*This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2007 proceedings.*

# Load Balancing in Large-Scale RFID Systems

Qunfeng Dong*, Ashutosh Shukla*, Vivek Shrivastava*, Dheeraj Agrawal*, Suman Banerjee* and Koushik Kar†

*University of Wisconsin-Madsion
Madison, WI 53706, USA
Email: {qunfeng, viveks, shukla, dheeraj, suman}@cs.wisc.edu

†Rensselaer Polytechnic Institute
Troy, NY 12180, USA
Email: koushik@ecse.rpi.edu

***Abstract*—A radio frequency identifier (RFID) system consists of inexpensive, uniquely-identifiable tags that are mounted on physical objects, and readers that track these tags (and hence these physical objects) through RF communication. In this paper we, therefore, address this load balancing problem for readers — given a set of tags that are within range of each reader, which of these tags should each reader be responsible for such that the cost for monitoring tags across the different readers is balanced, while guaranteeing that each tag is monitored by at least one reader. We show that a generalized variant of the load balancing problem is NP-hard and hence present a 2-approximation centralized algorithm. We next present an optimal centralized solution for a specialized variant. Subsequently, we present a localized distributed algorithm that is probabilistic in nature and closely matches the performance of the centralized algorithms. Our results demonstrate that our schemes achieve very good performance even in highly dynamic large-scale RFID systems.**

## I. INTRODUCTION

Radio frequency identifier (RFID) as a short-range radio technology for automated data collection is becoming an integral part of our life. Since its first emergence back in 1960s [8], advances in VLSI technology have enabled massive manufacture of RFID devices at extremely low costs. Nowadays, RFID has found hundreds of applications such as inventory management, supply chain automation, electronic toll collection, anti-theft of automobiles and merchandise, access control and security, etc.

Usually, RFID systems are composed of two types of devices: simple, inexpensive, and uniquely-identifiable *tags* and more powerful *readers*. Both tags and readers have an antenna for radio communication with each other. Readers communicate with the tags to detect them in their physical vicinity. Each tag has a small amount of memory which stores its unique identifier as well as some useful data. In typical RFID applications, tags are attached (embedded) onto (into) targets of interest so that the host targets can be effectively monitored by the system using tag readers. The architecture of such an RFID system is illustrated in Figure 1, where a central repository can gather data from readers through multi-hop wireless communication.

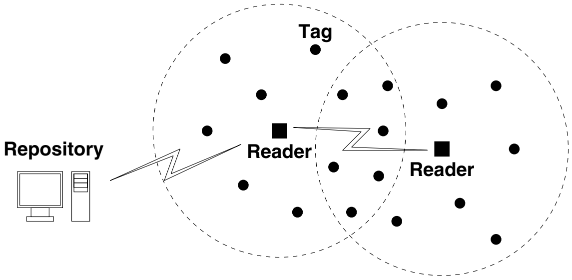


Fig. 1. An example RFID system. Square nodes represent readers and round nodes represent tags.

In increasingly deployed large-scale RFID systems, each RFID reader is responsible for retrieving data from a large number of RFID tags within its vicinity. After a reader sends out a tag poll message, if multiple tags respond simultaneously, radio interference at the reader will typically result in a failed transmission. In order to solve this problem many anti collision schemes like *binary tree-walking protocol* [10] and *Q protocol* [1] have been proposed. Even under such optimizations, the cost at each reader is proportional to the number of tags it is responsible to read. For various performance measures, it is important to design effective load balancing schemes for distributing tags among readers as evenly as possible.

For example, consider the case where the readers are battery-powered. In this case, more the number of tags assigned to each reader, the greater is its rate of energy depletion. In particular, as the distribution of tags to readers gets more skewed, some heavily loaded readers will exhaust all of its battery-power fairly quickly, leading to loss of coverage. Similarly, if each tag in the system is monitored periodically, then a reader with a higher load of tags will be able to monitor its tags less frequently. This will lower the average monitoring frequency of the system.

In this paper, we consider the problem of assigning tags to readers in order to minimize the maximum total cost required at any reader to retrieve data from its assigned tags. For different performance measures, the cost metric can model different physical quantities. For example, if energy efficiency is the performance measure for a battery-powered RFID system, then the cost models the energy expended by each reader to monitor all of its tags. Equivalently, this will maximize the lifetime of the system until the first failure of some reader due to battery depletion. For simplicity, we

Q. Dong, V. Shrivastava, D. Agrawala, A. Shukla, and S. Banerjee were supported in part by the following NSF grants: CNS-0520152, CNS-0627102, CNS-0639434, and CNS-0627589.

