Vol. 5, No. 10 October 2014 ISSN 2079-8407
Journal of Emerging Trends in Computing and Information Sciences

Load-aware Reader Placement Algorithms for RFID Networks

Waleed Alsalih
Computer Science Department, King Saud University, Riyadh, Saudi Arabia
E-mail: wsalih@ksu.edu.sa

Abstract—**Radio Frequency Identification (RFID) technology has been used in a wide variety of tracking, monitoring, and access control applications. The spreading adoption of RFID and its large scale deployments have given importance to some planning and coverage problems in this field. Tags coverage in RFID networks is one of these problems that can be defined generally as finding the number and locations of RFID readers needed to identify a set of RFID tags in a particular area. Accurate coverage is crucial in RFID systems as failing to cover a tag may result in missing an important event and/or losing asset and revenue. Furthermore, several technical issues, such as balancing the load among readers and avoiding tags/readers interference, make this placement problem challenging and different from other classical placement problems. In this paper, we look into the problem of placing RFID readers to cover a set of RFID tags with the objectives of minimizing the number of readers, reducing overlapping among readers interrogation ranges, and balancing the load among readers.**
We propose an approximation algorithm for the problem of covering all tags using the minimum number of readers. We then extend this algorithm to add the objective of reducing overlapping and interference among the coverage areas of readers. We finally present two load balancing algorithms that evenly distribute the tags identification load among different readers. The first load balancing algorithm assumes a fixed interrogation range for the RFID reader and the second one assumes a dynamic interrogation range. We also present our RFID planning tool which is a software that implements our algorithms to produce detailed plans for RFID networks. And finally, we present comprehensive simulations to study the performance of our algorithms.

Keywords: *RFID, approximation algorithms, load balancing, reader placement, set cover*

1 Introduction

Radio Frequency Identification (RFID) is a spreading wireless technology that is being used in a wide range of tracking, monitoring, and access control applications. An RFID system is composed of one or more RFID readers, a set of tags, and an application. An RFID reader interrogates and identifies objects located within its coverage (interrogation) range by reading the data stored in tags attached to these objects. RFID systems require full coverage as well as fast identification. Having accurate full coverage is essential for many RFID applications. In a self-checkout counter in a store, for instance, missing a tag is simply a loss of revenue. Several factors influence coverage in RFID networks; some of these factors are density of tags, size of the interrogation area, number of deployed readers, and overlapping among readers [1].

The total number of deployed readers directly affects the cost and complexity of the RFID system. Coverage in RFID systems, and in many other systems, is an optimization problem that needs to be dealt with carefully. While deploying too many readers may guarantee full coverage, it has the side effect of significant interference among readers, which results in more reader-to-reader collisions and, hence, a lower performance for the whole system. Furthermore, the number of tags within each reader's interrogation area determines the load and, hence, the identification process delay of that reader; and the overall identification delay of the whole system is determined by the longest delay associated with a single reader. Therefore, reducing overlapping and balancing the load among readers are essential to any coverage scheme in RFID systems.

Some RFID coverage schemes have been proposed in the literature. These schemes can be classified into two approaches: planned coverage and ad-hoc coverage. In the planned coverage approach, most of the coverage schemes are designed to find an optimal placement in which RFID readers are placed at the vertices of a grid [2]–[4], at the vertices of a honey grid [5], or at a predetermined set of locations [6]. While these schemes are able to achieve good coverage, that comes at the cost of using a large number of RFID readers and, hence, results in the previously mentioned problems associated with having too many readers. In the ad-hoc approach, readers are deployed randomly and an algorithm is executed afterwards to turn-off some redundant readers [7]. In most of these schemes, in order to eliminate redundancy or to manage the load, some state information is maintained within each tag [7]–[9]. Our objective, in this paper, is to have planned deployment schemes which guarantee full tag coverage using a reasonably small number of readers with a minimum overlapping among readers' interrogation areas and with a balanced workload amongst readers. Furthermore, we do not require tags to maintain any state information.

In this paper, we study the tags' coverage problem in RFID systems and propose a set-cover based approximation algorithm to solve it. We present an approximation algorithm that finds a relatively small number of readers and their locations to cover a set of tags. Our algorithm reduces the overall network cost by reducing the number of readers. It also improves the system performance by reducing interference and overlapping among readers coverage regions and, hence, having less collisions among readers. Moreover, our algorithm enhances the overall system reading rates by balancing the identification load over the deployed readers; since readers read tags in parallel, by minimizing the maximum delay encountered by a single reader we minimize the overall delay of the system.

We also present our RFID planning tool which is a software that implements our algorithms to produce detailed plans for RFID networks. We also evaluate the proposed algorithms via comprehensive simulations and we compare them with other schemes available currently in the literature. The simulation results have shown that the proposed approximation algorithm is effective in minimizing the number of readers, reducing their overlapping regions, and maintaining a good load balance.

The remainder of the paper is organized as follows. Section 2 surveys previous work related to RFID reader placement. Section 3 outlines the adopted model and its underlying assumptions. Section 4 describes the proposed reader placement algorithms. Section 5 presents and analyzes our simulation results. Section 6 presents our RFID planning tool that applies the proposed placement algorithms. Finally, section 7 concludes this paper by highlighting some future research directions.

2 Background

Several RFID placement schemes exist in the literature and they can be classified into two approaches: planned coverage and ad-hoc coverage [10].

