Load Balancing in Large-Scale RFID Systems

Qunfeng Dong, Ashutosh Shukla, Vivek Shrivastava, Dheeraj Agrawal, Suman Banerjee
University of Wisconsin-Madsion
{*qunfeng, viveks, shukla, dheeraj, suman*}*@cs.wisc.edu*

Koushik Kar
Rensselaer Polytechnic Institute
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. For many performance measures in large-scale RFID systems, the set of tags to be monitored needs to be properly balanced among all readers. 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 first present centralized solutions to different variants of this load balancing problem. We show that a generalized variant of the load balancing problem is NP-hard and hence present a 2-approximation 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. Finally we present detailed simulation results that illustrate the performance of the localized distributed approach, how it compares with the centralized optimal and near-optimal solutions, and how it adapts the solution with changes in tag distribution and changes in the reader topology. Our results demonstrate that our schemes achieve very good performance even in highly dynamic large-scale RFID systems.

1. 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 [15], 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. For example, the unique identifier of a tag can serve in place of the UPC bar code of an item in Walmart stores, and the tag is attached to that item for monitoring purpose. By reading the tag periodically using tag readers, the system is thus able to effectively monitor and manage all the tagged items. 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. In some RFID applications, tags may even be equipped with necessary modules to collect dynamically changing data about the object or environment into (onto) which they are embedded (attached).

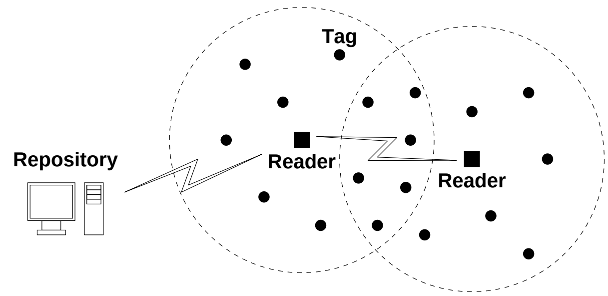


Figure 1: An example RFID system. Square nodes represent readers and round nodes represent tags.

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* [17] and *Q protocol* [1] have been proposed. Even under such optimizations, the cost at each reader is proportional to the number of 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 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.

Problem uniqueness and key contributions

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. Due to their low costs, it is practical to attach these tags to almost any object and are gaining great popularity in supply chain and inventory management applications. Therefore, it is not difficult to envision hundreds of these tags in very small areas, thereby making the load balancing problem particularly important. Passive tags communicate by using the reader-generated inductively-coupled electromagnetic field. They 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). Existing reader-tag communication protocols, e.g., those defined in the EPC Generation 2 UHF RFID specifications [1] use this set of operations to implement necessary communication functions. In fact, these hard limits of tag capability distinguish problems that arise in the domain of RFID systems with passive tags to sensor networking problems. (We comment on the practicality of our developed algorithms under these constraints in Section 6.)

Under these constraints, 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 for obtaining a solution that typically comes very close to the optimum and is guaranteed to be within 2 times the optimum in the worst case. We show that the MTA problem is polynomially solvable with centralized knowledge, and present a conceptually very simple algorithm for optimally solving MTA in polynomial time.

- In practice, localized[1] algorithms are often preferred because of their low complexity and overhead. We, therefore, also 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. By considering the load on the readers, the tags decide which reader to report to. Topology changes caused by join/leave of tags can be efficiently handled as well. Our results demonstrate that this low cost scheme can achieve very good performance even in highly dynamic large-scale passive RFID systems.

[1]A localized algorithm is a distributed algorithm where each node only needs knowledge about its immediate neighbors.

The rest of the paper is organized as follows. Section 2 gives an overview of the RFID technology and an brief experimental evaluation that motivate our work. System models and problem definitions are presented in Section 3. In Section 4, we present our results for the MCA problem and the MTA problem. Our localized scheme is presented in Section 5. In Section 6, we describe how the proposed schemes can be implemented. In Section 7, we evaluate the performance of our schemes. After reviewing related work in Section 8, we conclude the paper in Section 9.

