RFIDcover - A Coverage Planning Tool for RFID Networks with Mobile Readers

S. Anusha and Sridhar Iyer

Indian Institute of Technology Bombay, India
{anusha, sri}@it.iitb.ac.in

Abstract. Radio Frequency Identification (RFID) finds use in numerous applications involving item identification and tracking. In a typical application, RFID tags are attached to the items and are periodically queried by readers. Using a fixed placement of readers to guarantee complete coverage of all tags in a given area at all times increases the deployment costs. Also, most practical applications do not need complete coverage at all times. It is enough to provide complete coverage periodically, say each tag being covered every τ seconds. For such applications, using mobile readers to cover the area would be more cost-effective.
Given an area to be covered completely within a period τ, determining the number of mobile readers required, their placement and movement pattern, is a difficult problem. We have developed *RFIDcover* [1], an automated coverage planning tool, that addresses this problem. Given an application scenario and reader specifications, *RFIDcover* determines an optimal number of readers required to guarantee complete coverage within the specified period τ. It also generates a layout giving the placement and movement pattern of the readers. The architecture of *RFIDcover* is generic and extendible, making it easy to implement different application scenarios. In this paper, we present *RFIDcover* implementation for a retail inventory tracking application scenario and evaluate its effectiveness.

1 Introduction

Radio Frequency Identification technology finds a plethora of applications in various commercial sectors for tracking and identification of objects. The key components of an RFID system are the tags and readers. The RFID tag is a low functionality microchip with an antenna connected to it, that is attached to the item to be tracked, or identified, and stores the unique identification number of the item. The readers communicate with the tags for reading/writing the information stored on them. Readers may be fixed (stationary at a location) or mobile (within the given area).

The tags used for most applications are *passive*[1] [2], which puts a limit on the readers' *interrogation range* - the range within which a tag can be read by the reader. For example, RFID readers operating in the UHF band typically have an interrogation range of 3-5 m[2]. As a result, a large number of readers may be required to provide complete coverage for a given area, leading to significant deployment costs.

[1] This work was partly supported by Persistent Systems Pvt. Ltd. www.persistent.co.in

Some applications, such as retail inventory tracking, need complete coverage only periodically, say each tag being covered every τ seconds. For such applications, using mobile readers to cover the area would be more cost-effective. However, before deploying the readers, it is necessary to answer many important questions, such as: (i) how many readers are needed for providing complete coverage, (ii) where should the readers be placed, (iii) how should the mobile readers move and with what velocity, (iv) how does the number of readers required vary with increase, or decrease, in τ. Thus, given an area to be covered completely within a period τ, determining an optimal number of mobile readers required, their placement and movement pattern, is a difficult problem.

We have developed *RFIDcover*, an automated coverage planning tool that addresses this problem. Given an application scenario and reader specifications, *RFIDcover* determines an optimal number of readers required to guarantee complete coverage within the specified period τ. It does this by: (i) automatically generating a set of layouts (the placement and movement pattern of the readers), (ii) computing the performance metrics - *Cost of deployment* (as a function of the number of readers) and *Tag Reading Time (TRT)* (the time taken to read all tags in the given area) - for each layout, and (iii) selecting the layout which is optimal in terms of both.

The architecture of *RFIDcover* (Section 3) is generic and extendible, making it easy to implement different application scenarios. We consider the following retail inventory tracking application: An RFID tag is attached to each item in a supermarket. The items are then stacked up on the shelves separated by aisles. Periodic inventory checks are carried out using mobile readers moving along the aisles, reading the tags on the shelves as they move. We have implemented *RFIDcover* for such an application (Section 4), and evaluated its effectiveness (Section 5). To the best of our knowledge, there is no literature on coverage tools for RFID systems with mobile readers.

2 The Coverage Problem

In this section, we discuss the problem of providing complete coverage using fixed as well as mobile readers. We assume an interference free environment and a circular range for each reader. We derive theoretical results for the number and placement of readers to completely cover a given area. These results form the basis for the mobility models and heuristics used in *RFIDcover*.

2.1 Fixed Readers

Given the minimum-area rectangle (with dimensions $X \times Y$) that encloses the area to be covered, the problem of complete coverage using fixed readers is same as that of covering this rectangle with a number of fixed size circles, each of radius equal to the readers' interrogation range, r.

