Computer Sciences Department

Load Balancing in Large-Scale RFID Systems

Qunfeng Dong
Ashutosh Shukla
Vivek Shrivastava
Dheeraj Agrawal
Suman Banerjee

Technical Report #1568

August 2006

UNIVERSITY OF WISCONSIN MADISON

Load Balancing in Large-Scale RFID Systems

Qunfeng Dong Ashutosh Shukla Vivek Shrivastava Dheeraj Agrawal
Suman Banerjee
Department of Computer Sciences, University of Wisconsin-Madsion
{qunfeng, shukla, viveks, dheeraj, suman}@cs.wisc.edu
Koushik Kar
Electrical, Computer, and Systems Engineering Department, Rensselaer Polytechnic Institute
koushik@ecse.rpi.edu

ABSTRACT

Large-scale radio frequency identifier (RFID) systems are being increasingly deployed in many applications such as supply chain automation. An 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. In particular, we study different variants of the load balancing problem. We first present centralized solutions to these variants. 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. Although probabilistic, our localized algorithms guarantee that each tag is continuously monitored by some reader at every instant. 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.

Keywords

Energy Efficiency, Load Balance, RFID

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.

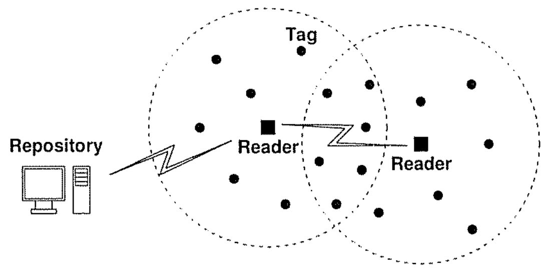


Figure 1: An example RFID system. Square nodes represent readers and round nodes represent tags.

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 multihop 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).

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* [19] 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 thus of central importance 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. (An analogous problem was first proposed for ad-hoc wireless networks by Chang and Tassiulas [6].) For simplicity, we refer to this problem as the *min-max cost assignment (MCA)* problem.

Using the energy efficiency analogy further, in many cases, it may also be the case that readers use a fixed transmission power for their interactions. Therefore, in this case 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 and 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 to 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 all of these problems by making the following key contributions.

- 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 also propose a simple and effective localized scheme for the problems we study. 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/readers 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 RFID systems.

The rest of the paper is organized as follows. Section 2 gives an overview of the RFID technology. 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 decribe 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

An typical RFID system comprises 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] which are widely used in supply chain and inventory management.

[1] defines the physical and logical requirements for a passive-backscatter, reader-talks-first RFID system operating in the 860 MHz - 960 MHZ frequency range. The reader transmits information to one or more tags by modulating an RF carrier using double-sideband amplitude shift keying (DSB-ASK), single-sideband amplitude shift keying (SSB-ASK) or phase-reversal amplitude shift keying (PR-ASK) using pulse-interval encoding (PIE) format. The Tag receives all their operating energy from this same modulated RF carrier. The reader receives information from the tag by transmitting an unmodulated continuous-wave RF signal to the tag and listening for a backscatter reply. The tag responds by backscatter-modulating the amplitude and/or phase of the RF carrier by changing the reflection coefficient of its antenna. The encoding format is either FM0 or Miller-modulated subcarrier.The system is reader-talk-first as the tag modulates its antenna reflection coefficient only after being directed to do so by the reader. Communication is half-duplex as the reader talks and tag listens and vice versa.

Each tag has a unique Electronic Product Code (EPC) identifier which identifies the product to which the tag is attached, a Tag Identifier (TID) which contains tag and vendor specific data and a read/write User memory which allows user-specific data storage and the organization of this memory is user-defined. Readers are capable of performing three basic operations:

Select The process by which a reader selects a tag population for Inventory and Access. The selection can be based on matching a bit sequence in EPC, TID or User memory.

Inventory The process by which a reader identifies (singulates) selected Tags and acquire their EPC identifiers. A slotted random anti-collision *Q protocol* is used for singulating tags. The reader chooses a slot-count parameter Q between 0 and 15 and broadcasts it in a *query* command. Upon receiv-

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

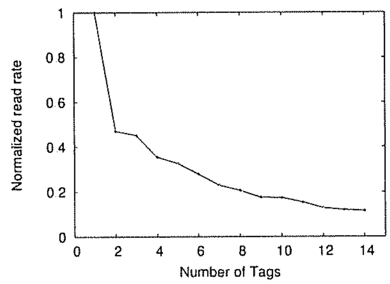


Figure 2: Decay in read-rate with increasing tag density

ing a *query* command, the tags selected in the Select phase pick a random value in the range (0,2^Q-1), inclusive and load this value into their slot counter. Tags that pick a zero reply immediately. If more than one tag replies there is a collision and the reader might not interpret the replies. If only one tag picks a zero, its reply can be understood by the reader resulting in singulation of the tag. This tag then moves into a sleep state and will not participate in this inventory round. Then the reader issues a command which causes the non-singulated tags to decrement their slot counter by one. Again, if any tag has a slot counter which reaches zero, it replies. The reader can also change the Q value in between and restart the inventory round. This goes on until all the tags have been singulated and have moved to the sleep state. This can be detected when the reader uses a Q value of zero and receives no replies.

Access The process by which a reader reads or writes to individual tags after singulation.