2. BACKGROUND AND MOTIVATION

RFID systems comprise of readers and tags which communicate with each other using radio waves. Tags can be classified into various types depending upon their capabilities. *Passive tags*(Class-1) do not have any power source of their own but use the energy of the reader, *Semi-Passive tags* have an integral power source so can communicate with the reader over a larger distance and *Active tags* can communicate to each other and have ad-hoc networking capabilities. In this paper we will be dealing with inexpensive (few cents) Class 1 passive tags compliant to EPC Generation 2 UHF RFID specifications [1].

In a supply chain management scenario, where readers are deployed to monitor the objects in the inventory, readers need to periodically detect the presence (or absence) of the corresponding tags. To achieve this goal, each reader first 'singulates' (detects EPC identity) tags in its vicinity, e.g., using the Q protocol [1]. Subsequently it can 'select' to 'read' these tags periodically for their presence based on matching bit sequence in EPC or User Memory. If no communication is possible with a singulated tag for a period of time, the tag is assumed to have departed from the reader's vicinity.

We first present an experimental evaluation of reader performance with increasing tag density that illustrates the need for efficient load balancing algorithms in large-scale RFID systems. In these experiments, we used the Alien ALR-9800 Generation 2 reader and ALL-9440 Squiggle tags [2]. The reader has a maximum read range of about 12 feet when operated at maximum RF power (1 Watt in this case). The reader provides software-controlled digital attenuation that reduces the emitted power but not the return signal. Thus, the read range of the reader can be varied by varying the attenuation. he RF attenuation value ranges from 0 (no attenuation, maximum power) to 160 (maximum attenuation,minimum power), in increments of 10, each representing an additional 1 dB of RF attenuation.

In our experiment, the Squiggle tags were kept at distance of 6 feet from the reader antenna and the attenuation was 0 (max power). Figure 2 shows that the average read rate of tags for a single reader *decreases rapidly as the number of tags increases.* Note that in the read rate shown in the plot is normalized to the case of a single tag in the system.

Since tag read rates fall sharply with increasing volume of

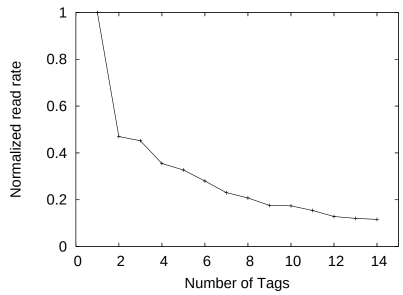


Figure 2: Decay in read-rate with increasing tag density. Experimental evaluation using off-the-shelf commercial readers and passive tags.

tags, it is important that tag reading tasks be distributed to readers in a load balanced way. In addition, for accurate inventory management, it would be preferred that only one reader be responsible for reading each tag. In this paper, we propose load balancing algorithms that would meet these objectives.

3. 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. Note that our model is sufficiently general 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). To facilitate our discussion, we here formally define the decision version of MCA 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 $u_i \in U$,
>
> $$\sum_{\substack{1 \le j \le n \\ u_i = \varphi(v_j)}} c_{ij} \le B?$$

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

4. CENTRALIZED SCHEMES

In this section, we formally analyze the complexity of the MCA problem and the MTA problem in the centralized setting. We prove that even in the restricted *unit-disk graph (UDG)* model, the MCA problem is NP-hard and that there does not exist any efficient approximation algorithm for the MCA problem that can achieve an approximation ratio less than $\frac{3}{2}$. In the UDG model, the communication range of all readers and tags are assumed to be the same, and equal to r. Thus, a reader and a tag can communicate with each other if and only if they are physically separated by a distance no greater than r. Since UDG is a special class of graphs, the NP-hardness and inapproximability results obviously hold for general graphs as well. Note that the UDG model is used only for proving the complexity results; the tag assignment schemes presented later in the paper are applicable to any arbitrary communication model, however. For the MTA problem, we show that it is polynomially solvable even in the general graph model, and present a conceptually very simple algorithm based on network flow for computing the optimal solution.

4.1 Min-max Cost Assignment (MCA)

The NP-hardness of MCA in the UDG model follows from a reduction from the PARTITION problem; the reduction is discussed in [9]. Given the NP-hardness of MCA, our goal is to design an efficient approximation algorithm for the problem. Any lower bound on the achievable approximation ratio, which can give us some idea of the inapproximability of the problem, is of interest to us as well. It turns out even in the general graph model, we can easily design a 2-approximation algorithm for MCA by reducing to the *minimum multiprocessor scheduling (MMS)*. Since the UDG model is a special case of the general graph model, the 2-approximation algorithm automatically applies in the UDG model as well.