Most of the planned coverage schemes are designed to find the optimal placement of readers in terms of either maximal coverage or minimum number of readers, and under some constraints on where readers can be placed [2]–[6], [11], [12]. In the optimal grid coverage approach [2], readers are to be placed only on vertices of a grid in which the length of edges is determined by the interrogation range (distance) of readers. After deployment, readers which are not covering any tag can be hibernated. Another similar approach is the honey grid in which readers are deployed in rings, of different sizes, centered at the center of the interrogation zone [5]. When rings are numbered based on their radii, the i^{th} ring contains $6i$ readers. Similar to the regular grid approach, after deployment, readers which are not covering any tag can be hibernated. While these two schemes can provide full area coverage, they usually result in deploying a large number of readers, significant readers collisions, and nonuniform load distribution. Some reader deployment schemes that take into account tags and readers orientation have been proposed [6]. The authors in [6] propose a scheme that finds the optimal number of readers, their locations out of a set of predefined locations, and their antennas orientation to maximize readability. However, this scheme, like other schemes in the literature, considers placement in a set of predefined locations and without load balancing. Our proposed scheme is not limited to a predefined set of locations and it is able to achieve full coverage of tags, and with a minimal overlapping and a better load balancing.

On the other hand, ad-hoc coverage schemes assume a dense deployment of readers and aim at detecting and hibernating redundant readers afterwards; these schemes are also called redundant reader elimination schemes [7]–[9]. The authors of [7] propose the RRE scheme in which each reader broadcasts its unique ID and its tag count, which is the number of tags within its interrogation region, to all tags within its interrogation region. Each tag keeps track of the ID of the reader that has the maximum tag count among readers it received messages from. Access to a particular tag is granted to the reader whose identity is stored in the tag. Readers which are not granted access to any tag are marked redundant and are hibernated. The RRE scheme involves significant communication overhead as it incurs many write operations to the tag. To overcome this problem, the LEO and the LEO+RRE schemes use the first-read first-own policy by which the RFID reader that manages to identify a tag first is granted access to that tag [9]. In case of the LEO+RRE algorithm [9], upon completion of LEO execution, the RRE algorithm is then used to further eliminate any redundant readers. Various other redundant reader elimination schemes have been proposed in the literature and they try to make enhancements over the RRE and LEO schemes [8], [13]. In [14], the authors have proposed a TREE 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 NULL which means that the reader is granted the ownership, or with an identity which is different from the query's embedded identity to tell the reader that the tag's ownership has already been assigned to another reader. A reader without any tag's ownership is marked as redundant and it is hibernated. Although the scheme is light-weight, it is not effective in eliminating all redundancies as compared with the LEO+RRE scheme for example. In [15], the authors introduced the idea that a reader with a smaller number of neighbours has a lower probability of interfering with other readers and, hence, should be selected (i.e., not hibernated). Using a cost function composed of the tags count and the number of neighbours, the chance that a particular reader is redundant is estimated. Another scheme with a different cost function is presented in [16].

The authors in [17] propose a load balancing scheme in which each reader writes its tag count into the tag memory, and the tag picks and responds back to the reader with the lowest load. Other load balancing schemes have also been proposed in the literature [18]. These ad hoc schemes are generally designed for environments having thousands of small readers that are pre-deployed in an ad-hoc manner over large areas and with the main objective of reducing energy consumptions of the deployed readers. Our research objective is to have a planned, pre-operation deployment strategy which guarantees full coverage of tags, uses a small number of readers, and also achieves a good performance by balancing the load and minimizing the overlapping among readers.

3 Assumptions and definitions

We assume an RFID system composed of n passive tags $\{t_1, t_2, \ldots t_n\}$ 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 read by a reader if and only if the distance between t_i

and the reader is at most r_i. Tags are assumed to be placed in a 3D space modeled as a rectangular cuboid, and readers are to be placed on particular sides of the rectangular cuboid; from now on, we use the term deployment faces to refer to those sides. This space can be looked at as a room and the readers are placed on some walls of that room. This environment is illustrated in Fig. 1.

The problem we address in this paper can be defined as follows:
Given the number of tags and their 3D locations, find the minimum number of RFID readers and their exact locations such that all tags are covered.
While the locations of tags are assumed to be known in this problem, we show later how to accommodate fluctuations in tags locations.

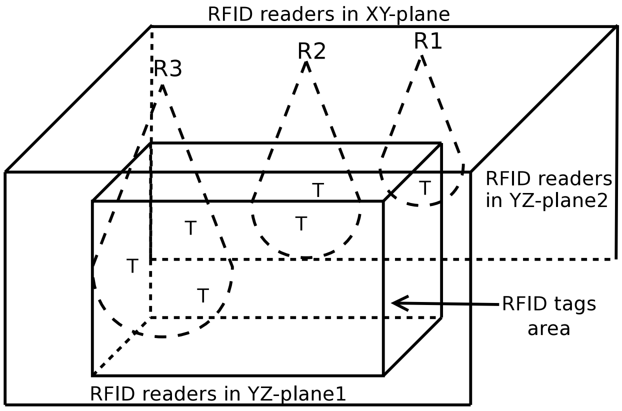

Fig. 1. 3D Coverage in RFID Networks.