With no overlaps among the readers, the maximum coverage that can be achieved is 0.906899682[3]. Hence, complete coverage is possible only if there is overlap amongst the readers. Kershner, in [4], states that the density of an optimal layout, i.e., the ratio of sum total of the area covered by all the circles to the total area to be covered, would be 1.209. [3] discusses a layout that meets this criteria, and hence, is optimal. The number of readers required for complete coverage using the layout is:

$$F_{opt} = \frac{2\sqrt{3}XY}{9r^2} \tag{1}$$

It can be easily seen that using only fixed readers may not be cost-effective as X and Y increase, especially if $r \ll X$, and/or, Y. For example, to cover an area of dimension $10m \times 10m$ with readers of interrogation range, $r = 2m$, the number of readers required would be 10. Whereas, to cover an area of dimension $50m \times 50m$, this would shoot up to 241. Hence, we explore coverage using mobile readers.

2.2 Mobile Readers

The approach for determining the optimal number of mobile readers to cover a given area is similar to that used for fixed readers, except for the following difference: The area covered by a mobile reader would not be of the shape of a circle of radius r, and would instead be an ellipse-like shape as discussed in [3]. As in the case of fixed readers, complete coverage can be guaranteed only if overlaps are allowed. The number of mobile readers required for one such layout with overlaps discussed in [3] that provides complete coverage is:

$$M = \lceil \frac{X \times Y}{2rv\tau} \rceil \tag{2}$$

Although M may not be the *minimal* number of mobile readers required, it gives a *sufficient bound* for the number of mobile readers. We note that a deployment using M readers would need a *to-and-fro* mobility model for each reader. Such a model may be restrictive and impractical for many scenarios. Hence, we use this value of M only as a comparison point for the evaluation of mobility models implemented in *RFIDcover*. In the next section, we describe the architecture of *RFIDcover* in detail.

3 *RFIDcover* Architecture

The architecture of *RFIDcover* is as shown in Figure 1. It has a three phase operation as follows:

1. *Selection Phase* - In this phase, the mobility model for the mobile readers and the MAC mechanism to be used by the readers for shared access to the medium, are chosen based on the application scenario. Depending on these two, an appropriate heuristic for layout generation is selected.
2. *Generation Phase* - In this phase, the selected heuristic is used to generate a set of possible layouts, each of which conforms to the input constraints and also completely covers the given area. The performance metrics, viz., the *Cost* (as a function of the number of readers), and the *TRT* (total time taken to read all the tags in the entire area) are computed for each such layout.
3. *Optimization Phase.* In this phase, an appropriate objective function for optimization is chosen and applied to the set of layouts generated earlier. This results in the selection of an optimal layout, which is recommended to the user.

An additional feature of *RFIDcover* is that the user can also provide constraints on the *Cost* and *TRT* as an input to the generation and optimization phases.

The following subsections briefly describe the different components shown in the architecture and their roles, using the supermarket inventory application mentioned earlier (Section 1).

3.1 Inputs

The user provides as input to *RFIDcover* the following:

- *Reader Specification* - This gives the details (as shown in Figure 1) of both fixed and mobile readers.
- *Topology Specification* - The dimensions of the minimum-area rectangle enclosing the region to be covered and the tag density distribution is provided.
- *Application Scenario* - The application scenario is chosen from a list of those currently supported by *RFIDcover*.
- *Input Constraints* - Additional constraints may be provided by the user which can be modified on-the-fly to generate a new set of layouts.

3.2 Functional Overview

Brief description of the functions of each of the basic components in *RFIDcover* is given below. The various components have been loosely grouped together to correspond to the three phases of its operation.