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 ϕ is the maximum execution time over all processors. Our objective in MMS is to find a schedule ϕ 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

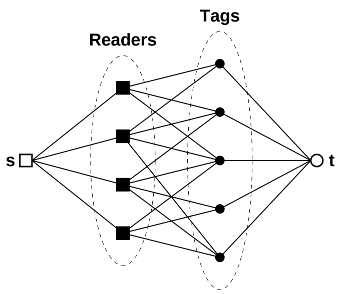


Figure 3: Transformation from MTA to MNF.

is α. Our α-approximation algorithm for MCA is composed of three phases.

(1) Transform the input MCA instance into an MMS instance as described above.

(2) Apply $\mathcal{A}$ on the constructed MMS instance to compute a schedule ϕ.

(3) Define an assignment φ 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 φ satisfies $C \leq \alpha \cdot OPT_{mms} = \alpha \cdot OPT_{mca}$ (proof is omitted here due to space constraints, and can be found in [9]). Using this procedure, the 2-approximation algorithm for MMS proposed by Lenstra *et al.* [16] will result in a 2-approximation to the MCA problem as well. The authors in [16] 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 [9] for details).

4.2 Min-max Tag count Assignment (MTA)

In the previous section, we have proved that the general MCA problem is NP-hard. In this section, we study the MTA problem, which is an interesting special case of MCA where link costs are all the same. Specifically, we show that MTA is polynomially solvable even in the general graph model, and present a conceptually simple algorithm based on network flow for computing the optimal solution.

At the high level, 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 will converge and result 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 (an example of the transformation is shown in Figure 3).

(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 φ 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 [9] 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 a capacity of 1.

We can now simply apply a standard maximum flow algorithm [8] 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.

5. LOCALIZED SCHEME (LPA)

Although the centralized algorithms presented in the previous section possess nice performance properties, in practice it is often of much interest to deploy a light-weight 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. Before we proceed to present the detailed design of our scheme, we first examine some relevant design issues that must be addressed. Our answers to these issues naturally lead to our design.

5.1 Design issues

Randomized vs Deterministic: So far we have been focused on deterministic centralized solutions, where each tag is bound with a fixed reader once it is assigned to it. It is not hard to see that we can do better than this by employing randomized schemes, where each tag may be assigned to multiple readers with some probability. When data is being retrieved from a tag, it flips a coin and decides based on the outcome to which reader it should report. In the long run, the expected load on each reader can potentially be decreased. For a simple example, consider a system consisted of two readers and three tags. Each tag can be assigned to any reader. In the optimal deterministic assignment, one reader must receive two tags while the other reader receives one. If we adopt a randomized approach, we can assign each tag to each reader with equal probability. The long term average load on each reader sums up to 1.5 only, which is more load balanced than the optimal deterministic assignment.

Tag-driven vs Reader-driven: Before we proceed to design a randomized scheme as described above, there is another key design problem that we cannot ignore. Specifically, there are two possible approaches to the design of a randomized scheme: tag-driven and reader-driven. In the tag-driven approach, each tag probabilistically decides to which reader it should report. In the reader-driven approach, each reader probabilistically determines if it should read a tag in its vicinity or not. While these two approaches may seem equally light-weight, we prefer the tag-driven approach. Because in the tag-driven approach we can easily guarantee that every tag will be read by some reader, while in the reader-driven approach some tags may not be read by any reader. Because there is always a positive probability that every reader decides to ignore those tags.

5.2 Basic scheme

In light of these observations, we 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

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

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.

5.3 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. For examples, readers may be turned on/off from time to time according to some power conserving strategy [6], existing tags may leave (e.g. due to merchandise) and new tags may join (e.g. when automobiles carrying tags enter the monitoring zone), etc. To facilitate our discussion, we make the following assumptions about typical RFID systems.

(1) Readers and tags are stationary or semi-mobile. Therefore, topology changes are assumed to be caused by join/leave of readers/tags instead of mobility. Nevertheless, our design does allow readers and tags to be be moved from time to time. Such move can be handled as if readers/tags leave and then join at their new location.