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

- $T = \{t_1, t_2, \ldots, t_n\}$ is the set of tags.
- p_i is a point representing the location of a tag t_i; $p_i.x$, $p_i.y$, and $p_i.z$ are the $x-$, $y-$, and $z-$ coordinates, respectively.
- r_i is the transmission range of a tag t_i.
- $|S|$ is the cardinality of a set S.
- $|p, q|$ is the Euclidean distance between two points p and q.
- For any point q in the 3D space, $Cover(q)$ is the set of all tags that have q within their transmission ranges (i.e., $Cover(q) = \{t_i : |p_i, q| \leq r_i\}$).
- For any point q in the 3D space, $CoverSorted(q)$ is $Cover(q)$ with tags sorted in a nondecreasing order based on their distance to q.
- For any point q in the 3D space, $SubCover(q, \beta)$ is the first β elements in $CoverSorted(q)$.

The algorithms we propose here are based on a well known approximation algorithm for the set covering problem which is known to be NP-hard [19]. For the sake of completeness, we give a formal definition for the set covering problem as follows:

An instance of the set covering problem consists of a set $\mathcal{C}$ of a finite number of elements and a set $\mathcal{F}$ whose elements are subsets of $\mathcal{C}$ with the condition that each element $c \in \mathcal{C}$ belongs to at least one subset $f \in \mathcal{F}$; when $c \in f$, f is said to cover c. The problem is to find a minimum size set $\mathcal{G} \subseteq \mathcal{F}$ such that each element in $\mathcal{C}$ is covered by at least one element in $\mathcal{G}$ [19].

4 RFID reader deployment algorithms

In this section we present our RFID reader deployment algorithms. We start with an algorithm that guarantees full coverage with the near minimum number of readers. We then extend and modify that algorithm to minimize overlapping among readers coverage areas. Finally, we present a deployment scheme that uses our algorithm to give load-balancing deployment plans.

4.1 Providing coverage

To provide full tags coverage we need to place at least one RFID reader within the transmission range of every tag. Since we assumed that RFID readers are placed only on the deployment faces of the rectangular cuboid, we need to find the intersection of the transmission spheres of tags and the planes representing the deployment faces of the rectangular cuboid. When a sphere intersects with a plane in a 3D space, their intersection is a circle on that plane. For example, a sphere centered at the point (a, b, c) with a radius of r intersects with the plane $z = 0$ if and only if if $r^2 - c^2 \geq 0$; and when they intersect, their intersection is a circle centered at the point $(a, b, 0)$ with a radius of $\sqrt{r^2 - c^2}$. Let us call such a circle a *side circle*. Each tag has at most one side circle on each side of the rectangular cuboid. In order to cover a particular tag t_i, at least one reader must be placed within one of the side circles associated with that tag.

To minimize the number of deployed readers while covering all tags, each reader should be placed in a location to cover as many tags as possible. Therefore, readers should be placed in areas where several side circles, associated with different tags, overlap. Our algorithms consider these overlapping regions as the candidate locations for readers. Furthermore, we focus on *intersection points* which are the points where the boundaries of side circles intersect. These intersection points are actually representatives to all possible overlapping regions (for more details on this, see our earlier work in [20]). Thereby, locations of RFID readers will be limited to intersection points of the side circles and, hence, the problem becomes a discrete optimization problem. Moreover, an instance of the RFID reader deployment problem can be reduced to an instance of the set covering problem in which $\mathcal{C} = T$ (i.e., the set of tags) and $\mathcal{F} = \{Cover(q) : q \text{ is an intersection point}\}$.

Thereby, the greedy approximation algorithm for the set covering problem can be used to solve the RFID reader deployment problem. For the sake of completeness, the greedy approximation algorithm is shown in Algorithm 1 [19].

This greedy algorithm has an approximation ratio of $\ln |C| + 1$ and a time complexity of $O(|C||\mathcal{F}| \text{ MIN}(|C|, |\mathcal{F}|))$.

Algorithm 1: The greedy set covering algorithm

Set Cover($\mathcal{C}$,$\mathcal{F}$)
$U = \mathcal{C}$;
$V = \phi$;
while $U \neq \phi$ **do**
    Find a set $S \in \mathcal{F}$ that maximizes $U \cap S$;
    $U = U - S$;
    $V = V \cup \{S\}$;
end
return V;

4.2 Reducing overlapping among readers

Reader-to-reader interference, which is caused by overlapping among interrogation ranges of RFID readers, results in readers collisions and, hence, an additional delay to the interrogation process. The general set covering greedy algorithm does not take into account overlapping among subsets; the sole objective therein is to minimize the number of subsets needed to cover all elements. For example, let $\mathcal{C} = \{1, 2, 3, 4, 5\}$ and $\mathcal{F} = \{\{1, 2, 3\}, \{2, 3, 4\}, \{1, 2, 5\}, \{4\}, \{5\}\}$. The greedy algorithm may pick the set $\{\{1, 2, 3\}, \{2, 3, 4\}, \{1, 2, 5\}\}$ to cover all elements with a minimum number of subsets, which is an optimal solution for this instance of the set covering problem but with significant overlapping. However, another optimal solution with no overlapping is $\{\{1, 2, 3\}, \{4\}, \{5\}\}$, which is much more suitable for reader deployment in RFID networks. We modify the greedy set covering algorithm (Algorithm 1) to make solutions with less overlapping preferred without affecting the cardinality of the covering sets.

