Slotted Scheduled Tag Access in Multi-Reader RFID Systems*

Zongheng Zhou
Ask.com
Email: zongheng@gmail.com

Himanshu Gupta, Samir R. Das
Dept. of Computer Science, Stony Brook University, NY
Email: {hgupta,samir}@cs.sunysb.edu

Xianjin Zhu
Microsoft Corp.
Email: xianjin@gmail.com

***Abstract*— Radio frequency identification (RFID) is a technology where a *reader* device can "sense" the presence of a close-by object by reading a *tag* device attached to the object. To improve coverage, multiple RFID readers can be deployed in the given region. In this paper, we consider the problem of slotted scheduled access of RFID tags in a multiple reader environment. In particular, we develop centralized algorithms in a slotted time model to read all the tags using near-optimal number of time slots. We consider two scenarios – one wherein the tag distribution in the physical space is unknown, and the other where tag distribution is known or can be estimated a priori. For each of these scenarios, we consider two cases depending on whether a single channel or multiple channels are available. All the above version of the problem are NP-hard. We design approximation algorithms with logarithmic bounds for the single channel and heuristic algorithms for the multiple channel cases. Through extensive simulations, we show that for the single channel case, our heuristics perform close to the approximation algorithms. In general, our simulations show that our algorithms significantly outperform Colorwave, an existing algorithm for similar problems.**

I. Introduction

RFID is an identification system that consists of readers and tags [1]. A tag has an ID (a bit string) stored in its memory. The reader is able to read the IDs of the tags in the vicinity by running a simple link-layer protocol over the wireless channel. In a typical RFID application, tags are attached to objects of interest, and the reader detects presence of an object by using an available mapping of IDs to objects. RFID tags can be *active* or *passive* depending on whether they are powered by battery. We focus on passive tags in this work. Passive tags are prevalent in supply chain management as they do not need a battery to operate. This makes their lifetime unlimited and cost negligible (only few US cents per tag). The power needed for passive tags to transmit their IDs to the reader is "supplied" by the reader itself.

An important performance metric of RFID systems is *read throughput* (number of tags read per time slot). High read throughput is critical when tags are exposed to readers only briefly. This happens when tags are mobile, as is often the case in supply chain management or manufacturing environments. So far, the research community has addressed the read throughput problem for a single reader only. However, large-scale RFID deployments in future will hardly involve a single reader. This is because each RFID reader has a limited *interrogation* region within which it can communicate with a tag. The interrogation region of a reader depends on many factors including antenna, presence of obstacles, tag characteristics, etc. It is not uncommon that a single reader is unable to cover the entire region of interest. This motivates the use of multiple RFID readers – geographically dispersed and networked in some fashion (in an ad hoc network, e.g.) – performing tag reading concurrently. Use of multiple readers not only improves coverage, but also improves read throughput by virtue of concurrent operation.

However, several collision problems might occur when multiple readers are used within close vicinity. This makes deployment of multiple readers a very different problem than a traditional sensor cover problem [9]. The collisions are not easy to handle either. Unlike traditional wireless networking, in RFID we deal with two different entities – readers and tags. The collision can happen in either of these two entities giving rise to newer issues. Collisions at tags are particularly problematic as tags have almost zero computing power. This makes carrier sense-based collision resolution either hard or overly conservative [15]. In this paper, we take a very different approach. We use a notion of slotted time and scheduled read operations similar to STDMA (Spatial Time Division Multiple Access) protocols [23] for collision resolution. However, due to the different nature of collisions, the traditional STDMA protocols are insufficient in our context.

To determine reading schedules, we take advantage of the fact that in multi-reader deployments, RFID readers are *static* and often carefully deployed in a planned fashion. They also typically have a wired backhaul which can be used for time synchronization. Planned deployment makes it possible to perform RF site surveys to measure the readers' locations and their interference patterns that are inputs to the scheduling algorithms developed here. The algorithms are centralized and offline, and need to run only once after the survey. Thus, their run-time is not a critical factor so long as they are reasonable. Like many STDMA scheduling problems in wireless networks, we will show that the scheduling in the RFID context is also NP-hard; thus, approximation algorithms are desired.