(2) Data retrieval is primarily done in a periodic round-by-round fashion. During each round of data retrieval, every tag should be read by at least some reader. In order to enable effective load balancing and self-adaptive management, readers should announce its presence through polling messages or announcement messages if necessary.

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.

5.4 An iterative optimization

Although this simple one-round localized scheme works well on average, it can be shown that even in the restricted UDG model, its load balancing performance can be arbitrarily bad in the worst case, even for the MTA problem which is just a special case of the general MCA problem. To see that, consider the example in Figure 4, where each node has a transmission range of 1. The system consists of $2n-1$ tags (represented by round nodes) and $n+1$ readers (represented by square nodes). The first row of round nodes represent $n-1$ tags and the third row of round nodes represent the other n tags. The second row of square nodes represent n readers. Edges have the same cost of 1. Each tag in the first row is adjacent to every reader in the second row, and each reader thus has a total cost of $l_i = n$. According to

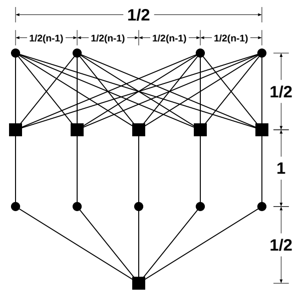


Figure 4: An analysis of the simple localized scheme. Square nodes represent readers and round nodes represent tags. Each node has a transmission range of 1. Each edge has a cost of 1.

the simple localized scheme described above, each tag in the third row decides that it should be assigned to the reader at the bottom with probability $\frac{1}{2}$. Consequently, the bottom reader receives an expected load of $\frac{n}{2}$. However, it is not hard to devise an assignment where any reader is assigned at most two tags. This gives us a lower bound of $\Omega(n)$ on the approximation ratio that can be achieved by the simple one-round localized scheme.

Proposed optimization: Here, the observation is that readers in the second row are disadvantaged by the misleading fact that each of them is adjacent to n tags. This is misleading because each of the $n-1$ tags in the first row is also adjacent to n readers, not just that reader itself. Therefore, the actual expected load on each reader in the second row is far less than the nominal value of n. Based on this observation, if we run one more round of load balancing where we let l_i of each reader u_i be its expected load assigned in the previous round, denoted by

$$\sum_{v_j \in N(u_i)} p_{ij} \cdot c_{ij},$$

we will be able to reach a much more load balanced assignment. For the example in Figure 4, a second round of load balancing reduces the maximum load on readers to below 3. This maximum load occurs on the bottom reader. In general, if necessary this iterative optimization can be executed for more rounds to achieve even more load balanced assignments. To enable this *iterative LPA (ILPA)* scheme, each reader u_i needs to store the p_{ij} of each tag v_j in $N(u_i)$. We comment on the performance of this optimization in Section 7.

6. IMPLEMENTATION

In this section we will discuss how the centralized and localized load balancing schemes can be implemented in RFID systems compliant to EPC Generation 2 UHF RFID specifications [1].

Centralized Scheme: In this approach, each reader needs to communicate information about all tags in its vicinity to a central entity, e.g.,. the Reader Network Controller (RNC) [3]. Such communication will be possible using standardized protocols such as the IETF's Simple Light-weight RFID Reader Protocol (SLRRP) [3]. Once the appropriate tag to reader assignment has been calculated and communicated back to the readers, the relevant reader can take responsibility of reading the corresponding tags by appropriately parameterizing the 'select' message, used in the reading process [1].

Localized Scheme: For implementing the localized scheme we assume that neighboring readers in the system have unique identifiers (RID) and the read range and write range of a reader is same. The user memory of the tag can be used to store the RID and tag count pairs. Here the size of the user memory may become a constraint but in practical scenarios it is not expected that a tag would fall in the range of many readers. Once all the readers have written their RID and tag count, the tag computes the probabilities as described in Section 5 and chooses one of the RIDs by generating a random. It should be noted that the tags are capable of generating random numbers as it is an integral part of Q protocol. This RID is stored at a predefined location RIDLOC in the user memory. When performing the read operation, the readers include their RID in the *select* query which is matched against RIDLOC in the user memory of each tag. In this way tags respond only to that reader whose RID is written at RIDLOC and ignore other readers.