In the greedy set covering algorithm, the covering set is built incrementally; it goes through several steps and at each step, it picks a subset that covers the maximum number of uncovered elements, and this continues until all elements are covered. This can be viewed as assigning a *weight* to each remaining subset which is equal to the number of uncovered elements it covers, and picking the one with the maximum weight at each step. The weight of a subset can be calculated in a different way that gives credit to those subsets that do not cover many already covered elements. Let U be the set of uncovered elements and $weight(S)$ be the weight of a subset S. In the general greedy set covering algorithm, $weight(S) = |U \cap S|$. We can modify this algorithm so that among subsets covering the maximum number of uncovered elements, it picks the one that covers the minimum number of already covered elements. This can be achieved by the following weight formula:
$weight(S) = |U \cap S| - \alpha(|S| - |U \cap S|)$, $0 < \alpha \leq 1/n$.
The greedy algorithm for set covering with less overlapping is shown in Algorithm 2.

When $0 \leq \alpha \leq 1/n$, the approximation ratio of $\ln \mathcal{C} + 1$ still holds for the weighted set covering algorithm. This is because both the weighted set covering algorithm and the general greedy set covering algorithm pick the same subset at each step except when there is a tie (i.e., several subsets covers the same maximum number of uncovered elements). In the case of having a tie, the weighted set covering algorithm

Algorithm 2: Weighted Approximation Set Cover (ASC) algorithm to minimize overlapping amongst subsets

ASC($\mathcal{C}$,$\mathcal{F}$,α)
$Z = \mathcal{F}$;
$U = \mathcal{C}$;
$V = \phi$;
while $U \neq \phi$ **do**
    foreach *subset* $S \in Z$ **do**
        $Weight(S) = |U \cap S| - \alpha(|S| - |U \cap S|)$;
    end
    Find a set $S \in Z$ that maximizes $Weight(S)$;
    if $Weight(S) \leq 0$ **then**
        /* This means that the set C can not be covered by subsets in $\mathcal{F}$ */
        return ϕ and exit;
    end
    $U = U - S$;
    $Z = Z - \{S\}$;
    $V = V \cup \{S\}$;
end
return V;

picks a subset that covers the minimum number of already covered elements. However, the approximation ratio of the general greedy algorithm holds regardless of how ties are dealt with.

4.3 Load balancing with static interrogation ranges

The two algorithms presented in the previous subsections have the objectives of reducing the total number of RFID readers and reducing overlapping among different readers. However, the issue of load balancing is not considered yet. Therefore, we may end up with solutions in which readers have significant variations in their identification workloads, where the workload of a reader is proportional to the number of tags it covers. This lowers the performance[1] of the whole system. This is actually because each RFID reader is assigned a subset of tags to identify and readers interrogate their assigned tags in parallel. Thus, the performance of the whole system is determined by the reader with the maximum workload because it will be the last to finish identifying its tags. For example, a system with two RFID readers each is covering 50 tags is much better than a system with two readers one is covering 90 tags and one is covering 10 tags.

We propose a load balancing algorithm that evenly distributes the workload among RFID readers. The main idea here is to put an upper bound on the number of tags β that can be covered by a single reader (e.g., a reader can not cover more than 10 tags). The smaller β is, the more balanced load we get. However, the objective of minimizing the number of deployed readers and that of balancing the load (i.e., a small β) conflict with each other. The smallest possible value for

[1]The performance herein is defined as the time needed to interrogate all tags.

Journal of Emerging Trends in Computing and Information Sciences

β is 1, which means that each reader covers a single tag but that will result in deploying too many readers. On the other hand, the minimum number of deployed readers to provide full coverage may be one reader covering all tags, which results in the maximum value for β and, hence, a low performance. This can be dealt with either by putting a constraint on β and finding the minimum number of readers meeting that constraint, or by putting a constrain on the maximum number of readers R and finding the minimum value of β meeting that constraint. The former option is straight forward and can be done by excluding intersection points whose coverage exceeds β (i.e., when $|Cover(q)| > \beta$, $Cover(q)$ is excluded from $\mathcal{F}$). The latter option is less trivial. In fact, one may need to try all possible values of β and pick the minimum value that satisfies the condition of having at most R readers. That can be done more efficiently by doing a binary search over all possible values of β (i.e., [1..MAX$_{S\in\mathcal{F}}|S|$]). This method is illustrated in Algorithm 3. The overall time complexity of this algorithm is $O(\log(\text{MAX}_{S\in\mathcal{F}}|S|)|C||\mathcal{F}|\ \text{MIN}(|C|,|\mathcal{F}|))$.