To study the performance of readers with increasing tag density we performed experiments using Alien ALR-9800 Generation 2 reader and ALL-9440 Squiggle tags [2]. The reader has a maximum read range of about 12 feets 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. The RF attenuation value ranges from 0 (no attenuation, maximum power) to 160 (maximum attenuation,minimum power), in increments of 10, each representing an additional 1db of RF attenuation. The reader can hold the EPC values of upto 6000 tags in its local memory.

In the experiment the tags were kept at distance of 6 feets from the reader antenna and the attenuation was 0. 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.

In Walmart kind of a scenario where readers are deployed to monitor the goods on the shelves, there are actually two kinds of events that a reader is interested in monitoring - new tags coming in its vicinity and old tags that were previously there in its vicinity but have now moved out. A simple way to find out such events is that each reader maintains a list of tags that it has seen. It polls all the tags in its list and if any tag does not reply it means it has left its vicinity and its entry is deleted. Now it can sleep all the tags in its list and singulate the new tags that have come in its vicinity and add them to the list. Here, it is possible that there would be tags that lie in the range of more than one readers. Given that read-rates reduce drastically with increasing tags, it becomes crutial that such tags are distributed to readers in a load balanced way and each tag is read by only one reader. This means that same tag should not have an entry in more than one list maintained by the readers for monitoring. In this paper, we propose load balancing algorithms that would run periodically and redistribute the tags to the readers. It should be noted that the new tags that come in the vicinity of more than one reader will have entry in the tag-list of multiple readers. But after doing load balancing such tags will have entry in a single reader's tag-list.

3. FORMULATION

In this paper, we consider RFID systems where readers and tags are equipped with omnidirectional antennas, which is much simpler and hence more cost efficient than directional ones [17]. 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.

Now we turn to a key problem in modeling a RFID system: *How can we decide if a pair of reader and tag can communicate with each other or not?* In the literature, there are two most popular models that have been widely used to model wireless communication systems using omnidirectional radios. In the *disk graph* model, each node is assumed to have a certain transmission range. Because radio signal attenuates as it propagates from the sender to the receiver [18]. In order for the receiver to correctly decode the transmitted information, the received signal must be strong enough to pass some threshold signal/noise ratio (SNR). Assuming that the sender transmits at its maximum transmission power, the range within which the transmitted signal can be correctly decoded is referred to as the *transmission range* of the sender. The disk graph model is based on the ideal assumption that if node v is located within the transmission range of another node u, then node u must be able to reach node v.

In RFID systems, we assume that readers have the same transmission range R and tags have the same transmission range r, where $R > r$. However, communication between readers and tags are bi-directional. If reader u_i can reach tag v_j but the latter cannot reach the former, then u_i cannot retrieve data from v_j, then the uni-directional link from u_i to v_j simply cannot be used in data retrieval. Therefore, we can safely pretend as if all the readers and tags have the same transmission range r. This eventually leads to the widely used **unit disk graph (UDG)** model as defined in [7].

While the disk graph model is conceptually simple and elegant, in real applications its ideal assumptions are often considered not perfectly accurate. In real world, the transmission range of an omnidirectional radio can hardly be a perfect disk,

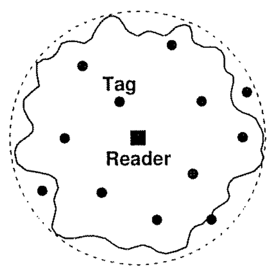


Figure 3: Irregular transmission range of a reader.

due to a variety of reasons such as obstacles, multi-path reflection, interference, etc. Instead, it is most likely an irregular region as shown in Figure 3. Therefore, researchers have also been using another most popular model, the **general graph** model. In the general graph model, any pair of nodes in the system can be connected by a link. Therefore, this model can accommodate any system. Note that the UDG model is a special case of the general graph model. In this paper, we study our problems in both models. In particular, we prove stronger hardness results for the restricted UDG model and prove stronger positive results for the general graph model.

Problem definitions: In this paper, we study the min-max optimization problem where our goal is to find an *assignment* $\varphi : V \rightarrow U$ 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 $\varphi : V \rightarrow U$ such that for each $u_i \in U$,
>
> $$\sum_{\substack{1 \le j \le n \\ u_i = \varphi(v_j)}} c_{ij} \le B?$$

An interesting special case of the MCA problem is the *min-max tag count assignment (MTA)* problem, where readers cannot adjust their transmission power and thus each edge has a fixed unit energy cost, namely $c_{ij} = 1$. Intuitively, our objective in MTA is to minimize the maximum number of tags assigned to individual readers.

Note that in these problems, energy costs in the UDG model should not be simply determined by the distance between endpoints, which is frequently used in the literature. For example, in indoor applications like inventory management, energy cost is a highly irregular function of distance because of obstacles and multi-path reflection, etc. In the MTA problem energy costs are always 1, regardless of the distance between endpoints. Moreover, we here aim to study a general problem where link costs may also have physical meanings other than energy cost. Therefore, the UDG model only determines the existence of edges between nodes, but not their costs.