7. EVALUATION

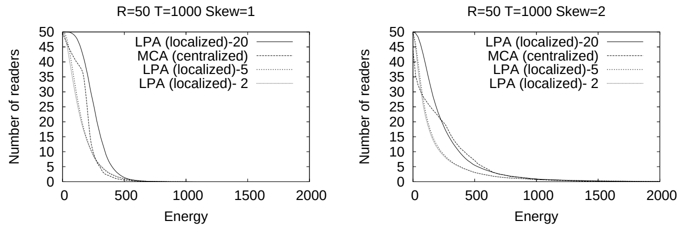
Here we present the simulation setup and assess the performance of centralized and localized load balancing algorithms in various RFID topologies. While we may use any general cost function for MCA, in this evaluation we use energy as a specific cost metric that we our formulation will minimize. Energy costs are only relevant to readers (tags have no power source of their own). For the MCA version of the problem the transmission energy used by readers is variable and is proportional to the square of the distance to the tags. For the MTA version, we assume the transmission energy used by readers is a fixed constant — as discussed before, this translates to balancing the number of tags across the readers.

Simulation Environment. All our experiments are performed by randomly deploying RFID tags and readers in a 1000×1000 square feet grid. The maximum transmission range of a reader is 12 feet as mentioned in Section 2. We analyze the efficacy of our proposed load balancing algorithms by varying the following parameters of the topology:

Tag Density: average number of tags in the range of a reader.

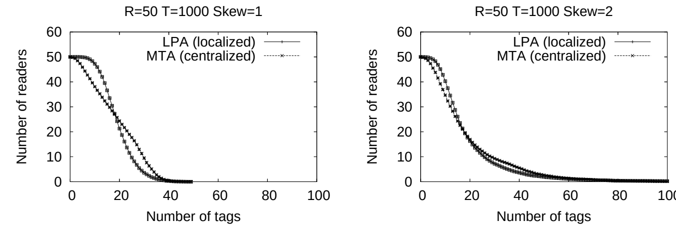
Skew: measures the variation in the number of tags in range of different readers. We implement this as follows. We assume readers are placed uniformly at random in the square area. For different values of a skew parameter, s, the x and y coordinates of tags are distributed is given by X^s, where X is a uniform random variable between $[0, 1]$. Greater the value of s, greater is the imbalance in the topology, i.e., there is a greater variability in the number of tags that are in range of different readers. By varying s, we therefore study topologies with different degrees of imbalance as might occur in practice.

Mobility: In typical RFID scenarios, we expect the number and position of the readers remains fixed, while the number of tags and their positions are highly dynamic will change over time. In this paper we evaluate such dynamic RFID systems by using two different mobility models that define the pattern of tag movement. *(i) Random Mobility*: At some time instants, chosen uniformly at random, some randomly chosen tags leave the system while new tags enter the system at random locations. The position of all the other tags remains unchanged. *(ii) Pattern-based Mobility*: Using the warehouse example, there will be specific instants when new tags enter the system, e.g., say a truck loaded with objects enter the warehouse. There will be other instants when exist-



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

Figure 5: 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

Figure 6: Tag load vectors of LPA and MTA with variation in skew. R and T refer to number of readers and tags respectively.

ing tags depart, e.g., a truck carrying another load of objects depart. We model such scenarios by varying the number of tags in the warehouse by increasing and decreasing the number of tags based on arrivals and departures of trucks. The number of tags in each truck is chosen uniformly at random.

Performance Metrics. To evaluate the efficacy of our proposed schemes we use the following metrics:

Load vectors : provide the entire distribution of cost for various readers in the system. Each element i in the load vector represents the number of readers whose cost exceeds i units. For the MCA problem, cost corresponds to energy consumption (we call it the Energy Load Vector or ELV), while for the MTA problem it corresponds to number of tags (we call it the Tag Load Vector or TLV). In addition, we also particularly examine the maximum and minimum loads for different schemes.