K. Kar was supported in part by the following NSF grants: CNS-0448316, CNS-0435141 and ECS-0330203.

refer to this problem as the *min-max cost assignment (MCA)* problem.

In many cases, the readers may use a fixed transmission power for their interactions. In such cases, the objective of the MCA problem is simply to minimize the maximum number of tags assigned to any reader. Clearly, this problem is a special case of the MCA problem, where the energy cost of sending a message to any tag (in vicinity) is always fixed to be the same. For simplicity, we refer to this problem as the *min-max tag count assignment (MTA)* problem.

In either case, a load balancing scheme cannot be considered scalable (hence practical in large-scale systems), if it involves high complexity and overheads and is centralized in nature. This is because, in typical deployments, e.g., in a warehouse, the number of monitored tags can be in millions. Therefore, designing efficient distributed load balancing schemes becomes a critical issue in the implementation of large-scale RFID systems.

In this paper, we address these load balancing problems in the context of very inexpensive (few cents) *passive* tags, i.e., tags that have no power source of its own and have very limited capabilities. Passive tags support a very small set of operations including: (i) a reader can store some value in the tag, (ii) it can query the tag for stored values, and (iii) it can ask the tag to respond in a probabilistic manner (based on a probability that the reader announces).

In this scenario, we make the following key contributions to the problem of load balancing in large-scale passive-tag based RFID systems:

- We show that even with centralized knowledge about the system, the general MCA problem is NP-hard and cannot be approximated within a factor less than $\frac{3}{2}$. An efficient 2-approximation algorithm is then presented. We also present a conceptually very simple algorithm for optimally solving MTA in polynomial time using centralized knowledge.
- We propose a simple and effective localized scheme for these problems that can be practically implemented in passive RFID tag systems. Our localized scheme is probabilistic and tag driven. Our results demonstrate that this low cost scheme can achieve very good performance even in highly dynamic large-scale passive RFID systems.

## II. RELATED WORK

In the literature, Carbunar *et al.* [3] have studied the *redundant reader elimination* problem caused by *reader collision*, where tags covered by multiple readers suffer from interference caused by simultaneous transmissions by these readers. Their objective is to turn off as many readers as possible (without sacrificing tag coverage), so that reader collision is minimized and energy consumption is reduced as well. Our tag assignment problems can be viewed as orthogonal to the redundant reader elimination problem: after redundant readers are powered off, our schemes can be applied to assign tags to active readers in a load balanced manner.

In [7], Kodialam and Nandagopal consider the problem of efficient estimation of the number of RFID tags in the system upto a desired level of accuracy. The authors present a scheme that estimates the cardinality of the tag-sets of any size in near-constant time. Note that our objective is quite different from the one considered in [7]: whereas the algorithm in [7] can be used to estimate the number of tags in the vicinity of each reader, our algorithm assigns tags so as to distribute the load evenly amongst readers, once the tags in the neighborhood of each reader have been identified.

Another related work comes from the well researched maximum lifetime broadcast problem [6], where the objective is also to minimize the maximum energy cost at any node. The key difference between their problem and ours lies in the definition of nodal energy cost. In our problem, the energy cost of a reader is the aggregate energy cost of reading individual tags. In their problem, because nodes are broadcasting instead of collecting information, one single broadcast transmission suffices to distribute the information to all neighbors in transmission range. Therefore, their definition of the energy cost of a node is the minimum energy cost required to reach all of its children in the broadcast tree. This definition leads to an optimization problem that is quite different from ours.

In the context of WLAN, Bejerano *et al.* have recently studied a closely related load balancing problem [2] where the objective is to assign WLAN clients to access points (APs) in a load balanced manner. Their objective is also to find an assignment of clients to APs, where the edge between an AP and a client has a cost that is inversely proportional to its effective bit rate. However, the performance measure of an assignment is to obtain max-min fairness among APs. Although their problem is seemingly more general, it is actually not the case for the general MCA problem and their approximation algorithm does not automatically yield the same result for our MCA problem. In the special case where edge costs are fixed to be the same, they gave an optimal solution to the max-min fairness problem, which can be directly used to solve our MTA problem. Nonetheless, our solution to the MTA problem is conceptually much simpler than their solution, as their solution is targeted on an essentially different problem.