4. CENTRALIZED SCHEMES

In this section, we formally analyze the complexity of the MCA problem and the MTA problem in the centralized setting. In particular, we prove that even in the restricted 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 $\frac{3}{2}$. The NP-hardness proof of MCA can be found in Appendix. These hardness results automatically hold in the general graph model. Given that, we provide an efficient 2-approximation algorithm for the general graph model, which comes very close to the optimal solution and is guaranteed to be at most 2 times the optimum in the worst case. The approximability and inapproximability results for MCA are based on equally simple reductions between MCA and the *minimum multiprocessor scheduling (MMS)* problem, which possesses the same approximability and inapproximability properties. 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)

Given the NP-hardness of MCA, we cannot expect to find an efficient algorithm for computing the optimal solution, unless $P = NP$. Instead, we should try to design an efficient approximation algorithm $\mathcal{A}$ that can find an assignment φ such that the maximum total cost over all readers is at most α times as large as the optimal solution. α is referred to as the *approximation ratio* of $\mathcal{A}$ and gives us some idea of the approximability of the problem in study. On the other hand, if we are not able to achieve any better approximation ratio, it will be useful to figure out some lower bound on the achievable approximation ratio. Such a lower bound can give us some idea of the inapproximability of the problem in study.

4.1.1 Approximability

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)* problem, which is approximable within a factor of 2 [16]. 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* $\phi : T \rightarrow P$ 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. In [16], Lenstra *et al.* have proposed an approximation algorithm for MMS that guarantees to find for any instance of MMS a schedule ϕ such that the finish time is at most two times as much as the optimal solution. We here demonstrate that the same approximation ratio of 2 can be achieved for MCA as well, simply by reducing MCA to MMS.

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.

The transformation is clearly polynomial, and we next prove that the optimal solution of the input instance of MCA (denoted by OPT_{mca} is always equal to the optimal solution of the constructed instance of MMS (denoted by OPT_{mms}).

PROOF. ($\Rightarrow$) Given any assignment φ for the MCA instance, we can define a schedule ϕ for the constructed MMS instance such that for each pair of task t_j and processor p_i

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

It is easy to verify that if φ leads to a maximum total cost of C, then ϕ leads to a finish time of C as well. Therefore, we have $OPT_{mca} \geq OPT_{mms}$.

($\Leftarrow$) Given a schedule ϕ for the constructed MMS instance, we can define an assignment φ for the given MCA instance such that for each pair of reader u_i and tag v_j

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

It is easy to verify that if ϕ leads to a finish time of C, φ leads to the same maximum total cost of C. Therefore, we have $OPT_{mca} \leq OPT_{mms}$. $\square$

Without loss of generality, let $\mathcal{A}$ denote the best known approximation algorithm for MMS whose approximation ratio 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

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

The maximum total cost C derived from φ satisfies $C \leq \alpha \cdot OPT_{mms} = \alpha \cdot OPT_{mca}$.

4.1.2 *Inapproximability*

In [16], Lenstra *et al.* have also proved 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. Again, the same inapproximability result automatically holds for the general graph model as well. Given an instance of MMS, we transform it into an instance of MCA in the UDG model as follows.

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

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

(3) Set the transmission range R of the readers to be sufficiently large to cover all the tags, and also set the transmission range r of the tags to be sufficiently large to cover all the readers.

(4) For each pair of processor p_i and task t_j, add an edge between the corresponding u_i and v_j, whose cost is $c_{ij} = l_{ij}$.

The transformation is clearly polynomial, and we can similarly show that the optimal solution of the input instance of MMS (denoted by OPT_{mms} is always equal to the optimal solution of the constructed instance of MCA (denoted by OPT_{mca}).

PROOF. ($\Rightarrow$) Given any schedule ϕ for the MMS instance, we can define an assignment φ for the constructed MCA instance such that for each pair of reader u_i and tag v_j

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

It is easy to verify that if ϕ leads to a finish time of C, φ leads to the same maximum total cost of C. Therefore, we have $OPT_{mca} \leq OPT_{mms}$.

($\Leftarrow$) Given any assignment φ for the constructed MCA instance, we can define a schedule ϕ for the given MMS instance such that for each pair of task t_j and processor p_i

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

It is easy to verify that if φ leads to a maximum total cost of C, then ϕ leads to a finish time of C as well. Therefore, we have $OPT_{mca} \geq OPT_{mms}$. $\square$

Assume $\mathcal{A}$ is the best known approximation algorithm for MCA whose approximation ratio is α. We can define an α-approximation algorithm for MMS that is composed of the following three phases. (1) Transform the input MMS instance into an MCA instance as described above. (2) Apply $\mathcal{A}$ on the constructed MCA instance to compute an assignment φ. (3) Define a schedule ϕ for the given MMS instance such that for each pair of processor p_i and task t_j

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

The finish time C of ϕ satisfies $C \leq \alpha \cdot OPT_{mca} = \alpha \cdot OPT_{mms}$. Therefore, since MMS cannot be approximated within a factor of less than $\frac{3}{2}$ unless $P = NP$, the same inapproximability result holds for MCA as well.

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, i.e., an assignment of each tag v_j to some reader u_i such that the maximum number of tags (i.e., load) assigned to readers is minimized.

At the high level, our MTA algorithm is essentially an iterative binary search process, which may start with some obviously feasible load (e.g. n) Within each iteration, we test some specific load B to see if there exists some assignment $\varphi : V \rightarrow U$ 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 (according to the standard binary search algorithm). Otherwise, we increase the value of B (according to the stand binary search algorithm). Eventually, the binary search process will converge to the minimum feasible load.