Jain's fairness index [12]: measures the fairness of different load balancing schemes using a single metric. For a load vector $\vec{L} = (l_1, l_2, ..., l_n)$, this is given by $((\sum_{i=1}^{n} l_i)^2)/(n \cdot \sum_{i=1}^{n} l_i^2)$ Intuitively, a load vector's Jain's Fairness Index is 1 if it is perfectly fair (i.e., all readers receive equal load), and is $\frac{1}{n}$ if it is completely unfair(i.e., only one reader is assigned all the tags and all other readers are idle).

Summary of results: 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, skew, and mobility models. LPA, with its low overheads, and limited need for interactions, is therefore a technique 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 can be found in the companion technical report [9].

Results

In this section, we present a comparison of LPA against the centralized algorithms for both static and mobile scenarios. The plots have been generated taking an average over 200 runs. We also report the 90% *confidence interval* of these runs, and since the bounds are tight, sometimes they are not clearly visible in the figures. 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 5, which shows the ELV plots for different skew parameters. For the LPA algorithm we use increments of 2, 5, and 20. LPA uses a greedy approach in acquiring tags, it does well in balancing load across readers. For LPA, when the increments are large, the load balancing across readers improve. (In fact, as our later discussion will show, that LPA-2 and LPA-5 leads to a better load balanced solution than MCA.) In comparison, the MCA algorithm successfully minimizes the maximum value, something which the localized algorithm does not match.

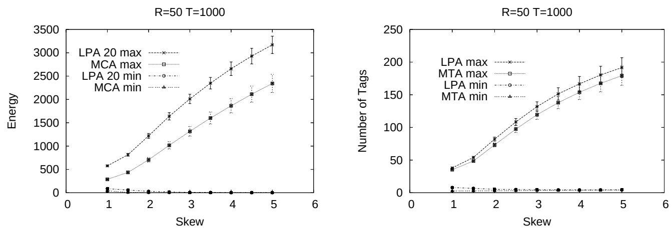
Finally, as the skew in the system increases, the achievable load balance is poorer for all algorithms.

In other experiments we observed that the performance of algorithms is relatively insensitive to varying tag densities in the environment. These results are omitted due to space constraints and can be found in the companion technical report [9].

LPA vs MTA. We compare the performance of LPA and MTA algorithms, with different skew values in Figure 6. As before, the load vectors for the two algorithms indicate that LPA achieves a better load balance — for the skew value of 1, the TLV curve for LPA remains higher than the TLV curve for MTA up to a load of 20 and for the latter part of the graph, the TLV curve for LPA remains below the TLV curve for MTA. In contrast, MTA achieves its goal of minimizing the maximum tag count. Note that the goal of the MTA algorithm is to minimize the maximum tag count only, while LPA has been designed more with a load balancing objective. The trend with increase in skew is similar as in the variable cost version of the problem.

We have also performed a second set of simulations varying the density of tags in the system keeping a constant skew of 2. As expected the behavior of LPA still remains the same and closely follows the behavior of MTA. The upper bound on the number of tags in the vicinity of any reader again remains equal to that for MTA.

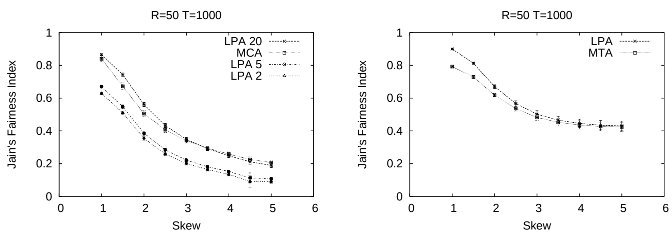
Maximum and minimum values. To demonstrate that, indeed, the MCA and MTA algorithms are better at minimizing the maximum energy and tag load values respectively, in Figure 7 we take a closer look at the maximum (and minimum) values of these metrics for topologies with different tag skews. In each case, the MCA and MTA algorithms has a lower maximum value of energy cost and tag load respectively, when compared to their LPA counterparts. The results also indicate that the LPA algorithm lags behind by only a small amount in most cases, implying that the cost



(a) Max and min energy costs (LPA and MCA).

(b) Max and min tag loads. (LPA and MTA).

Figure 7: Min and max metric values for different algorithms.



(a) LPA vs MCA.

(b) LPA vs MTA.