## III. FORMULATION

For the purpose of assigning tags to readers, we only need to consider links between tags and readers. Thus, the RFID system can be modeled as a bipartite graph $G = (U \cup V, E)$, where $U = \{u_1, u_2, \cdots, u_m\}$ denotes the set of $m$ readers and $V = \{v_1, v_2, \cdots, v_n\}$ denotes the set of $n$ tags. Moreover, communication between tags and readers are bi-directional, and thus the bipartite graph is an undirected graph. There is an (undirected) edge $(u_i, v_j)$ between reader $u_i$ and tag $v_j$ if only if they can communicate with each other. Each edge $(u_i, v_j)$ has a non-negative energy cost $c_{ij}$ representing the energy cost of reader $u_i$ to read tag $v_j$ once. In principle, $c_{ij}$ can also represent other meaningful metrics. For each reader $u_i$, let $N(u_i)$ denote the set of tags it can read. Similarly, let $N(v_j)$ denote the set of readers that can read tag $v_j$. Our model is general enough to allow any communication range pattern, like the irregular patterns where the effective transmission range of any node may not be the same in all directions.

*Problem definitions:* In this paper, we study the min-max optimization problem where our goal is to find an *assignment* $\varphi : V \to U$ of each tag $v_j$ to some reader $u_i = \varphi(v_j)$ such that the maximum total energy cost

$$C_i = \sum_{\substack{1 \le j \le n \\ u_i = \varphi(v_j)}} c_{ij}$$

over all readers is minimized. We refer to this problem as the *min-max cost assignment (MCA)* problem. Note that although we use energy cost as an example, in general $c_{ij}$ can represent any meaningful performance metric (e.g. the amount of time that it takes reader $u_i$ to retrieve data from tag $v_j$). The decision version of MCA is formally defined as follows.

> **INSTANCE** Bipartite graph $G = (U \cup V, E)$, a cost $c_{ij} \in Z^+$ for each edge $(u_i, v_j)$ and a bound $B \in Z^+$.
>
> **QUESTION** Is there an assignment $\varphi : V \to U$ such that for each $u_i \in U$,
>
> $$\sum_{\substack{1 \le j \le n \\ u_i = \varphi(v_j)}} c_{ij} \le B?$$

The *min-max tag count assignment (MTA)* problem is a special case of the MCA problem, where readers cannot adjust their transmission power and thus each edge has a fixed unit energy cost, namely $c_{ij} = 1$.

## IV. CENTRALIZED SCHEMES

In this section, we formally analyze the complexity of the MCA problem and the MTA problem in the centralized setting.

### *A. Min-max Cost Assignment (MCA)*

The argue NP-hardness of the MCA problem, we consider a restricted *unit-disk graph (UDG)* model, where the communication range of all readers and tags are assumed to be the same ($r$). We can show that MCA is NP-hard in the UDG model (and thus in the general graph model as well) through a reduction from the PARTITION problem (see [5] for details). Given the NP-hardness of MCA, our goal is to design an efficient approximation algorithm for the problem. It turns out that in the general graph model (and therefore in the special case of the UDG model as well), we can easily design a 2-approximation algorithm for MCA by reducing to the *minimum multiprocessor scheduling (MMS)*.

In MMS, we are given a set $T = \{t_1, t_2, \cdots, t_n\}$ of *tasks* and a set $P = \{p_1, p_2, \cdots, p_m\}$ of *processors*. Each task $t_j \in T$ has a positive *length* $l_{ij} \in Z^+$, which represents the amount of time needed to execute task $t_j$ (completely) on processor $p_i$. A *schedule* $\phi : T \to P$ is an assignment of each task $t_j \in T$ to some processor $p_i \in P$. The execution time on processor $p_i$ is thus the total execution time of all the tasks assigned to it. The *finish time* of a schedule $\phi$ is the maximum execution time over all processors. Our objective in MMS is to find a schedule $\phi$ such that the finish time is minimized.

Given an instance of MCA, we transform it into an instance of MMS as follows.

(1) For each reader $u_i \in U$, create a processor $p_i \in P$.

(2) For each tag $v_j \in V$, create a task $t_j \in T$.

(3) For each pair of reader $u_i$ and $v_j$, let $l_{ij} = c_{ij}$ if $(u_i, v_j) \in E$ and let $l_{ij} = \infty$ otherwise.