In this paper, we consider two cases viz., when the tag distribution is not known and when the tag distribution is known. For each case, we will address both single channel and multi-channel scheduling algorithms for multiple RFID readers. For single channel cases, we are able to develop approximation algorithms with logarithmic approximation factors, while for multiple channel cases, we develop only heuristics. Our approximation proofs are based on the assumption that the size

*Preliminary version of the paper appeared in *Proceedings of the International Conference on Network Protocols (ICNP), 2007.*

of the “time slot” chosen is large enough to allow each active reader to “read” a tag (see Section IV). We evaluate all solutions via extensive simulations. A key advantage of our approach is that the scheduling works as an overlay on the link-layer. Existing link-layers used in single reader context can still be used with our algorithms.

Paper Organization. The rest of the paper is organized as follows. In Section II, we provide some background on RFID systems, describe our problem, and in Section III, we discuss related work. In the following two sections, we develop algorithms for the appropriately defined Minimum Covering Schedule and Minimum Reading Schedule problems. We present our simulation results in Section VII, and concluding remarks in Section VIII.

II. Background on RFID Systems

Interrogation and Interference Regions. Each RFID reader is associated with a three-dimension *interrogation region* and a three-dimensional *interference region*. The *interrogation region* is the region around a reader where a tag can be successfully read in the absence of any collisions. The *interference region* is the region around a reader where the signal from the reader reaches with sufficient intensity so as to interfere with a tag response. *No relationship between these regions is assumed. We also do not make any assumptions about the shapes of these regions.* However, these regions must be known. This can be done by a RF site survey using a localization device and radio signal strength measurement device. We assume that the RFID reader deployment is planned so that such surveys are practical.

Given a set of readers, we use the term *region monitored* by the readers to mean the union of the interrogation regions of the readers. We also assume that depending on the application and environment, there may be multiple orthogonal channels available to a reader for communication.

Collisions in Multi-Reader Systems. Simultaneous transmissions in RFID systems lead to collisions. In particular, there are three types of collisions.

1) *Tag-tag collision:* This occurs when multiple tags are present in the interrogation region of a reader and transmit IDs at the same time. See Figure 1(a). To schedule the tag responses in a collision-free manner, we need an appropriate link-layer protocol such as framed Aloha [21] or tree-splitting [14], [17]. We describe these protocols in Section III.
2) *Reader-tag collision:* This happens when a reader is in the interference region of another reader. In Figure 1(b), interference from A can “drown” the signal from tag x targeted for B. Reader-tag collision can be avoided by assigning different channels to near-by readers [7], or by scheduling the near-by readers to be active at different times.
3) *Reader-reader collision:* This happens when two readers with overlapping interrogation regions are active at the same time. In such a case, the tags in the overlapped region can not differentiate between the two signals from the two readers. See Figure 1(c). Interestingly, this collision cannot be avoided by operating the readers in different channels. The only way to avoid this collision is to not activate the interfering readers at the same time.

In this paper, we focus on alleviating reader-tag and reader-reader collision problems in a multiple-reader environment by using an STDMA style single-channel or multi-channel scheduling. The basic idea is to use synchronized slots on the readers and activate appropriate readers in appropriate channels in appropriate time slots. The tag-tag collisions are resolved using an independent link layer protocol (such as framed-Aloha based [21] or a tree-splitting protocol [17]). Thus, no fundamental change in the link layer is needed.

III. Related Work

Recently, several approaches have appeared in literature to avoid collisions in RFID systems. Below, we classify them into two groups depending on the type of collisions they address.