Selection Phase The two main components in this phase are:

- *Model Selector* : The model selector maps the given application scenario and tag distribution to a suitable mobility model and an appropriate MAC mechanism (medium access control). It then passes on the selected mobility model and MAC mechanism as an output to the next phase. The model selector uses the following components for the mapping:
  - *Mobility Model Set* : This is a collection of mobility models. A mobility model defines how the mobile readers would move and with what velocity. We assume a homogeneous system, where all the mobile readers follow the same mobility model. A given mobility model may be suitable for some application scenarios and may not be useful for others. Also, note that a mobile reader can not move faster than $TRS/(2r \times Tag\ density)$ in order to cover all tags in the range.
  - *MAC Set* : The readers may use a number of Medium Access Control (MAC) mechanisms, such as TDMA, FDMA, CSMA, in order to share the medium and read the tags efficiently. MAC set is the collection of such mechanisms currently supported by *RFIDcover*. Any new MAC mechanism can be implemented and added to this set.
- *Layout Generating Heuristic Selector* : Once the mobility model and MAC mechanism has been fixed, and given the tag distribution, reader specifications and the dimensions of the area to be covered, the layout generating heuristic selector finds an appropriate heuristic to be used for generating the layouts. These layouts provide complete coverage of the area and conform to the input constraints. The layout generating heuristic selector uses the following component for its section decision:

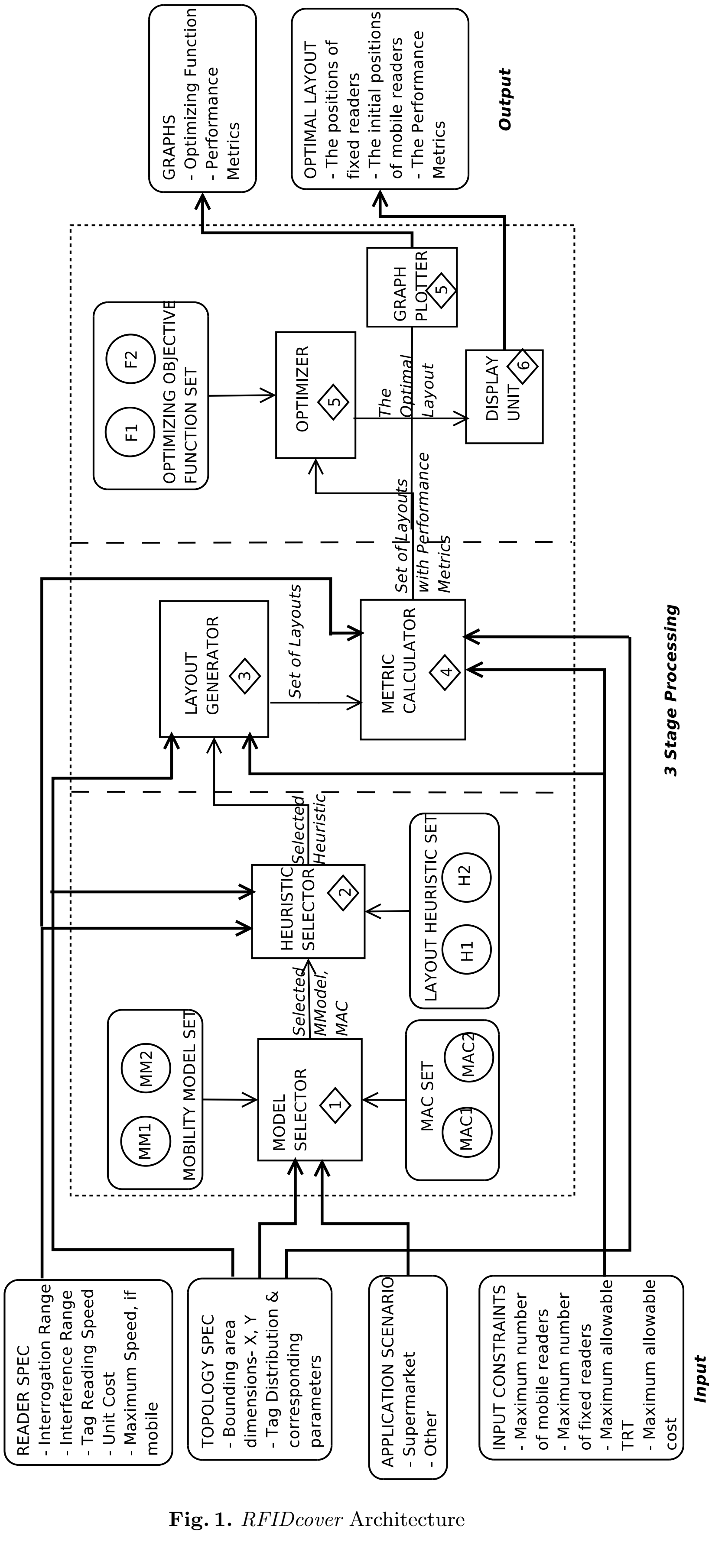


Fig. 1. *RFIDcover* Architecture

- *Layout Generating Heuristic Set* : This is a collection of heuristics that can be used for generating the layouts. Each heuristic has associated with it some mobility models that it supports. Hence, a heuristic along with the mobility model would result in a layout for complete coverage of the area. For example, a layout generating heuristic could place fixed readers at four corners of the area to be covered and place one mobile reader, initially at left top corner, and let it move in a pattern that ensures coverage of the area.

Generation Phase At the end of the *selection phase* we have all the information needed for the *generation phase*, for generating the layouts for complete coverage. This phase consists of two steps and uses the following components:

- *Layout Generator* : It takes as input the topology specifications, the reader specifications, the input constraints, and the chosen layout generating heuristic, and applies that heuristic to generate a set of possible layouts that would provide complete coverage in conformance with the input constraints.
- *Metric Calculator* : For each layout generated by the layout generator, the performance metrics are computed, and keeping in mind the input constraints, a subset of the layouts that conform to the input constraints is generated. The performance metrics include simple generic ones like the number of fixed and mobile readers, the total *Cost* incurred and the *TRT* (or Tag Reading Time, which is the total time it takes to read all tags in the area). In addition, other metrics specific to MAC mechanism or application scenario could also be computed.

Optimization Phase Once the layouts conforming to the input constraints have been generated, the *optimization phase* then determines best layout from the set of those generated. The basic building blocks of this phase are:

- *Optimizer* : Since the layout is generated using a heuristic, there can be a number of layouts that would conform to the input constraints. Various optimizing objective functions are possible. The optimizer applies a suitable optimizing objective function to the layouts generated and recommends the result to the user, along with a summary of the other conforming layouts. The optimizer uses the following component while choosing the objective function.
  - *Optimizing Objective Function Set* : This is a collection of optimizing objective functions. For each layout generated that conforms to to the input constraints, we have a set of performance metrics. An optimizing objective function can be applied on some/all of these metrics for determining the most suitable layout. For our supermarket example, we might be interested in recommending a layout that uses the minimum number of readers. Then the objective function should be the "minimum" function and it should be applied on the number of readers.
- *Graph Plotter* : This is used to plot graphs using the layouts conforming to the constraints. The plots reflect the variation of number of fixed readers, mobile readers, *Cost*, *TRT* and other metrics with different layouts. It also displays the optimizing objective function.
- *Display Unit* : This is used to graphically display in detail the optimal layout and the performance metrics associated with it.

3.3 Outputs

Following are the outputs that *RFIDcover* provides to the user.

- *Graphs* : The various graphs generated by the graph plotter are shown to the user for analysis.
- *Optimal Layout* : The details of the layout recommended to the user, are shown to the user on a graphical interface.

In the next section, we present the implementation of *RFIDcover* for the retail inventory tracking application.

4 *RFIDcover* Implementation

The architecture presented in Section 3 has been implemented in Java. *RFIDcover* currently supports the supermarket inventory application scenario. It implements the zig-zag mobility model and the static coloring MAC that are suitable for the supermarket application. LGH_1, a heuristic specific to the zig-zag mobility model is used to generate the set of layouts. The *least square sum* optimizing function is then applied on the parameters *TRT* and *Cost*, to get the optimal layout that is suggested to the user.

4.1 Zig-zag Mobility Model

In the zig-zag mobility model, the mobile readers move within a rectangular area in a zig-zag fashion. This rectangular area (as shown in Figure 2) forms the basic unit which is replicated throughout the area to be covered. So, the dimensions of this rectangle could be as large as $X \times Y$, or as small as $length\ of\ aisle \times inter\ aisle\ distance$.

This mobility model is suitable for supermarket scenario where tagged items are stacked up on shelves in rows with aisles separating them. The mobile readers move from left to right (or right to left) along the aisle and then move an inter-aisle distance perpendicular to it, and then move from right to left (or left to right) along the aisle and repeat the whole process again. The model is as shown in Figure 2.