Algorithm 3: Approximation Set Cover algorithm with Load Balancing (ASC-LB)

ASC-LB(T,α, R)
$C = T$;
$Q =$ the set of all intersection points;
$\mathcal{F} = \{Cover(q) : q \in Q\}$;
$\beta_{max} = \text{MAX}_{S\in\mathcal{F}}|S|$;
$\beta_{min} = 1$;
while $\beta_{max} > \beta_{min}$ **do**
    $\beta = \lfloor \frac{\beta_{max}+\beta_{min}}{2} \rfloor$;
    $Z = \{S : S \in \mathcal{F} \wedge |S| \leq \beta\}$;
    $V =$ASC($\mathcal{C}$,Z,α);
    if $0 < |V| \leq R$ **then**
        $\beta_{max} = \beta$;
    else
        $\beta_{min} = \beta + 1$;
    end
end
$Z = \{S : S \in \mathcal{F} \wedge |S| \leq \beta_{max}\}$;
$V =$ASC($\mathcal{C}$,Z,α);
if $0 < |V| \leq R$ **then**
    return V;
else
    return ϕ;
end

4.4 Load balancing with dynamic interrogation ranges

Most of today's commercially available RFID readers come with dynamic (variable) transmission ranges (e.g., SkyeTek/M10 RFID reader [21]). This makes it possible to tune the size of the interrogation area of those readers. We make use of this feature to achieve better load balancing. Instead of excluding an intersection point because it covers too many tags, we reduce the transmission range associated with that intersection point until it covers the maximum allowed number of tags only (i.e., it covers exactly β tags). When a reader is placed at an intersection point, its range is set to the range associated with that intersection point. This gives more options for the locations of readers which boost the chances to find better solutions by the approximation algorithm.

This method is illustrated in Algorithm 4. The overall time complexity of this algorithm is the same as that of Algorithm 3 (i.e., $O(\log(\text{MAX}_{S\in\mathcal{F}}|S|)|C||\mathcal{F}|\ \text{MIN}(|C|,|\mathcal{F}|))$).

Algorithm 4: Approximation Set Cover algorithm with load balancing using Dynamic interrogation Range (ASC-DR)

ASC-DR(T,α, R)
$C = T$;
$Q =$ the set of all intersection points;
$\mathcal{F} = \{Cover(q) : q \in Q\}$;
$\beta_{max} = \text{MAX}_{S\in\mathcal{F}}|S|$;
$\beta_{min} = 1$;
while $\beta_{max} > \beta_{min}$ **do**
    $\beta = \lfloor \frac{\beta_{max}+\beta_{min}}{2} \rfloor$;
    $Z = \{SubCover(q, \beta) : q \in Q\}$;
    $V =$ASC($\mathcal{C}$,Z,α);
    if $0 < |V| \leq R$ **then**
        $\beta_{max} = \beta$;
    else
        $\beta_{min} = \beta + 1$;
    end
end
$Z = \{SubCover(q, \beta_{max}) : q \in Q\}$;
$V =$ASC($\mathcal{C}$,Z,α);
if $0 < |V| \leq R$ **then**
    return V;
else
    return ϕ;
end

4.5 Displaced tags

Our algorithms are designed with the assumption that exact locations of tags are known. However, it is expected in real life practice to have some location fluctuations (i.e., some tags are displaced from their planned locations). Fortunately, this can be easily accommodated in our scheme by reducing the actual transmission ranges of tags by a value of d units which makes our scheme resilient to any displacement of at most d units from the planned locations.

5 Performance evaluation

In this section, we show the results of the simulations we conducted to evaluate the performance of our placement algorithms. We compare and evaluate different schemes based on the number of deployed readers and the maximum workload assigned to a single reader. Our algorithms are evaluated with reference to a general grid based scheme in which readers are deployed on the vertices of a grid on the deployment faces. This grid based scheme is a modified version of two existing reader deployment schemes which are the optimal grid coverage and the honey grid coverage [2] [4] [5].

5.1 Simulation setup

We have developed a software to simulate the RFID environment and we have implemented and compared our schemes with the general grid based scheme. The grid based scheme is a modified version of the optimal grid coverage and the honey grid coverage which are known of their relatively high performance as compared with existing schemes in the literature [2] [4] [5]. The grid based scheme deploys readers on the vertices of a grid on the deployment face in a way that guarantees full area coverage, and then any reader that is not covering any tag is removed. In the rest of this section, Algorithm 2, Algorithm 3, and Algorithm 4 are referred to by the Approximate Set Cover (ASC) algorithm, the Approximate Set Cover with Load Balancing (ASC-LB) algorithm, and the Approximate Set Cover with Dynamic Range (ASC-DR) algorithm; respectively. The grid based scheme is referred to by the GRID scheme. We used two criteria to compare the RFID deployment schemes: the total number of deployed readers to cover all tags and the maximum workload assigned to a single reader (i.e., the maximum number of tags covered by a single reader), which is an indicator for how the load is distributed. A lower number of deployed readers leads to a lower cost, less overlapping among readers, and, hence, less reader collisions. And a lower value for the maximum workload amongst readers leads to a better performance and faster reading. We evaluated the different schemes under different scenarios, using different tags densities, and with various interrogation ranges.

Our simulation environment was configured as follows. RFID readers have an interrogation range of 10m. Tags are placed in a rectangular cuboid with a volume of $15\text{x}10\text{x}200\text{m}^3$. Readers are to be placed on two deployment faces of the rectangular cuboid: the face $x = 0$ and the face $x = 15$. Locations of tags are generated randomly inside the rectangular cuboid. Values of the performance criteria are averaged over twenty different simulation runs.

5.2 Simulation results

Fig. 2 shows the comparison results of different schemes in terms of the total number of deployed readers. The ASC algorithm outperforms all other schemes. On average, the ASC scheme reduces the total number of readers by a factor of 33% as compared to the GRID scheme. When the tags density grows and exceeds a certain threshold, the GRID scheme quickly reaches the maximum number of readers, which cover the whole area. On the other hand, this saturation point requires much higher tags densities when the ASC, the ASC-LB, or the ASC-DR algorithms are used. In extremely dense environments, all schemes have a similar performance in terms of the number of readers. This is because sufficiently large number of tags enforces all schemes to cover the whole area rather than to cover particular tags.