Avoiding Tag-Tag Collisions. Recently, several papers [5], [14], [17], [21] have designed link layer protocols to avoid tag-tag collisions. In particular, [14], [17] propose a *tree-splitting* protocol, where the reader organizes the entire ID space of tags into a binary *tag tree* with each tag ID mapped to a leaf. The reader then traverses the tree in a depth-first order. At each tree node, it broadcasts a query message with the bit string corresponding to the tag tree node. A tag, on receiving a query message, responds iff the bit string in the message is the prefix of its own ID. If multiple tags respond, the response messages collide and the reader continues with the depth-first traversal of the tree. No collisions at an interior node u means that there are no more tags remaining in the subtree rooted at u, and thus, the subtree is not traversed further. In a recent work, [18] proposes optimizations to tree traversal.

In Framed Aloha [21] (based on slotted Aloha protocol [3]), a query frame is chosen with a sufficiently large number of subframes[1] and each tag chooses a random subframe to send a response. The reader sends confirmation when it hears a tag response correctly. If collision happens, the colliding tags must choose another random subframe to send a response. The reader adjusts the frame size (number of subframes) according to the number of collisions detected in the previous frame.

Avoiding Reader-Reader or Reader-Tag Collisions. Colorwave [22] is the one of the first works to address reader-reader collisions. It only considers a single available channel. In particular, it tries to randomly color the readers such that each pair of interfering readers have different colors. If each color represents a time slot, then the above coloring should eliminate reader-reader collisions. If conflicts arise (i.e., two interfering readers pick the same color), only one of them sticks to the chosen color and the other picks another color.

[1]We use the term subframe instead of the original term time-slot to avoid confusion with our own concept of time slots.

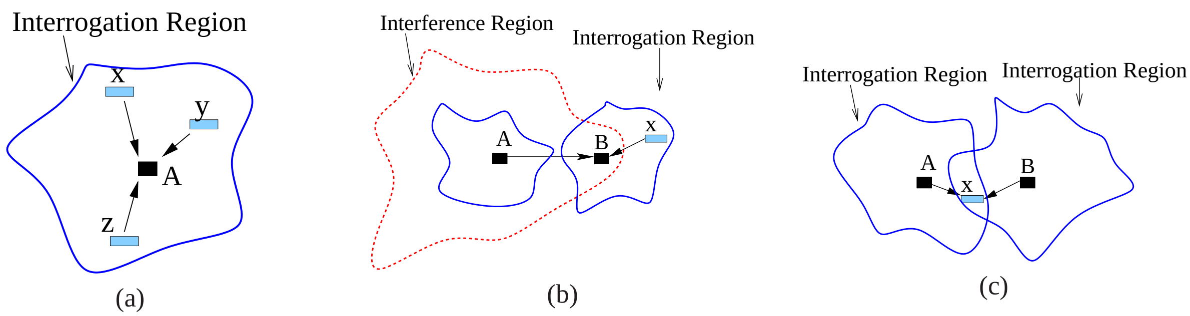

Fig. 1. Collisions in RFID systems. (a) Tag-Tag collision - Tags x, y, and z respond to reader A simultaneously, causing collision at A. (b) Reader-Tag collision – Response from tag x to reader B is "drowned" by the signal from reader A. (c) Reader-Reader collision: Signal/queries from reader A and B collide at tag x.

In [7], the authors suggest coloring of the interference graph (as defined in Definition 6) using c colors, where c is the number of available channels. If the graph is not c-colorable using their suggested heuristic, then the authors suggest removal of certain edges and nodes from the interference graph. This work aims at avoiding the reader-tag collisions exclusively.

In the recent EPCGlobal Gen 2 standard [2], a dense reading mode has been proposed, where the tag responses happen in different channels than the readers. If the number of channels are sufficient, this technique eliminates reader-tag collisions, but requires a relatively sophisticated tag technology.

For a given network of readers and communication pattern, [11] proposes a Q-learning process that yields an optimized resource (channel and time slot) allocation scheme after a training period. The training process determines the channel and time slot to allocate to a reader, when a new read request comes in. The above work considers both reader-reader and reader-tag collisions, but assumes that readers involved in a reader-reader or reader-tag collisions can somehow communicate with each other. Moreover, they assume a fixed number of time slots, and aim at maximizing the frequency and time utilization ratio rather than the more practically important metric of total reading time. Finally, the above work does not provide any performance guarantee.