Now it only remains to design an algorithm for the feasibility test of B. Namely, to answer the decision version of the MTA problem with the bound B, which is basically the same as the decision version of the MCA problem presented in Section 4.1, except that link costs are always 1 in MTA. Here, we solve this problem by reducing MTA to the *maximum network flow (MNF)* problem. Given an instance of the 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 4. (1) Create a virtual source s and a virtual sink t. (2) For each reader $u_i \in U$ in the given MTA instance,

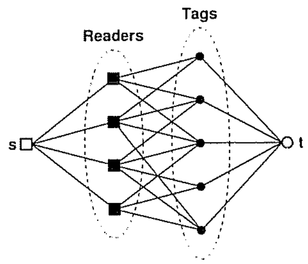


Figure 4: Transformation from MTA to MNF.

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 now prove 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. 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.

PROOF. ($\Rightarrow$) If the given MTA instance has an assignment $\varphi : V \rightarrow U$ such that each reader receives at most B tags, in the constructed MNF instance a flow of n can be routed from s to t as follows. (1) For each edge (u_i, v_j), if $\varphi(v_j) = u_i$, assign a flow of 1 from u_i to v_j; otherwise, edge (u_i, v_j) should carry no flow. (2) For each edge (s, u_i), assign a flow from s to u_i that is equal to the aggregate outgoing flow from u_i, so that flow conservation is satisfied at u_i. Since φ assigns at most B tags to each tag, we are guaranteed that (s, u_i) carries a flow of at most B. (3) Assign a flow of 1 to each edge (v_j, t).

($\Leftarrow$) If the constructed MNF instance admits an integral flow of n from s to t such that each edge carries a non-negative integral flow, then an assignment $\varphi : V \rightarrow U$ for the given MTA instance can be easily defined as such that $\varphi(v_j) = u_i$ if and only if edge (u_i, v_j) carries flow. Since each u_i has at most B incoming flow from s, it has at most B outgoing flow as well, due to flow conservation. Consequently, the φ we have defined assigns at most B tags to each reader. $\square$

Having proven that, we can simply apply a standard maximum flow algorithm [8] on the constructed MNF instance. If the maximum achievable flow is equal to the total number of tags, n, we know that bound B is achievable. If the maximum achievable flow is less than n, that means bound B is not achievable.

5. LOCALIZED SCHEME (LPA)

In previous sections, we have studied the complexity of MCA and MTA. The algorithms we have proposed are all presented as centralized algorithms. While these algorithms are shown to 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.Dynamic updates (i.e., join/leave of tags/readers) can be efficiently handled as well. 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

Localized vs Distributed: A distributed scheme is not enough. In principle, every algorithm can be implemented in a distributive manner. The most straightforward solution is for some node to serve as the central control: it collects relevant information from all other nodes, locally executes a centralized algorithm to compute a solution, and then floods the computed solution back to other nodes. In this paper, we aim to design a distributed scheme with extremely low message complexity. Therefore, we turn our attention to localized algorithms, which are a special kind of distributed algorithms where there is no central control needed and each node only needs knowledge about one-hop neighbors. A lot of state maintenance as well as communication overheads can thus be avoided in such a localized scheme.

Randomized vs Deterministic: So far we have been focused on deterministic 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 expected load on each reader sums up to 1.5 only, which is even 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. First of all, note that in a localized scheme each node only has knowledge about its one-hop neighbors. From the perspective of load balancing, 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. Thus, a localized scheme can only rely on such local information. Second, 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. Otherwise, the tag should report to it with a relatively high 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

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

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 [5], 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 5, 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 read-

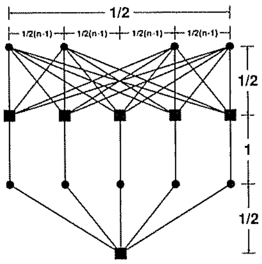


Figure 5: An analysis of the simple localized scheme. Square nodes represent readers and round nodes represent tags. Each node has a transmission range of 1. Each edge has a cost of 1.

ers. 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 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 5, 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].

6.1 Centralized Scheme

Typical RFID deployment comprise of a network of RFID readers controlled by one or more Reader Network Controller (RNC) [12]. The RNCs are connected to the host/servers running client applications that consume the acquired tag data. Standards for communication between RFID readers and backend client application have been developed by various standards bodies. IETF has proposed a Simple Lightweight RFID Reader Protocol (SLRRP) [12] to convey configuration, control parameters and transfer tag information to and from readers having TCP/IP stack. This infrastructure can be easily used for implementing the proposed centralized load balancing schemes. Using SLRRP, the readers can transfer the EPC of the tags in their vicinity to the backend server which can then compute the load balanced assignment and convey it back to the readers.

6.2 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 is 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 citesec:localized 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 slelect 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 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 X 1000 square feet grid. The maximum transmission range of a reader is 12 feets 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. By varying tag density, we can evaluate our scheme on increasing loads of tags per reader.

Skew Skew is defined as the variation in the number of tags in the range of various readers in the system. We evaluate our scheme on inherently imbalanced topologies, where most of the tags are clustered in the vicinity of a few readers while other readers have very few tags in their vicinity. Such imbalanced topologies are quite possible in warehouses and supermarkets, where tag mobility over a period of time can lead to different densities of tags in different parts of the store. 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 most practical RFID systems, the number and position of the readers remains fixed, while the number of tags and their positions are highly dynamic and may change across very small time intervals. Here we analyze such dynamic RFID system by using mobility models which define the pattern of tag movement. Note that in these mobility models, the position of the RFID readers remain fixed while the position and number of RFID tags change as tags enter and leave the system. We use the following mobility models to capture the dynamics of a RFID topology:

Random Mobility Model: Here 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. Here the number of tags between any two instants of time vary randomly. So the overall number of tags and their position varies randomly between any periods of interest.