While the ASC algorithm deploys the smallest number of readers, it has the worst performance in terms of load balancing. Fig. 3 shows the simulation results of the different schemes with respect to the maximum load assigned to a single reader. Since the ASC-LB and the ASC-DR are the only schemes that consider load balancing, it is not a surprise that they outperform other schemes in this aspect. We noticed that improvements achieved by the ASC-LB and the ASC-DR schemes are magnified when the number of tags grows. It is interesting to find that the ASC-LB and the ASC-DR algorithms deploy a little more readers to have a better load balancing as compared to the ASC algorithm. We also noticed that the ASC-LB and the ASC-DR schemes have very similar performance in terms of both the total number of deployed readers and the load balancing.

Other simulations with different interrogation areas and different tag counts have shown similar results.

6 RFID planning tool

We have developed a planning tool which implements our algorithm to provide full coverage for the tags placed inside a room with a good interrogation performance. This tool also provides the user with an interactive 3D view of the room with the readers and tags placed inside it.

The tool is implemented using Java2 (Standard Edition) and Java 3D drawing libraries. It is platform-independent and it only requires the Java Virtual Machine to be installed on the system. The input to this tool includes the dimensions of the room and the physical 3D (x, y and z) coordinates of all tags placed in the room. When the (Design) option from the (RFID) menu is selected, a pop-up window prompts the user to enter the physical dimensions (width, height, and depth) of the room as shown in Fig. 4. The user also has to enter the reading range of the readers. The user can choose any side of the room (front, back, left, right, top, and bottom) as a deployment face.

The placement results are shown to the user in the form of a scaled 3D room showing locations of both tags and readers. A sphere around the reader shows the area and tags it covers (as shown in Fig. 4). The tool provides the option to save these views and results in different formats. The source code of this tool is available online at: http://faculty.ksu.edu.sa/Alsalih/NPST_RFID

7 Conclusion

RFID systems have a great potential to enable many monitoring, tracking, and automatic identification applications. Yet efficient design and operation schemes are still needed to make this technology cost-effective . Network planning and placement of RFID readers are two important problems in that direction. While some planning and placement schemes exist in the literature, some major performance issues have not yet received enough attention; examples of this are the coverage overlapping and load balancing among readers. In this paper, we look into the problem of placing a minimal

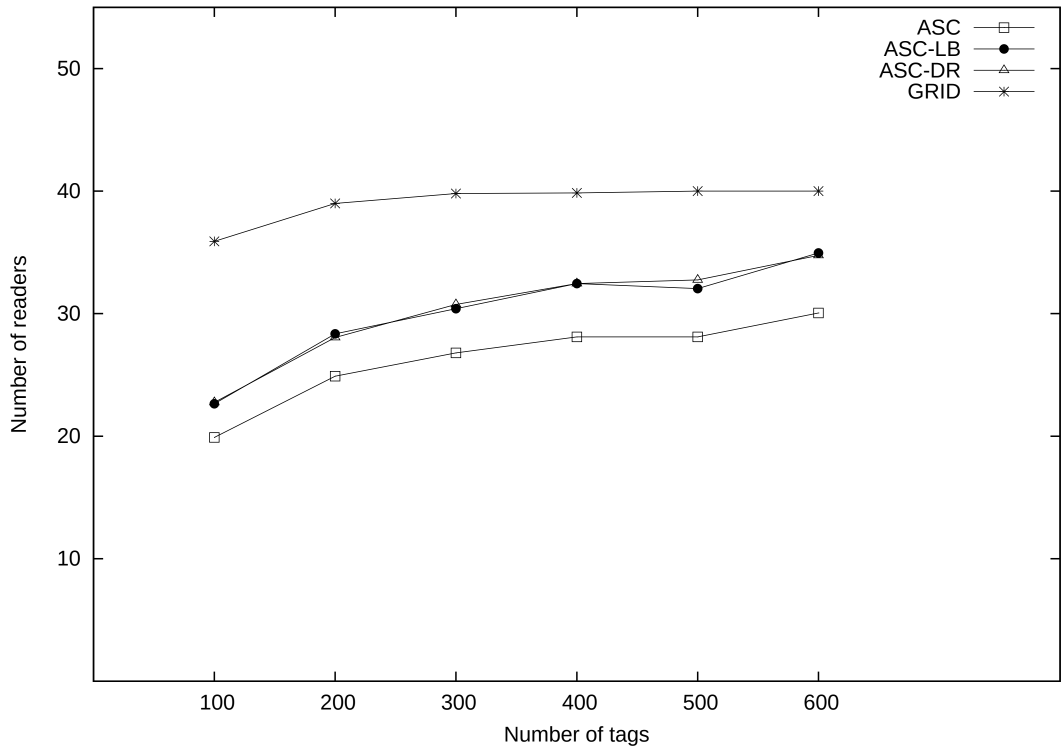

Fig. 2. Number of readers deployed by different schemes.