IV. **Problem Formulation**

We develop algorithms for two key scenarios – when the spatial distribution of tags is unknown, and when it is known. *The spatial distribution of tags plays a critical role in the algorithm because of our reliance on common link layer protocols, wherein time required to read tags is proportional to the number of tags to be read [17], [21].* Thus, without the knowledge of tag distribution, the relative importance of the various "subregions" cannot be estimated, i.e., how long should each subregion be covered/read by a reader. The above is true even if the total number of tags can be estimated [16]. Thus, in the context of unknown distributions, we consider the "minimum covering schedule" problem of computing the smallest slotted-schedule of readers such that the computed schedule "covers" the entire given region. To read all the given tags in the region, such a designed schedule is repeated iteratively until all tags are read. If tag distributions vary widely, then the above strategy (of iterating over a covering schedule) may be inefficient, since in the later iterations some of the readers may not have any tags to read. However, when tag distribution is unknown, any scheduling algorithm will suffer from the same issue. On the other hand, with the knowledge of tag distribution, such inefficiencies can be alleviated.

In this section, we formally define the minimum covering schedule (MCS) problem for the "unknown tag distribution" scenario. The corresponding problem in the "known tag distribution" scenario will be formulated and addressed in Section VI. Before we formally define the MCS problem, we discuss the concept of time slots and give a few definitions.

Time Slots. As noted before we are using a slotted time model. In each time slot, each reader is either *active* or *inactive*. In addition, in a time slot, each active reader operates on an appropriately chosen channel, and tries (not necessarily with success) to read the tags in its interrogation region. The size of the time slot is chosen to be sufficiently large so that each active reader A is able to read at least one tag within the time slot, as long as there are some tags that can possibly be read (i.e., well-covered tags, as defined below) by the reader A. In other words, the time slot is chosen large enough to be able to mitigate tag-tag collissions to the extent that one tag can be read by each reader.

In the context of the tree-splitting algorithm [17], the time slot size can correspond to the time required to traverse a certain number of tree edges such that one tag is read. In the case of Aloha protocol, the time slot size corresponds to the size of the query frame that will allow at least one subframe to be free of tag-tag collissions. Thus, in the case of underlying link-layer protocol being tree-splitting algorithm, the time slot size depends only on the number and distribution of tag IDs around the readers, while in the case of Aloha protocol the time slot size depends on the number of tags around the readers. We note that if we chose a very small time slot then in the worst case our solutions may result in no tags ever been read; thus, we chose a larger than sufficient time slot, to be safer. In Section VII, we conduct a small empirical study to determine the "optimal" size of a time slot, and the discuss the associated trade-off.

Definitions. We now give two definitions that will aid in formally defining the MCS problem. First, we define when a tag is considered "readable" by a reader. Then, we define the concept of a covering schedule of readers. Informally, our MCS problem is to determine the shortest covering schedule of readers for a given set of reader locations and channels, in a *centralized* and *offline* manner.

Definition 1: (**Well-Covered Tag/Location.**) A tag G or its location is said to be *well-covered* by a reader A in a time slot, wherein $\mathcal{R}$ is the set of active readers, if the below conditions hold.

- The reader A is in $\mathcal{R}$, and the tag G is in the interrogation region of A.
- The reader A is not in the interference region of any other reader $A' \in \mathcal{R}$ such that A' is operating on the same channel as A in the given time slot. This condition ensures that there are no reader-tag collisions.
- There is no other reader A' in $\mathcal{R}$ such that the tag G is in the interrogation region of A'; the reader A' may be operating on any channel. This condition ensures that there are no reader-reader collisions.