4.2 Static Coloring MAC

Static coloring MAC is a TDMA based MAC mechanism for shared access of the readers to the medium of communication. We chose a TDMA based mechanism because this is simple and is commonly used by commercially available RFID readers.

In TDMA based mechanism, each reader is assigned a time slot wherein it can read tags in its range, while the rest of the readers remain silent during the period. Any two readers in the interference range of each other will be assigned different time slots, thus, avoiding any interference or collision. Assigning time slots to readers is analogous to the graph coloring problem [5], where the readers form the vertices of the graph and every pair of interfering readers are adjacent, that is, they have an edge between them. In case of fixed readers, determining these pairs of interfering readers and assigning TDMA slots is straightforward.

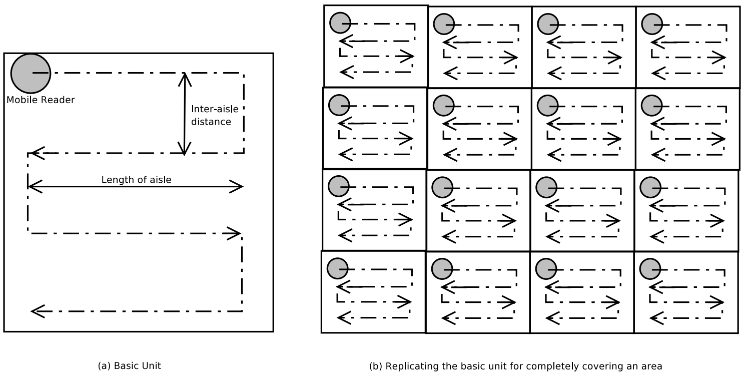

Mobile Reader
Inter-aisle distance
Length of aisle
(a) Basic Unit
(b) Replicating the basic unit for completely covering an area

Fig. 2. The Zig-zag Mobility Model

In case of mobile readers, determining the number of TDMA slots is not easy, since the readers may move in and out of the interference range of each other. One simple way to overcome this difficulty is to determine the number of interfering readers in worst case scenario and assign as many slots. The static coloring MAC considers the mobility model, determines all pairs of readers that might come into the interference range of each other, and assigns time slots on a worst case basis. This approach works for many simple mobility models and deterministic layouts, although in general it may be a bit inefficient.

4.3 LGH_1 Layout Generating Heuristic

RFIDcover currently implements a heuristic, which we call LGH_1, for generating the layouts appropriate for the zig-zag mobility model. It generates a hybrid layout consisting of fixed as well as mobile readers. The layouts differ in the strategic points where the fixed readers are placed. LGH_1 works as follows: It places fixed readers at the end of every aisle, or at the end of every two aisles and so on, in each direction. The fixed readers, thus, form a grid-like structure over the area to be covered. Each cross section of the grid forms the rectangular region within which one, or more, mobile readers move as per the zig-zag mobility model. More details are given in [3].

4.4 Optimization

The layout generating heuristic produces a set of layouts. For each layout, the number of fixed and mobile readers, the performance metrics like the *Cost* and *TRT*, are calculated and those conforming to the input constraints are retained. One of these is chosen and recommended to the user, as per an appropriate optimizing objective function.

In our supermarket application, we would like to use minimum number of readers and yet provide complete coverage as often as possible. Hence, we use the *least square sum* as the objective function and apply it on the parameters *TRT* and *Cost*. Thus, the layout with minimum value for $TRT^2 + Cost^2$ is chosen as the optimal one.

In the next section, we provide an evaluation of the implementation for the supermarket scenario.

5 *RFIDcover* Evaluation for Supermarket Scenario

In this section, we discuss the correctness and usability of *RFIDcover*'s current implementation for the supermarket inventory application scenario. We consider the example shown in Table 1. We ran *RFIDcover* on the example inputs, to get the data used to plot the graphs in Figure 3. Figure 3(a) shows the graphs of the number of mobile readers (NMR) and the optimal number of fixed readers (NFR) with varying *TRT*. From the graphs in the figure, we can see that for providing complete coverage of an area, the number of mobile readers needed is much lesser than the number of fixed readers. The number of mobile readers required decreases drastically as *TRT* increases and would reach 1 when *TRT* approaches ∞. This is as expected.