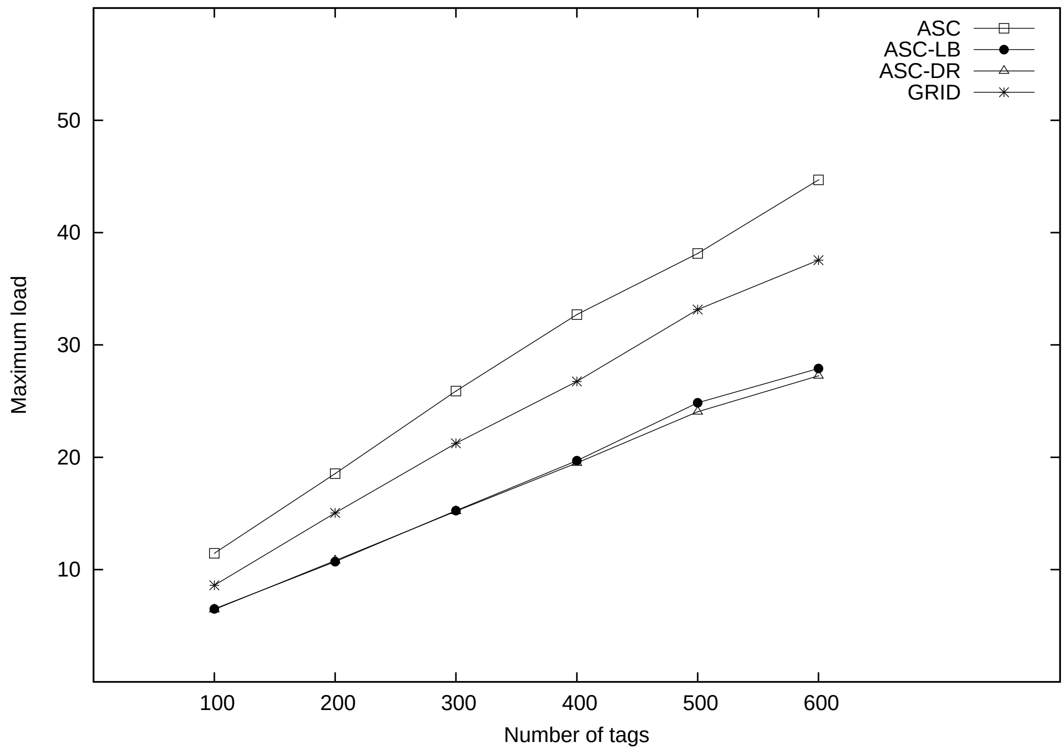

Fig. 3. Maximum load assigned to a single reader using different schemes.