Without loss of generality, let $\mathcal{A}$ denote the best known approximation algorithm for MMS whose approximation ratio is $\alpha$. To derive an $\alpha$-approximation algorithm for MCA, we transform the input MCA instance into an MMS instance as described above, and apply $\mathcal{A}$ on the constructed MMS instance to compute a schedule $\phi$. We then define an assignment $\varphi$ for the given MCA instance such that for each pair of reader $u_i$ and tag $v_j$

$$\varphi(v_j) = u_i \iff \phi(t_j) = p_i.$$

Then the maximum total cost $C$ derived from $\varphi$ satisfies $C \le \alpha \cdot OPT_{mms} = \alpha \cdot OPT_{mca}$ (see [5] for proof). Using this procedure, the 2-approximation algorithm for MMS proposed by Lenstra *et al.* [9] will result in a 2-approximation to the MCA problem as well. The authors in [9] also show that MMS cannot be approximated within a factor less than $\frac{3}{2}$, unless $P = NP$. We can show that even in the restricted UDG model the same inapproximability bound holds for MCA, simply by reducing MMS to MCA (see [5] for details).

### *B. Min-max Tag count Assignment (MTA)*

Our MTA algorithm is essentially an iterative binary search process; in each iteration, we test some specific load $B$ to see if there exists some assignment $\varphi : V \to U$ such that the number of tags assigned to any reader is no more than $B$. If it is the case, we decrease the value of $B$; otherwise, we increase the value of $B$. This iterative process terminates and results in minimizing the maximum load on the readers.

Next, to solve the feasibility test of $B$, or a decision version of the MTA problem, we construct an instance of the MNF problem as follows.

(1) Create a virtual source $s$ and a virtual sink $t$.

(2) For each reader $u_i \in U$ in the given MTA instance, create a reader node $u_i$ in the MNF instance. Connect the source $s$ with each reader node using an edge of capacity $B$.

(3) For each tag $v_j \in V$ in the given MTA instance, create a tag node $v_j$ in the MNF instance as well. Connect the sink $t$ with each tag node using an edge of capacity 1.

(4) For each edge $(u_i, v_j)$ in the given MTA instance, create its counterpart in the MNF instance and assign it a capacity of 1.

We can show that there exists an assignment $\varphi$ satisfying the bound $B$ in the given MTA instance if and only if the maximum flow that can be routed from $s$ to $t$ in the constructed MNF instance is exactly $n$ (see [5] for details). Note that it is not possible to route a flow larger than $n$ from $s$ to $t$ since the sink $t$ is only incident to $n$ incoming edges each having unit capacity.

We can now simply apply a standard maximum flow algorithm [4] on the constructed MNF instance. Since $B$ is upper bounded by $n$, a binary search algorithm to find the optimum $B$ will require $O(\log n)$ runs of the maxflow algorithm.

## V. LOCALIZED SCHEME (LPA)

In practice it is often of much interest to deploy a lightweight distributed scheme that delivers reasonably good performance. In this section, we meet this challenge by designing such a distributed scheme, which can also handle dynamic updates (i.e., join/leave of tags/readers) efficiently.

### A. Basic scheme

We now propose the *localized probabilistic assignment (LPA)* scheme, a very simple localized scheme for finding such a tag-driven probabilistic assignment of tags to readers. In this localized scheme, each tag only knows which readers are in its vicinity and what is the load on those readers. Similarly, each reader only knows which tags are in its vicinity and how much (expected) load is each of these tags putting on itself. In order to achieve a more load balanced assignment, in a tag-driven scheme each tag should decide its probability of reporting to some reader based on the load on the latter. If a reader in vicinity has a relatively high load (compared with other readers in vicinity), the tag should report to it with a relatively low probability.

Based on these intuitions, the LPA scheme is designed as follows. Specifically, each reader $u_i$ computes and announces in its polling message the total cost of its incident edges, denoted by

$$l_i = \sum_{v_j \in N(u_i)} c_{ij}.$$

After collecting this total cost from each reader in its vicinity, each tag $v_j$ computes the probability $p_{ij}$ of reporting to reader $u_i$ by

$$p_{ij} = \frac{\left(\sum_{u_k \in N(v_j)} l_k\right) - l_i}{\sum_{u_k \in N(v_j)} l_k} \times \frac{1}{|N(v_j)| - 1} \tag{1}$$

It can be verified that for each tag $v_j$,

$$\sum_{u_i \in N(v_j)} p_{ij} = 1.$$

Therefore, every tag is guaranteed to be read by some neighboring reader in its vicinity, if we ignore communication error at this point. Suppose $N(v_j) = \{u_{i_1}, u_{i_2}, \cdots, u_{i_d}\}$ is the set of readers in the vicinity of tag $v_j$. We can view all the $p_{i_k j}$'s of tag $v_j$ in the form a vector $(p_{i_1 j}, p_{i_2 j}, \cdots, p_{i_d j})$, which we refer to as the *probabilistic binding vector (PBV)* of tag $v_j$. To facilitate later discussion, we refer to such an interactive process between tags and readers as a *round* of load balancing. We also assume that each tag $v_j$ will record the load $l_i$ of each reader $u_i$ in $N(v_j)$, and refer to the vector $(l_{i_1}, l_{i_2}, \cdots, l_{i_d})$ as the *neighbor load vector (NLV)* of tag $v_j$.