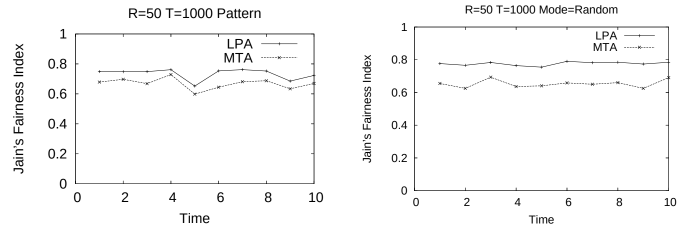
Figure 8: Jain's Fairness Index.

of its distributed implementation is not too high. Finally, the figure illustrates an increase in the maximum values of the corresponding metrics with an increase in skew of tag distribution in the system, as is expected.

Fairness. We next examine the fairness properties of the algorithms using the Jain's fairness index metric in Figure 8. In panel (a) we can observe that the LPA algorithm with iteration step of 20 has better performance than the MCA algorithm, while panel (b) indicates a superior fairness performance of LPA over the MTA algorithm. Finally, we observe that fairness decreased with increase in skew due to reduced opportunity of load balancing in such scenarios.

Mobility models. In our evaluation, the results observed in static scenarios was echoed in mobile scenarios as well. We present two snapshots, for the MCA case, from our diverse mobility experiments in Figure 9. The figure shows the variation of the Jain's fairness index over time as tags entered and left the RFID system. In panel (a) we present results for the random mobility model while in panel (b) we present results for the patterned mobility model. In the case of the reference centralized algorithm (MCA), the algorithm was executed immediately after each mobility event and provided the best case reference for the localized LPA algorithm.

Impact of tag storage capacity. RFID tags have very limited storage capacity so in LPA all the readers in the vicinity of a tag may not be able to append their cost values on the tag. We vary the tag storage capacity and assess the impact of limited storage on the performance of LPA on random topologies with varying skew. Figure 10 contrast the impact of limited storage on random topologies with skew of 1 and 3 respectively. As evident from these figures, the effect of limiting the storage is more profound in more evenly balanced topologies (skew 1) and has minimum effect on imbalanced topologies (skew 3). This can be attributed to the fact that imbalanced topologies are inherently difficult to



(a) Pattern-based mobility.

(b) Random Mobility.

Figure 9: Different mobility models to compare performance of the LPA and MTA algorithms for the tag load metric.

balance and limiting the storage does not effect the performance of LPA. Also Figure 10(c) shows the level of fairness achieved by varying storage capacities for the same skew of 3. It is evident from Figure 10(c) that the impact on fairness is maximum in the case of low skews and becomes negligible as skew increases.

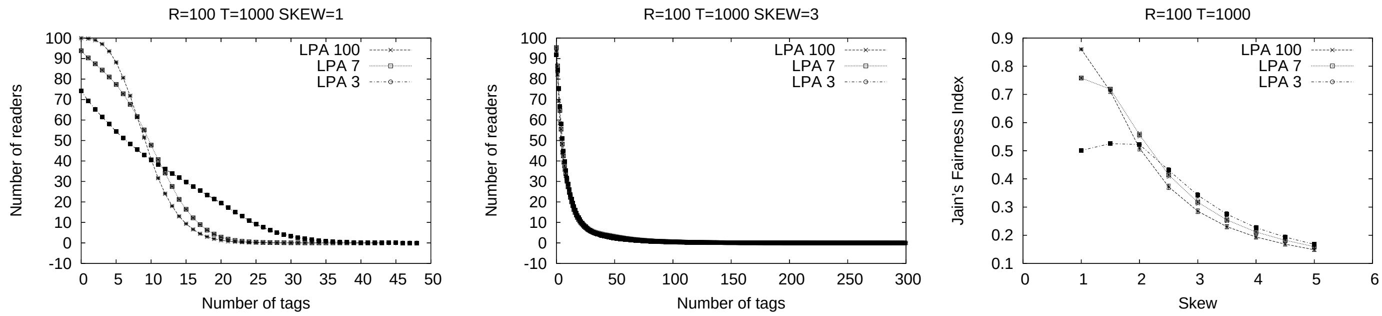
8. RELATED WORK

In the literature, Carbunar *et al.* [6] 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 [14], 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 [14]: whereas the algorithm in [14] 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 [13]. The authors adopt the same definition of lifetime as the time until first node failure, which is previously proposed by Chang and Tassiulas [7]. Therefore, their objective is also to minimize the maximum energy cost at any node. The key difference between their problem and our problem 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 clearly 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 [5] where the objective is to assign WLAN clients to access points (APs) in a load balanced manner. The edge between an