Pattern-based Mobility Model: Using the warehouse example again, it is quite likely that the number of tags in the system will change over time. There will be specific periods when new tags enter the system, e.g., say new truckloads of objects enter the warehouse. There will be other periods when existing tags depart, e.g., truckloads of objects are carried away.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 : 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).

Fairness: We use Jain's fairness Index [13] to evaluate the fairness provided by individual schemes. The Jain's Fairness Index for a load vector $\vec{L} = (l_1, l_2, ..., l_n)$ is given by

$$\frac{(\sum_{i=1}^{n} l_i)^2}{n \cdot \sum_{i=1}^{n} l_i^2}$$

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).

Maximum load: It is defined as the maximum load on any reader in the RFID system after MTA, MCA or LPA algorithms have performed load balancing.. We use maximum load on any reader in the system as an indication of the efficacy of MCA and MTA algorithms. We also compare LPA algorithm with centralized algorithms on this metric to evaluate the ability of probabilistic assignment in minimizing the maximum load on the system.

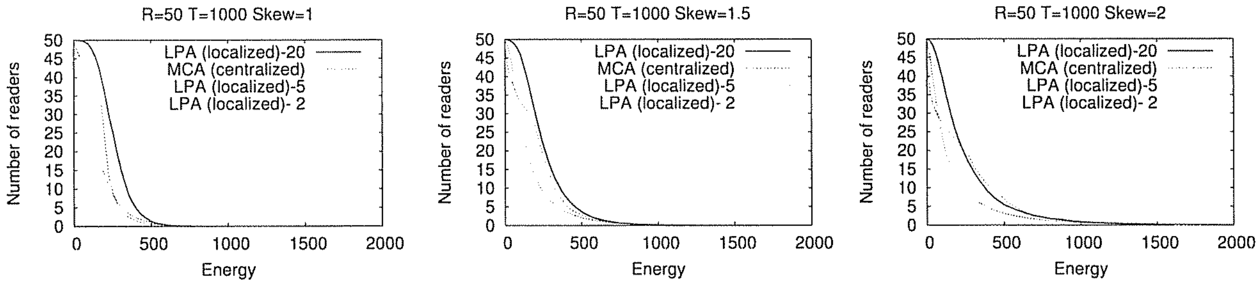
Summary of results: Our results reported next can be summarized as follows: The proposed localized heuristic (LPA) performs nearly as well as the various optimal and near-optimal centralized algorithms (MTA and MCA) across a wide-range of scenarios. LPA, with its low overheads, and limited need for interactions, is therefore a technique for efficient load balancing in RFID systems.

Results

The results are structured as follows: We first compare the efficiency of LPA with MTA for balancing number of tags in the RFID system. Next we compare the performance of LPA with MCA for balancing the energy consumption for the readers. Finally, we evaluate the stability of the localized algorithm in dynamic RFID systems where readers are fixed and tags appear and disappear at random locations in the system. We plot TLV and ELV to compare the performance of various load balancing schemes. The plots have been generated taking an average over 200 runs of random topologies with the same skew and same number of readers and tags. We have also reported the 90% *confidence interval* of these runs, and since the bounds are tight, 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.

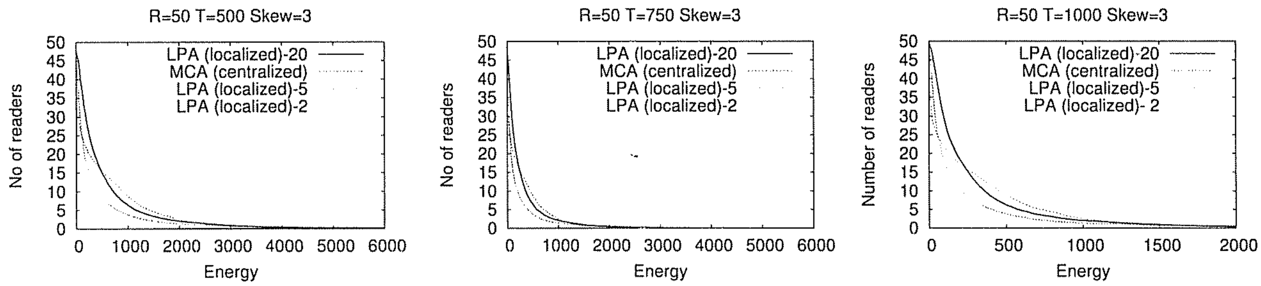
LPA vs MCA. Here we present the performance comparison of LPA and MCA for balancing energy consumption of readers in RFID system. Figures 6 and 7 compare the performance of MCA and LPA for energy assignment. Figures 6(a) to 6(c) show the ELV plots for increasing skews. Each plot shows the ELV's for LPA with increments of 2, 5 and 20 and ELV for MCA. For energy assignment, the LPA uses a greedy approach to acquire tags resulting in lower total load on the readers but fails to limit the upper bound load on any reader which is reflected in Figure 6(a) .However, if the increments are large the readers will have a better understanding of the topology and will do better load balancing. Therefore, the ELV's of LPA for increments of 2 and 5 fall drastically whereas that of increment 20 fall gradually. Note that in this case 20 is the maximum range of the reader and hence results in only one iteration of the algorithm, which is similar to the normal LPA with variable cost. By increasing the skew to 1.5 (Figures 6(b)) and then to 2.0 (Figure 6(c)) the ELV's for both LPA and MCA fall sharply. This is because as the skew in the system increases, few readers have large number of tags in their vicinity while others have few tags. So most of the tags cannot be distributed in a balanced manner between the readers leading to a sharp fall in the load vector. The performance only worsens with a higher skew of 2. In Figures 7 (a) to (c), plotted for increasing tag density and a skew of 3, again as expected the behavior of the ELV for both LPA and MCA do not change for different tag densities and also the vectors are very close to each other owing to the reason mentioned above. Note that the rate of fall in ELV increases with increasing tag density. This is because the effect of skew is enlarged with increasing tag densities.