In the basic LPA scheme we have described so far, each tag $v_j$ can be assigned to any reader that can cover $v_j$ with maximum transmission range. A possible improvement is the following greedy assignment approach, where readers increase their transmission power from a minimum value to the maximum transmission power in certain predefined increments. At each transmission power level, readers probe tags in their current transmission range. If a tag is now probed but has never been probed before, it records as its *candidate readers* the readers that have probed itself at this transmission power level. It is clear that the candidate readers of a tag are the readers that can reach that tag at the minimum transmission power level among all the transmission power levels that are tested in the greedy assignment approach. Subsequently, in the LPA scheme, each tag will only consider reporting to its candidate readers instead of all readers that can cover it with maximum transmission range. We evaluate the performance of this greedy assignment approach with different increments in our results.

### B. Self-adaptive mechanism

Our discussion so far has been conducted on the basis of a static topology. However, in many real applications a load balancing scheme should be able to effectively handle frequent topology changes due to a number of different causes. To be practically useful, a localized assignment scheme should be able to handle such topology changes in a self-adaptive manner. Here, we extend our LPA scheme to incorporate such a self-adaptive mechanism.

*Reader join:* When a reader $u_i$ joins the system and has been ready for retrieving data from tags, it broadcasts a message announcing that its current load is $l_i = 0$. Upon receiving this announcement, each tag in its vicinity expands its NLV to include it. Based on the current load of other readers stored in its NLV, the tag computes a new PBV according to Equation (1). During the next round of data retrieval, the tag will probabilistically report to its neighboring readers including the new reader according to its new PBV. The announcement message broadcast by the new reader is the only overhead of handling its join.

*Tag join:* When a new tag joins a system operating in the passive mode, it can wait until the following round of data retrieval, during which it overhears polling messages from all readers in its vicinity. Based on the load value announced in the overheard polling messages, the new tag defines its own NLV and PBV. During the next round of data retrieval, the tag will be able to participate as usual. No additional message is needed to handle the tag join.

*Reader/Tag leave:* After each round of data retrieval, each reader and tag automatically obtains up-to-date knowledge about its vicinity. Their load, NLV and PBV are then updated based on this up-to-date knowledge. If a reader or tag leaves the system, it will be automatically detected at least after the next round of data retrieval. Therefore, no additional processing is needed to handle reader/tag leaves.

## VI. PERFORMANCE EVALUATION

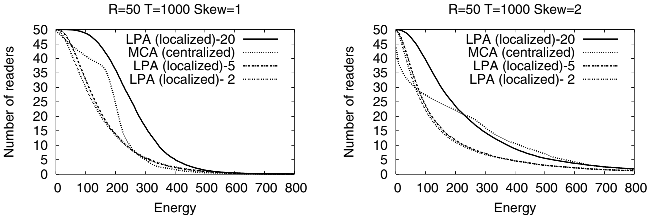
All our experiments are performed by randomly deploying RFID tags and readers in a $1000 \times 1000$ square feet grid. The maximum transmission range of a reader is 12 feets. We analyze the efficacy of our proposed load balancing algorithms by varying the following parameters of the topology: (i) *Tag Density*: Average number of tags in the range of a reader; (ii) *Skew:* The variation in the number of tags in the range of various readers in the system. By varying tag density, we can evaluate our scheme on increasing loads of tags per reader. We generate skewed topologies by choosing a non-linear random function that biases tag placement towards one end of the grid. This bias increases with the increase in skew parameter, pushing more tags in the vicinity of a few readers (readers have uniform random distribution in the grid). Let us consider a one dimensional example to understand the skew parameter in our topologies. Suppose we need to choose a coordinate for a tag in the range [0,1]. So for a skew of $\alpha$, the coordinate of the tag is given by $X^{\alpha}$, where $X$ is a uniform random variable in the interval [0,1]. It is obvious, that higher values of skew push more tag coordinates towards the lower end of the interval [0,1].