Due to the first and the last condition, a tag can be well-covered by at most one reader in any time slot. □

Definition 2: (**Covering Schedule of Readers**.) Consider a set of readers $\mathcal{R}$ and a set of available channels F. Let $\mathcal{M}$ be the region monitored by $\mathcal{R}$ (i.e., the union of their interrogation regions), and τ (number of time slots) be some positive integer. A *covering schedule of readers* for $\mathcal{R}$ is an assignment $\Psi : (\mathcal{R} \times \{1, 2, \ldots, \tau\}) \rightarrow (F \cup \{\text{Inactive}\})$ of readers to channels (or being inactive) in each time slot, such that each location in $\mathcal{M}$ is well-covered by some reader in one of the time slots. Here, τ is called the *size* of the covering schedule of readers. □

Use of Covering Schedule of Readers to Read Tags. As mentioned before, the time slot size is chosen such that each active reader A is able to read at least one tag within the time slot, if there is at least one tag well-covered by A. Thus, if we iterate over a covering schedule of readers, then we are guaranteed to read *any* distribution of tags in the region monitored by the given readers. This is easily achieved by rendering a tag passive (using a lower layer protocol) when it is read; thus, an already read tag does not participate in later iterations. The number of iterations required to read all the tags is equal to the maximum number of tags well-covered by a reader in any time slot of the given covering schedule. We now formally define the MCS problem for the case of unknown distribution of tags.

Minimum Covering Schedule (MCS) Problem. Given a set of readers $\mathcal{R}$ (with locations and associated regions) and a set of channels F, the *Minimum Covering Schedule (MCS) Problem* is to find the minimum-size covering schedule of readers for $\mathcal{R}$.

The above defined MCS problem is NP-hard, since it reduces to set-cover for the special case of single channel and very large interference regions. We note that most geometric versions of set-cover remain NP-hard [4], [12].

V. Minimum Covering Schedule (MCS) Problem

In this section, we develop algorithms for solving the Minimum Covering Schedule (MCS) problem for both single and multiple channel settings, when the spatial distribution of tags is not known a priori. Before developing the formalisms, we first informally describe our approach for the single channel; generalization to multiple channels is relatively straightforward.

Basic Idea of the Greedy Approach. The basic idea is to use a greedy algorithm to activate a set of non-interfering readers in each time slot such that a maximum possible amount of "new" area is covered in each slot. The new area means the area not covered in a prior slot. The area here is measured in terms of the number of atomic subregions (called subelements) formed by the intersection of interrogation regions of the readers. Thus, for each time slot, the problem boils down to choosing an independent set in the "interference graph" of readers that covers the maximum number of new subelements. This "weighted" independent set problem being NP-hard, we develop an approximation algorithm. In essence, our overall greedy algorithm for MCS uses this approximation algorithm as a subroutine.

The greedy algorithm for the single channel case is called GA-1. The weighted independent set problem is called DWIS (dynamic weighted independent set). The word "dynamic" is added to signify that the weights for readers are not constant; the weights change from slot to slot as more and more subelements are covered. Finally, the approximation algorithm for DWIS is called DWIS-PTAS as it uses a polynomial-time approximation scheme (PTAS).

Definitions. Now, we define the concepts of subelement, coverage, and interference graph for more formal treatment of the above described greedy algorithm. Informally, a subelement is an atomic subregion in the intersection of interrogation regions; a subelement is defined to be unread in a time slot if it hasn't been covered before; weight of a set of readers $\mathcal{A}$ is the number of unread subelements well-covered by $\mathcal{A}$. Finally, independent set of readers is defined as a set of readers that do not interfere with each other.

Definition 3: (Subelement; Well-Covered Subelement.) A *subelement* is a geographic region. Two points belong to same subelement if and only if they belong to the interrogation regions of the same set of readers. See Figure 2, where there are 13 subelements corresponding to 4 readers and their interrogation regions $R1$ to $R4$.

A subelement s is said to be *well-covered* by a set of readers $\mathcal{A}$ *in presence* of a set of active readers $\mathcal{A}_1$ ($\supseteq \mathcal{A}$) if some[2] location in s is well-covered by some reader in $\mathcal{A}$ (based on Definition 1) when the set of active readers is $\mathcal{A}_1$. Note that

[2]Note that if some point in s is well-covered by a reader B, then all the points in s are well-covered by B.