LPA vs MTA. We first examine the effect of changing skew on LPA and MTA algorithms in figure 8. As can be seen from figure 8(a) , for a low skew value of 1, the TLV for LPA remains significantly higher than the TLV for MTA up to a load of 20. This implies that for LPA more readers have at least



(a) Skew = 1 (b) Skew = 1.5 (c) Skew = 2.0

Figure 6: 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) Tag Density = 10 (b) Tag Density = 15 (c) Tag Density = 20

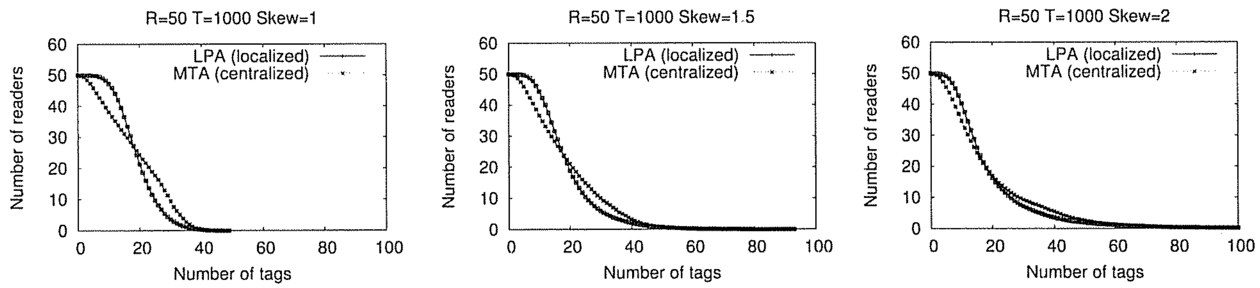
Figure 7: Energy load vectors of LPA and MCA with variation in tag density. R and T refer to number of readers and tags respectively. Tag density has no impact on the realative performance of the algorithms.

a threshold number of tags leading to better load balancing. For the latter part of the graph, the TLV for LPA remains below the TLV for MTA which is also advantageous as we would also want to minimize the number of readers having large number of tags. This can be attributed to the fact that MTA only aims at minimizing the maximum load but LPA is targeted towards load balancing and hence assigns most of the readers at least 10 tags as shown by the initial flatness in the curve. LPA is also able to achieve the same upper bound of 49 as that for MTA. However, with increasing skews (Figures 8(b) and 8(c)) the system reaches extremes where some readers have very high load whereas some other have very low load. For this reason, LPA performance degenerates for higher skew but still closely resembles that of MTA. This effect is only exacerbated in figure 8(c) where the skew is much higher. However, the upper bound for LPA is still the same as compared to MTA in both Figures 8(b) and 8(c).

We performed a second set of simulations varying the density of tags in the system keeping a constant skew of 2, as shown in Figure 9. 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.

Fairness. Figure 11 illustrates the fairness comparisons between the algorithms for increasing skews using Jain's fairness index as the metric of comparison. From figure 11(a),which is plotted for the energy assignment, the fairness index values for LPA with iteration step of 20 remains the highest followed by MCA and then LPA with iteration step of 5 and 2. This trend is seen because the objective of LPA is to balance load thereby leading to higher fairness. On the other hand, MCA only tries to minimize the maximum load on any reader and therefore might fail to take care of fairness. However, the fairness index values for LPA with small iteration steps of 2 and 5 is lower than that of MCA because increased greediness leads to readers getting assigned many tags with low energy resulting in higher cost on the reader and also that the readers which have acquired tags from previous lower ranges now have to accept tags in higher ranges which are covered by no other readers resulting in an unbalanced assignment of loads. Figure 10(a) and 11(c) compare the fairness of LPA and MTA over time for *pattern based* and *random* mobility models respectively. Again, the fairness index of LPA always remains above that of MTA. The variation of the fairness index over time is very small for each algorithm.

Maximum-minimum bounds. Since MCA gives only twice as bad an upper bound from the optimal solution it outper-

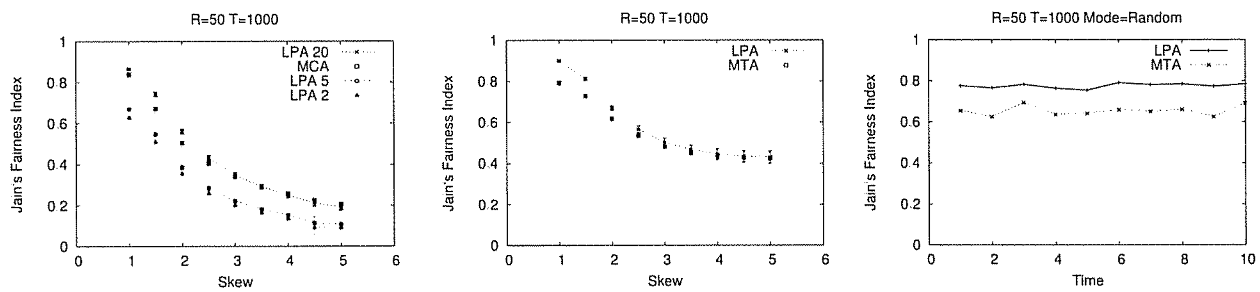


(a) Skew = 1

(b) Skew = 1.5

(c) Skew = 2

Figure 8: Tag load vectors of LPA and MTA with variation in skew. R and T refer to number of readers and tags respectively. The upper bound for tags assigned to any reader for LPA is slightly greater than that of MTA.