Journal of Emerging Trends in Computing and Information Sciences

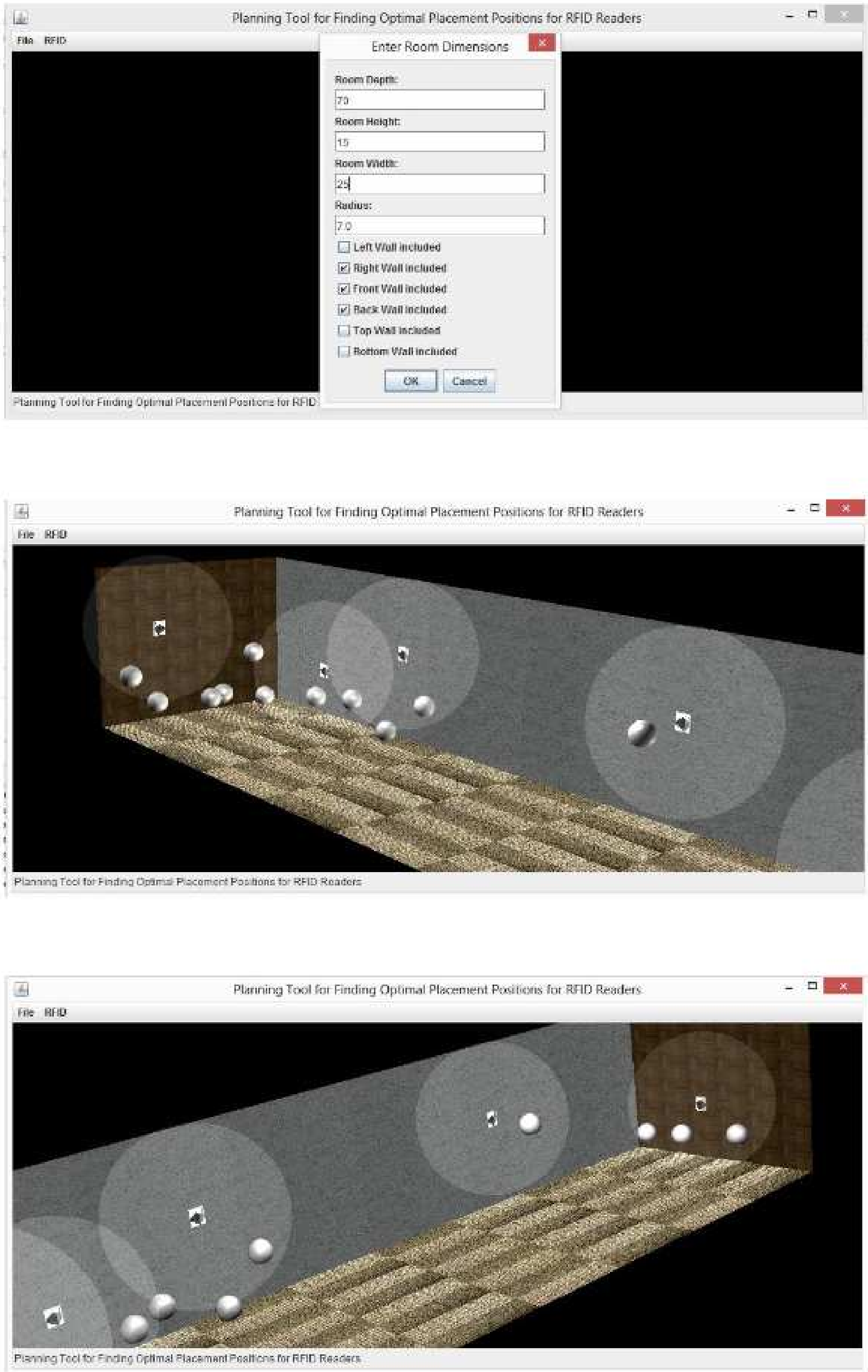

Fig. 4. Snapshots of the RFID planning tool.

number of readers to cover a set of tags with the objectives of minimizing the overlapping among readers interrogation areas and balancing their load. We present an approximation algorithm that finds a minimal number of readers and their locations in a 3D space. We also modify our algorithm to make it prefer solutions with less overlapping among readers interrogation zones. We further extend that algorithm to provide solutions with a balanced distribution of load among readers. Comprehensive simulations are used to evaluate the performance of our algorithms and to compare it with that of some existing deployments schemes.

In our future research, we plan to look into the problem of covering mobile tags that move along some trajectories with different velocities. We are interested in the objectives of minimizing the number of readers and maximizing the reliability of the system which can be defined as the probability of identifying a mobile tag.

Acknowledgement

This research is supported by the National Plan for Science and Technology Program at King Saud University, Project no. 11-INF1920-02.

The author likes to thank Dr. Hossam Hassanein and Dr. Kashif Ali from Queen's University, Canada, for their invaluable input and discussions about the work presented in this paper. He also likes to thank Dr. Kashif Ali and Mr. Saad Ijad from King Saud University, Saudi Arabia, for their support in the programming parts presented in this paper.

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] A. Oztekin, F. Mahdavi, K. Erande, Z. Kong, L. K. Swim, and S. T. S. Bukkapatnam, "Criticality index analysis based optimal RFID reader placement models for asset tracking," *International Journal of Production Research*, vol. 48, no. 9, pp. 2679–2698, 2010.

[3] A. W. Reza and T. K. Geok, "Investigation of indoor location sensing via RFID reader network utilizing grid covering algorithm," *Wirel. Pers. Commun.*, vol. 49, no. 1, pp. 67–80, 2009.

[4] A. Reza, T. Geok, and K. Dimyati, "Tracking via square grid of RFID reader positioning and diffusion algorithm," *Wireless Personal Communications*, pp. 1–24, 2010.

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

[6] L. Wang, B. A. Norman, and J. Ragopal, "Placement of multiple RFID reader antennas to maximize portal read accuracy," *International journal of Radio Frequency Identification Technology and Applications*, vol. 1, no. 3, pp. 260–277, 2007.

[7] 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.

[8] J.-W. Hung, I.-H. Li, H.-H. Lin, and J.-A. Cai, "The first search right algorithm for redundant reader elimination in RFID network," in *SEPADS'10: Proceedings of the 9th WSEAS international conference on Software engineering, parallel and distributed systems*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2010, pp. 177–183.

[9] 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.

[10] A. Alfagih, F. Al-Turjman, H. Hassanein, and W. Alsalih, "Coverage-based placement in RFID networks: An overview," in *Proceedings of the FTRA International Conference on Mobile, Ubiquitous, and Intelligent Computing*, 2012, pp. 220–224.

[11] Q. Guan, Y. Liu, Y. Yang, and W. Yu, "Genetic approach for network planning in the RFID systems," in *ISDA '06: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 567–572.

[12] H. Chen, Y. Zhu, and K. Hu, "Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning," *Appl. Soft Comput.*, vol. 10, no. 2, pp. 539–547, 2010.

[13] 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.

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

[15] 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.

[16] K. Ali, H. Hassanein, and W. Alsalih, "Using neighbor and tag estimations for redundant reader eliminations in RFID networks," in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, March 2011, pp. 832–837.

[17] Q. Dong, A. Shukla, V. Shrivastava, D. Agrawal, S. Banerjee, and K. Kar, "Load balancing in large-scale RFID systems," in *IEEE INFOCOM 2007. 26th IEEE International Conference on Computer Communications*, May 2007, pp. 2281–2285.

[18] H. S. Chae, J. G. Park, J. F. Cui, and J. S. Lee, "An adaptive load balancing management technique for RFID middleware systems," *Softw. Pract. Exper.*, vol. 40, no. 6, pp. 485–506, 2010.

[19] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction To Algorithms (2nd Edition)*. McGraw-Hill, 2001.

[20] W. Alsalih, S. Akl, and H. Hassanein, "Placement of multiple mobile data collectors in underwater acoustic sensor networks," in *Proc. IEEE International Conference on Communications (ICC)*, May 2008.

[21] (2014, Sept.) Skyemodule M10. [Online]. Available: http://www.skyetek.com/products/rfid/skyemodule-m10