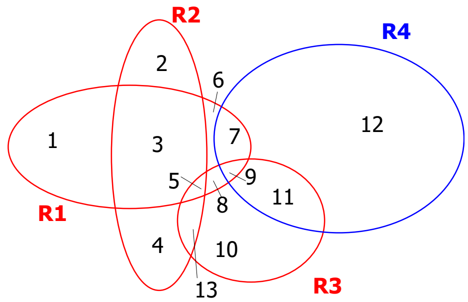

Fig. 2. Illustrating the concept of subelements.

whether a subelement is well-covered by $\mathcal{A}$ or not depends on the given set $\mathcal{A}_1$ of active readers. □

Definition 4: (Unread Subelement.) A subelement s is considered *unread* at a given time slot if some location in s has not been well-covered by any reader in any of the *previous* time slots. □

Note that the MCS problem is essentially to "read/cover" all the subelements using a minimum-size schedule of readers.

Definition 5: (Weight of Readers.) The *weight* of a set of readers $\mathcal{A}$ in the given time slot is denoted by $w(\mathcal{A})$, and is defined as the number of unread subelements in the given time slot that are well-covered by $\mathcal{A}$ in presence of $\mathcal{A}$. Above, each reader in $\mathcal{A}$ is associated with a channel (which will be either stated or evident from the context).

For clarity, we use $w(A)$ for $w(\{A\})$ where A is a reader. Note that $w(\mathcal{A}_1 \cup \mathcal{A}_2)$ may be less than $w(\mathcal{A}_1) + w(\mathcal{A}_2)$ (due to the collisions). □

Definition 6: (Interference Graph; Independent Set of Readers.) The *interference graph* is an undirected[3] graph over the set of readers in the system such that an edge (A, A') exists in the interference graph if A lies in the interference region of A' or vice versa. An edge (A, A') in the interference graph signifies that A and A' will incur a reader-tag collision if they are active on the same channel in the same time slot.

A set of readers is called *independent* if it forms an independent set of vertices in the interference graph. □

Remark on Interference Graph. Note that the above interference graph is defined based on only the interference regions. Essentially, our strategy is to *completely* avoid reader-tag collisions by picking an independent set (as defined above) of readers in each time slot. This makes sense since reader-tag collisions between two readers renders at least one of the readers completely useless (incapable of reading any tags based on Definition 1). On the other hand, reader-reader collisions between two readers only disallow certain tags (in the intersection of the interrogation regions) to be well-covered by any reader. Thus, we *minimize* (rather than eliminate) reader-reader collisions by picking an independent set of reader of near-maximum weight to activate in each time slot.

[3]Even though the interference between two readers may be directed (due to different interference ranges), it is sufficient to consider an undirected graph for the purposes of computing an independent set since presence of an edge (A, A') (whether directed or undirected) must only serve the purpose of preventing A and A' to be in an independent set together.

A. Single Channel Setting

We now formally address the MCS problem for the single channel, and present a greedy algorithm (GA-1). Recall that GA-1 uses (as a subroutine) the DWIS-PTAS algorithm for an appropriately defined DWIS problem. We start by describing the greedy algorithm. Then, we define the DWIS problem, describe the DWIS-PTAS algorithm (an approximation algorithm for the DWIS problem), and prove the approximation bound of the DWIS-PTAS algorithm using a few lemmas. Finally, we prove the approximation bound of GA-1, the greedy algorithm for the MCS problem.

Greedy Algorithm (GA-1). The Greedy Algorithm (GA-1) algorithm for the single channel MCS problem works in steps.

- In the q^{th} step, the DWIS-PTAS algorithm (described below) is used to select an independent set of readers with near-maximum weight.
- The selected set of readers are to be activated in the q^{th} time slot with the same available channel.
- GA-1 terminates when there are no more unread subelements.

Note that the algorithm is run statically (for a given set of static reader) to determine the schedule. This needs to be done only once. For actual reading of tags, the readers are simply activated according to the computed schedule. We will now show that the above GA-1 algorithm delivers a near-optimal schedule of readers. We first formally state the DWIS problem of selecting an independent set of readers with maximum weight, and then, present the DWIS-PTAS algorithm.