(a) Energy Balancing. Fairness index degrades rapidly with increase in skew. LPA(20) performs the best and is closely followed by MCA for varying skews.

(b) Tag Balancing. LPA performs better than MTA, however the gap decreases with increasing skew

(c) Random Mobility Model. LPA performs better than MTA by considerable margin

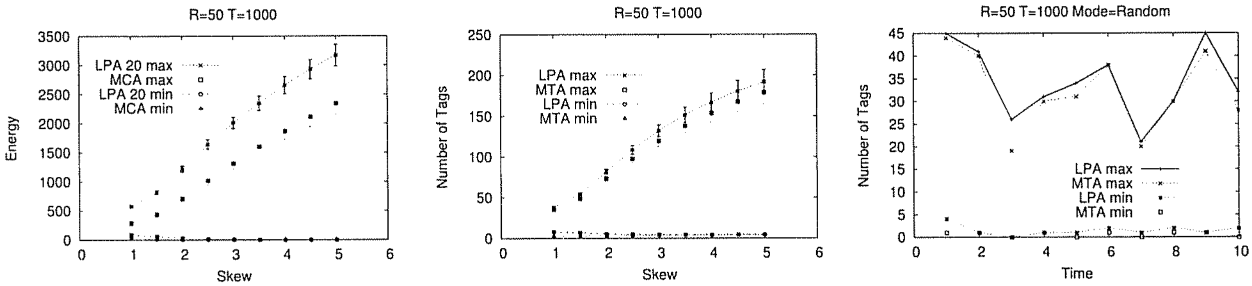
Figure 11: Jains Fairness Index.

forms LPA on keeping the upper bound on the energy costs low as seen in Figure 12(a). Again, due to the greedy nature of LPA for iteration steps of 2, the upper bound for it is the worst. As shown in Figure 12(b), for the tag assignment costs, MTA performs better than LPA as MTA provides an optimal solution for the minimum upper bound. However, the upper bounds for LPA are still quite close to that of MTA providing a reasonable upper bound. Figure 12(c) and 10(b) show the maximum bounds *random* and *pattern-based* mobility models respectively. We observe that over time, the maximum bounds for LPA in both the models remains quite close to MTA. Hence, it clearly shows the efficacy of LPA for providing near-optimal bounds in both the models.

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 13 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 balance and limiting the storage does not effect the performance of LPA. Also Figure 13(c) shows the level of fairness achieved by varying storage capacities for the same skew of 3. It is evident from Figure 13(c) that the impact on fairness is maximum in the case of low skews and becomes negligible as skew increases.

8. RELATED WORK

In this paper, we address the tag assignment problem in RFID system, which results from the *tag collision* problem as introduced in Section 1. In the literature, Carbunar *et al.* [5] 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

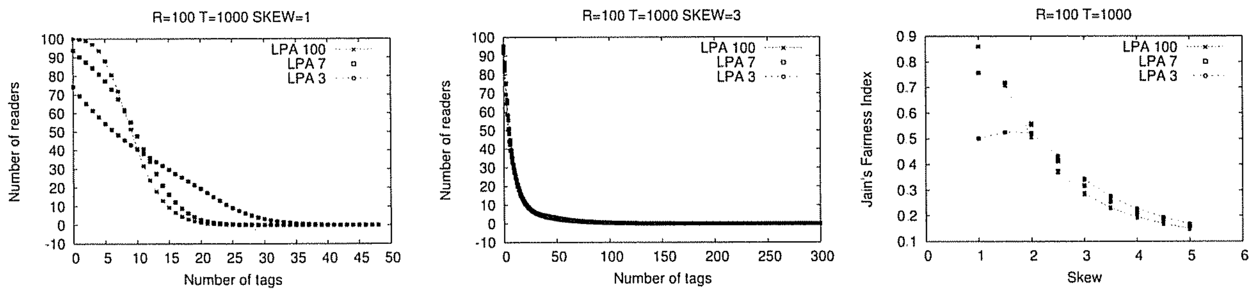


(a) Energy Balancing. MCA performs better than LPA and the performance gap increases with increasing skew

(b) Tag Balancing. MCA performs better than LPA by a small margin and the performance gap increases with increasing skew

(c) Random Mobility. MCA is slightly better than LPA over time

Figure 12: Min Max assignment comparison for LPA, MTA and MCA.



(a) TLV for Skew = 1

(b) TLV for Skew = 3

(c) Jain's Fairness Index

Figure 13: 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.

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.

