finding the maximum number of disjoint subsets of sensors, where each subset completely covers the same area as the entire set of sensors. All the above work defines coverage in terms of continuous areas. Instead, our goal is to detect a discrete set of points in the coverage area of a reader network. Moreover, we define coverage only in terms of the set of discrete points, tags. While this approach has the potential to discover more redundant readers, the problem is complicated by the scarce resources of tags.

Medium Access Control protocols for wired and wireless networks share several details with our reader collision avoidance algorithm. The first MAC protocol, proposed for packet radio networks, is ALOHA [21]. When the transmission of a node results in collision, the node must wait for a random interval before retransmitting. However, RFID systems do not have the mechanisms to detect collisions occurring at tags, making ALOHA unsuitable for avoiding reader collisions. Multiple access with collision avoidance (MACA) [22] is a protocol that employs a handshake to avoid hidden-node problems. The sender broadcasts an RTS message and the receiver replies with a CTS message. All the nodes that hear the RTS and CTS messages delay their transmissions. Such a protocol cannot be used in RFID system, since the purpose of an RFID reader is to detect *all* the RFID tags in its interrogation zone. Such a reader does not know the identities of the RFID tags and thus cannot send individual RTS messages. Moreover, the simultaneous reception of CTS messages initiated by RFID tags leads to tag collision problems. Carrier sensing multiple access with collision detection (CSMA/CD) [23], employed in the standard Ethernet is based on the ability of nodes to detect collisions. Upon detecting a collision, a node waits for a random interval before retransmitting. In case of subsequent collisions, the node waits twice as long before attempting to retransmit, also known as exponential back-off. However, as noted before RFID systems lack the ability of detecting remote collisions.

Privacy-related issues of RFID systems have been extensively studied in [12], [24], [25]. A detailed description of computation and communication mechanisms and constraints of RFID systems, together with several suggestions for RFID protection are presented in [12]. A solution for preserving the

privacy of RFID tags, using hash functions for locking tags, is proposed in [24]. Locked tags are prevented from revealing their unique identifier until unlocked with the corresponding inverse hash value. The work in [25] provides an in-depth presentation of security and privacy challenges of RFID systems and proposes the use of additional, "blocker" RFID tags in order to prevent unauthorized RFID readers from accessing protected RFID tags.

VIII. Conclusions

In this paper we address the problem of extending the lifetime of the reader network by detecting and temporarily turning off redundant readers. We define redundancy in terms of discrete sets of points, RFID tags, and prove that the optimization version of the problem is NP-complete. We present distributed and localized algorithms, based on a randomized querying technique, that ensures, w.h.p., the accurate receipt of reader queries by RFID tags. We provide a probabilistic analysis of the algorithms. Our extensive experiments show that our redundant reader elimination heuristic is efficient, as compared to a centralized greedy approximation of the optimum solution.

Acknowledgements

Hoffmann has been supported in part by NSF grants DMS-013098,DCNS-0216131, DHER-0227828, DSC-0325227, DCMS-0443148, and by an IBM Faculty Scholar award. The authors would like to thank Suresh Jagannathan, Ronaldo Ferreira and Asad Awan for useful discussions and pointers. We would also like to thank the anonymous reviewers for their helpful suggestions.

References

[1] S.E. Sarma. Towards the five-cent tag. Technical Report MIT-AUTOID-WH-006, MIT Auto ID Center, 2001.

[2] Edwin Kalischnig. RFID: Making sense of sensor-based technology. Manufacturing and Logistics IT, July 2004.

[3] Ann Bednarz. Wireless technology reshapes retailers. Network World, 12 August 2002.

[4] SkyeTek. http://www.skyetek.com/readers_Mini.html, January 2004.

[5] SmartCode. http://www.smartcodecorp.com/.

[6] Battery Assisted Passive Microwave RFID tags. http://www.alientechnology.com/products/rfidbattery/bap_tags.php.

[7] Di Tian and Nicolas D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the 1st ACM WSNA*, pages 32–41. ACM Press, 2002.

[8] Honghai Zhang and Jennifer Hou. Maintaining coverage and connectivity in large sensor networks. In *International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad hoc Wireless and Peer-to-Peer Networks*, Feb 2004.

[9] Fan Ye, Gary Zhong, Songwu Lu, and Lixia Zhang. Peas:a robust energy conserving protocol for long-lived sensor networks. In *23rd IEEE ICDCS*, 2003.

[10] B. Carbunar, A. Grama, J. Vitek, and O. Carbunar. Coverage-preserving redundancy elimination in sensor networks. In *IEEE SECON*, 2004.

[11] Murali K. Ramanathan, Bogdan Carbunar, Suresh Jagannathan, and Ananth Grama. Reader collision avoidance in rfid systems. Technical Report 05-014, Purdue University, 2005. http://www.cs.purdue.edu/homes/carbunar/rca.pdf.

[12] Sanjay E. Sarma, Stephen A. Weis, and Daniel W. Engels. RFID systems and security and privacy implications. In *CHES '02*, pages 454–469. Springer-Verlag, 2003.

[13] Henri E. Bal, Raoul Bhoedjang, Rutger Hofman, Ceriel Jacobs, Koen Langendoen, Tim Ruhl, and M. Frans Kaashoek. Performance evaluation of the Orca shared-object system. *ACM Trans. Comput. Syst.*, 16(1):1–40, 1998.

[14] R. Fowler, M. Paterson, and S. Tanimoto. Optimal packing and covering in the plane are np complete. *Information Processing Letters*, 12(3):133–137, 1981.

[15] D. W. Engels and S. E. Sarma. The reader collision problem. In *IEEE International Conference on Systems, Man and Cybernetics*, 2002.

[16] J. Waldrop, D.W. Engels, and S.E. Sarma. Colorwave: a MAC for RFID reader networks. In *Wireless Communications and Networking (WCNC)*, 2003.

[17] Michael J. McGlynn and Steven A. Borbash. Birthday protocols for low energy deployment and flexible neighbor discovery in ad hoc wireless networks. In *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, pages 137–145. ACM Press, 2001.

[18] Yu-Chee Tseng, Chih-Shun Hsu, and Ten-Yueng Hsieh. Power-saving protocols for ieee 802.11-based multi-hop ad hoc networks. *Comput. Networks*, 43(3):317–337, 2003.

[19] Rong Zheng, Jennifer C. Hou, and Lui Sha. Asynchronous wakeup for ad hoc networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 35–45. ACM Press, 2003.

[20] Sasa Slijepcevic and Miodrag Potkonjak. Power efficient organization of wireless sensor networks. In *IEEE ICC*, 2001.

[21] N. Abramson. The ALOHA system: Another alternative for computer communications. In *AFIPS Conf. Proc., Fall Joint Computer Conf*, 1970.

[22] P. Karn. MACA a new channel access method for packet radio. In *ARRL/CRRL Amatuer Radio 9th Computer Networking Conf.*, 1990.

[23] Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications. Institute of Electrical and Electronics Engineers, 1996.

[24] Stephen Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Security in Pervasive Computing*, pages 201–212. Lecture Notes in Computer Science, 2003.

[25] Ari Juels, Ronald L. Rivest, and Michael Szydlo. The blocker tag: selective blocking of RFID tags for consumer privacy. In *CCS '03: Proceedings of the 10th ACM Conference on Computer and communications security*, pages 103–111. ACM Press, 2003.