(a) TLV for Skew = 1 (b) TLV for Skew = 3 (c) Jain's Fairness Index

Figure 10: Effects of storage limit on tags on various metrics. The impact of tag storage limit is maximum for skew =1 and mitigates for skew = 3.

AP and a client also has a cost, which is inversely proportional to its effective bit rate. Their objective is also to find an assignment of clients to APs. However, the performance measure of an assignment is not the maximum cost of any AP. Instead, they try to optimize the 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.

As has been demonstrated in our analysis, our load balancing problems are also closely related to the classical minimum multiprocessor scheduling problem. We refer interested readers to the literature [11, 10, 16, 18, 4] for detailed results about the problem and a number of its variants.

9. CONCLUSIONS

In this paper, we study load balancing in large-scale RFID systems. Our objective is assigning tags to readers in such a way that the maximum total cost required at any reader to retrieve data from its assigned tags is minimized. The cost metric is general in nature and can be used to model various performance measures, e.g., energy costs, time taken to read tags, etc. For the purpose of illustration, in this paper we use energy costs as an example performance measure. We show that even with centralized knowledge about the system, this general cost 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 consider an interesting special case where readers use a fixed transmission power, and thus our objective is simply to minimize the maximum number of tags assigned to any reader. We show this problem is polynomially solvable with centralized knowledge, and present a conceptually very simple polynomial time algorithm for optimally solving it. We also propose a simple and effective localized scheme for the problems we study. Our results demonstrate that this extremely low cost scheme can achieve very good performance even in highly dynamic large-scale RFID systems.

10. REFERENCES

[1] EPCglobal. Class 1 Generation 2 UHF Air Interface Protocol Standard Version 1.0.9, 2005.

[2] http://www.alientechnology.com/products/.

[3] Reva Systems. Whitepaper: Simple Lightweight RFID Reader Protocol.

[4] Y. Bartal, S. Leonardi, A. Marchetti-Spaccamela, J. Sgall, and L. Stougie. Multiprocessor scheduling with rejection. In *ACM-SIAM SODA*, 1996.

[5] Y. Bejerano, S.-J. Han, and L. E. Li. Fairness and load balancing in wireless lans using association control. In *ACM MobiCom*, 2004.

[6] B. Carbunar, M. K. Ramanathan, M. Koyuturk, C. Hoffmann, and A. Grama. Redundant-reader elimination in rfid systems. In *IEEE SECON*, 2005.

[7] J. H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *IEEE INFOCOM*, 2000.

[8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms (Second Edition)*. MIT Press, 2001.

[9] 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*, July 2006.

[10] D. S. Hochbaum and D. B. Shmoys. Using dual approximation algorithms for scheduling problems: theoretical and practical results. *Journal of ACM*, 34(1):144–162, January 1987.

[11] E. Horowitz and S. Sahni. Exact and approximate algorithms for scheduling nonidentical processors. *Journal of ACM*, 23(2):317–327, April 1976.

[12] R. Jain, editor. *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, 1991.

[13] I. Kang and R. Poovendran. Maximizing static network lifetime of wireless broadcast adhoc networks. In *IEEE ICC*, 2003.

[14] M. Kodialam and T. Nandagopal. Fast and reliable estimation schemes in rfid systems. In *ACM Mobicom*, 2006 (To appear).

[15] J. Landt. The history of RFID. *IEEE potentials*, 24(4):8–11, 2005.

[16] J. K. Lenstra, D. B. Shmoys, and E. Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical Programming*, 46(3):259–271, 1990.

[17] S. E. Sarma, S. A. Weis, and D. W. Engels. RFID systems and security and privacy implications. In *CHES*, 2002.

[18] D. B. Shmoys and E. Tardos. Scheduling unrelated machines with costs. In *ACM-SIAM SODA*, 1993.