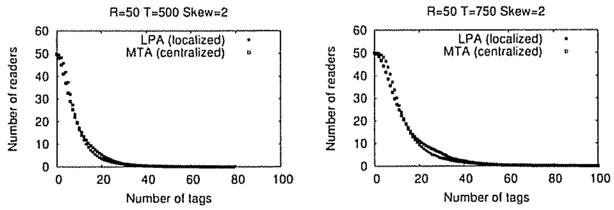
Another related work comes from the well researched maximum lifetime broadcast problem [14]. The authors adopt the same definition of lifetime as the time until first node failure, which is previously proposed by Chang and Tassiulas [6]. Therefore, their objective is also to minimize the maximum energy 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 [4] where the objective is to assign WLAN clients to access points (APs) in a load balanced manner. The edge between an 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. We refer interested readers to the literature [4] for further details. 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, 20, 3] for detailed results about the problem and a number of its variants.

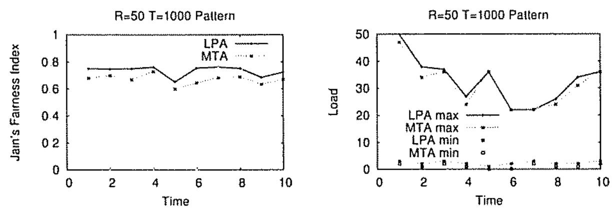
9. CONCLUSIONS

In this paper, we study load balancing in large-scale RFID



(a) Tag Density = 10 (b) Tag Density = 15

Figure 9: Tag load vectors of LPA and MTA with variation in tag density. Tag density has minimal impact on the relative performance of LPA and MTA. Though the maximum number of tags assigned to any reader increases with density.



(a) Jain's Fairness Index : Over the period of time, LPA balances load more efficiently than MTA

(b) Minimum and Maximum tag count : Maximum tags assigned to any reader is greater for LPA as compared to MTA

Figure 10: Results for Pattern-based mobility model.

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

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

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

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

[7] B. N. Clark, C. J. Colbourn, and D. S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1-3):165–177, 1990.

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

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability : A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

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

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

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

[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] R. Ramanathan. On the performance of ad hoc networks with beamforming antennas. In *ACM MobiHoc*, 2001.

[18] T. S. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall, first edition, 1996.

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

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

Appendix: MCA is NP-hard in the UDG model

PROOF. Our NP-hardness proof of MCA in the UDG model is based on an easy reduction from the PARTITION problem, which is well known to be NP-hard [9]. The decision version of the PARTITION problem is as follows: given a set of el-

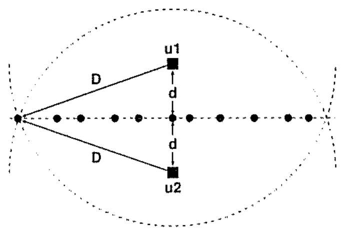


Figure 14: Transformation from PARTITION to MCA in the UDG model.

ements each having a positive *size*, can we partition this set into two subsets of equal total size? A formal definition of the PARTITION problem is given below.

INSTANCE Set $A = \{a_1, a_2, \cdots, a_n\}$ and a size $s(a_i) \in Z^+$ for each $a_i \in A$.
QUESTION Is there a subset $A' \subseteq A$ such that $\sum_{a \in A'} s(a) = \sum_{a \in A-A'} s(a)$?

Given an instance of PARTITION, we transform it into an UDG instance of MCA as follows. For simplicity, let D and d denote the maximum size and minimum size of elements in the given PARTITION instance, respectively.

(1) Create two reader nodes u_1 and u_2, which are at a distance of $2d$ from each other as shown in Figure 14. Moreover, both of them have a transmission range of D.

(2) For each element $a_i \in A$, create a tag node v_i and connect it with both u_1 and u_2 using an edge of cost $s(a_i)$. Note that if necessary, we can also place v_i at one of the points whose distance to u_1 and u_2 are both $s(a_i)$.

(3) Define $B = \frac{1}{2}\sum_{i=1}^{n} s(a_i)$.

This transformation is clearly polynomial, and it only remains to prove that there is a subset $A' \subseteq A$ such that $\sum_{a \in A'} s(a) = \sum_{a \in A-A'} s(a)$ if and only if there is an assignment $\varphi : V \to \{u_1, u_2\}$ such that

$$\sum_{\substack{1 \le j \le n \\ u_1 = \varphi(v_j)}} s(a_j) \le B \quad \text{and} \quad \sum_{\substack{1 \le j \le n \\ u_2 = \varphi(v_j)}} s(a_j) \le B.$$

($\Rightarrow$) If there is a subset $A' \subseteq A$ such that $\sum_{a \in A'} s(a) = \sum_{a \in A-A'} s(a) = B$, the assignment φ we are looking for simply assigns elements in A' to u_1 and the other elements in $A - A'$ to u_2.

($\Leftarrow$) If there is an assignment $\varphi : V \to \{u_1, u_2\}$ such that

$$\sum_{\substack{1 \le j \le n \\ u_1 = \varphi(v_j)}} s(a_j) \le B \quad \text{and} \quad \sum_{\substack{1 \le j \le n \\ u_2 = \varphi(v_j)}} s(a_j) \le B,$$

it follows the definition of $B = \frac{1}{2}\sum_{i=1}^{n} s(a_i)$ that

$$\sum_{\substack{1 \le j \le n \\ u_1 = \varphi(v_j)}} s(a_j) = B \quad \text{and} \quad \sum_{\substack{1 \le j \le n \\ u_2 = \varphi(v_j)}} s(a_j) = B.$$

Therefore, we can simply define the subset A' to be either $A' = \{a_i | \varphi(a_i) = u_1\}$ or $A' = \{a_i | \varphi(a_i) = u_2\}$. $\square$