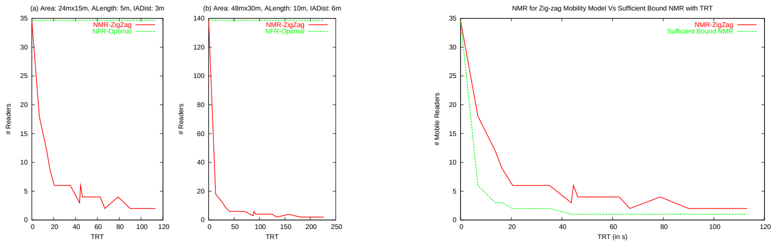
Table 1. The example

Reader Spec	Topology Spec	Application
Interrogation Range: $2m$	Dimension X: $24m$	Supermarket Scenario
Interrogation Range: $2.5m$	Dimension Y: $15m$	Aisle Length: $5m$
Tag Reading Speed: $70tags/s$	Tag Distribution: Uniform	Inter Aisle Distance: $3m$
Unit Cost: 1	Tag Density: $5/m^2$	MAC mechanism
Max. Speed: $5m/s$	Aisle length along X	Static Coloring

For practical applications, very high values of *TRT* may not be meaningful. Hence, the region of interest is the area of the graph near the origin, where the *TRT* values are within the range of a few seconds. Here also, we find that except when the *TRT* is negligibly small, the number of mobile readers required drops significantly with slight increase in *TRT*. Hence, using mobile readers for such values of *TRT* would be cost-effective.

Another important observation is that in the region of interest the slope of the number of mobile readers curve falls very steeply if $r \ll X$, or Y but at a much lower rate if r is comparable to X and Y. For example, the curve in the left graph of Figure 3(a) (which corresponds to an area of dimensions $24m \times 15m$ and *aisle length* $= 5m$ and *inter aisle distance* $= 3m$), falls from 35 readers to around 11 readers as the *TRT* increases from $0s$ to $15s$, whereas in the right graph of Figure 3(a) (which corresponds to an area of dimensions $48m \times 30m$ and *aisle length* $= 10m$ and *inter aisle distance* $= 6m$), it falls from 139 readers to 17 readers, even though the number of aisles is the same in both cases.

The graph in Figure 3(b) compares how the number of mobile readers (NMR) needed for complete coverage using the zig-zag mobility model varies with respect to the *sufficient bound* on the number of mobile readers. As can be observed, the zig-zag mobility model curve follows the *sufficient bound* quite closely. This



(a) NMR for Zig-zag Mobility Model Vs NFR

(b) NMR for Zig-zag Mobility Model Vs Sufficient Bound NMR

Fig. 3. Observations

asserts the practical suitability of the zig-zag mobility model for the supermarket application scenario.

6 Conclusions

We have presented *RFIDcover*, an automated coverage planning tool that determines an optimal layout of readers required to guarantee complete coverage for a given application scenario. We have also evaluated the effectiveness of *RFIDcover* for the retail inventory tracking application. *RFIDcover* provides the user with crucial deployment-specific information such as the number of readers needed, their placement, movement pattern, and the deployment cost. It also gives the user the flexibility to input additional constraints on-the-fly, thereby making it a very useful tool for RFID deployment.

The architecture of *RFIDcover* is generic and extendible, enabling easy implementation of other application scenarios. Even for existing application scenarios, other mobility models, MAC mechanisms and layout generating heuristics can be implemented. This would enable a comparison of various deployment options for the same application.

References

1. Radio frequency identification - a basic primer. White Paper, AIM Inc WP-98/002R2 (2001) http://www.aimglobal.org
2. Finkenzeller, K.: RFID Handbook : fundamentals and applications in contactless smart cards and identification. dritte edn. Chichester : John Wiley, Leipzig (2003)
3. Anusha, S.: *RFIDcover* - a coverage planning tool for rfid networks with mobile readers. Master's thesis, Indian Institute of Technology Bombay (2005)
4. Kershner, R.: The number of circles covering a set American Journal of Mathematics, Vol. 61, 665-671 (1939)
5. Engels, D.W.: The reader collision problem. Technical report, epcglobal.org (2002)