Our results, reported next, can be summarized as follows: The proposed localized heuristic (LPA) performs nearly as well as the various optimal and near-optimal centralized algorithms (MTA and MCA) across a wide-range of scenarios varying tag densities and skew. LPA, with its low overheads, and limited need for interactions, is therefore a appropriate choice for efficient load balancing in RFID systems. Due to space constraints, we only present a representative set of interesting results next. An extended version of our evaluation, which considers a larger set of metrics and simulation parameters (including different mobility models), can be found in the companion technical report [5].

We compare the efficacy of our schemes in the aforementioned scenarios in terms of the load vector metric, representative of energy consumption of the reader or tags assigned to the reader. In particular, we consider the following load vectors: (i) *Energy Load Vector (ELV)*: Each element $i$ of ELV represents the number of readers having energy consumption (for communicating with the tags assigned to it) greater than $i$ units in the system; (ii) *Tag Load Vector (TLV)*: Each element $i$ of TLV represents the number of readers assigned more than $i$ tags in the system.

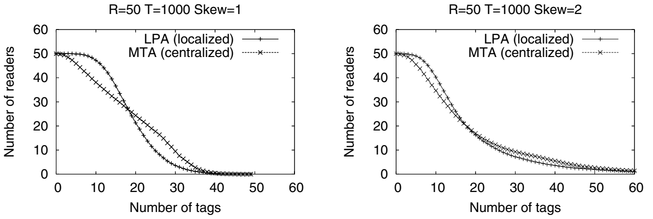
For the sake of clarity, in all the figures presented in this section, the legends are in the same order (from top to bottom) as the curves in the figure.

**LPA vs MCA:** We compare the performance of LPA and MCA for balancing energy consumption of readers in RFID system in Figure 2, which shows the ELV plots for different skew parameters. For the LPA algorithm we use increments (in the transmission power level) of 2, 5, and 20. LPA uses a greedy approach in acquiring tags, and it does well in balancing load across readers. Also, the load balancing across readers improve when the increment is large. In comparison, the MCA algorithm successfully minimizes the maximum value, something which the localized algorithm does not match. Finally, as the skew in the system increases, the achievable load balance becomes poorer for all algorithms.



(a) Skew = 1 (b) Skew = 2.0

Fig. 2. Energy load vectors of LPA and MCA with variation in skew. R and T refer to number of readers and tags respectively. With increasing skew, maximum bound of energy consumption increases, however ELV for LPA remains close to that of MCA.



(a) Skew = 1 (b) Skew = 2

Fig. 3. Tag load vectors of LPA and MTA with variation in skew. R and T refer to number of readers and tags respectively.

**LPA vs MTA:** We compare the performance of LPA and MTA algorithms, with different skew values in Figure 3. As before, the load vectors for the two algorithms indicate that LPA achieves a better load balance than MTA. while MTA achieves its goal of minimizing the maximum tag count. The trend with increase in skew is similar as in the variable cost version of the problem.

## REFERENCES

[1] EPCglobal. Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1.0.9, 2005.
[2] Y. Bejerano, S.-J. Han, and L. E. Li. Fairness and load balancing in wireless lans using association control. In *ACM MobiCom*, 2004.
[3] B. Carbunar, M. K. Ramanathan, M. Koyuturk, C. Hoffmann, and A. Grama. Redundant-reader elimination in rfid systems. In *IEEE SECON*, 2005.
[4] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (Second Edition)*. MIT Press, 2001.
[5] Q. Dong, A. Shukla, V. Shrivastava, D. Agrawal, S. Banerjee, and K. Kar. Load balancing in large-scale rfid systems. In *UW-CS Technical Report 1568 (www.cs.wisc.edu/techreports/2006/TR1568.pdf)*, July 2006.
[6] I. Kang and R. Poovendran. Maximizing static network lifetime of wireless broadcast adhoc networks. In *IEEE ICC*, 2003.
[7] M. Kodialam and T. Nandagopal. Fast and reliable estimation schemes in rfid systems. In *ACM Mobicom*, 2006.
[8] J. Landt. The history of RFID. *IEEE potentials*, 24(4):8–11, 2005.
[9] J. K. Lenstra, D. B. Shmoys, and E. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46(3):259–271, 1990.
[10] S. E. Sarma, S. A. Weis, and D. W. Engels. RFID systems and security and privacy implications. In *CHES*, 2002.