Dynamic-Weighted Independent Set (DWIS) Problem. Let G be the interference graph of the given set of readers. Let each reader/vertex A in G be associated with $w(A)$, the weight of A in the *given* time slot. The *DWIS problem* is to select a maximum weighted independent set in the interference graph. The DWIS problem is NP-hard since its special case corresponding to null interrogation regions and uniform weights is equivalent to the NP-hard problem of unweighted independent set in unit-disk graphs [13].

Below, we present DWIS-PTAS, a polynomial-time approximation scheme (PTAS) for the DWIS problem in two dimensions, and then, generalize it to three dimensions. The below DWIS-PTAS is a generalization of the PTAS for the unweighted independent set problem in unit-disk graphs presented in [13] (which in turn uses the "shifting strategy" introduced by [12]). The main difficulty in generalizing the result of [13] arises due to the fact that in our context $w(\mathcal{A}_1 \cup \mathcal{A}_2)$ may be *less* than $w(\mathcal{A}_1) + w(\mathcal{A}_2)$ for two sets of readers $\mathcal{A}_1$ and $\mathcal{A}_2$. Note that we do not make the unit-disk assumption; however, the time-complexity of our algorithms depends on T, S (as defined below), and the lower bound on the area of the interrogation region (see Equation 1 and the following discussion).

Definition 7: (Interference Reach (T); Interrogation Reach (S).) Let T be such that interference region of each reader is *contained* in a sphere or disk of radius T. Similarly, let S be such that the interrogation region of each reader is contained in a sphere or disk of radius S. We refer to T and S as *interference* and *interrogation reach* respectively. *Note that T and S values are bounded, due to the bounded reader's transmission power or tag's limited power/circuitry.* □

DWIS-PTAS (in two-dimensions). Consider an interference graph G with associated weights as defined above. The DWIS-PTAS algorithm consists of the following steps. Let k be a given positive integer (higher k entails higher time-complexity, but better approximation ratio).

- Divide the whole rectangular region[4] into horizontal strips of width $\max(T, 2S)$. Note that if two readers A_1 and A_2 are at least $\max(T, 2S)$ distance away, then (i) they do not interfere, and (ii) $w(\{A_1, A_2\}) = w(A_1) + w(A_2)$.
- For each i, $0 \le i \le k$, partition the graph G into l disjoint subgraphs $G_{i1}, G_{i2}, \ldots, G_{il}$ by removing nodes in horizontal strips congruent to $i \mod (k+1)$. See Figure 3.
- Find a near-optimal independent set in each subgraph G_{ip}. Based on Lemma 2 (described later), we can actually find an independent set of weight at least $\frac{k}{k+1}$ times the optimal weight in polynomial time.
- For each i, take the union of the independent sets of G_{ip} ($1 \le p \le l$). Since the width of the horizontal strip is at least $\max(T, 2S)$, the union forms an independent set in

$$G_i = \bigcup_{1 \le p \le l} G_{ip}$$

  and the weight of the independent set in G_i is the sum of the weights of the independent sets of G_{ip}.
- Pick the best (maximum weighted) of the independent sets of G_i's as the independent set of G.

Lemma 1 shows that an optimal independent set of one of the subgraphs G_i has a weight of at least $\frac{k}{k+1}$ times the maximum weight of an independent set in G. Thus, by Lemma 1 and 2, we have that the above described DWIS-PTAS yields a $(\frac{k}{k+1})^2$-approximate independent set for any given integer k. This constitutes Theorem 1. We now develop these lemmas/theorems to prove the approximation ratio of DWIS-PTAS.

Lemma 1: Let the maximum weight of an independent set in G_i be W_i and in G be W. Then,

$$\max_{0 \le i \le k} W_i \ge \frac{k}{k+1} W.$$

PROOF. Let O be the optimal solution of DWIS problem, i.e., the maximum-weight independent set in G. Let

$$O_i = O \cap (G - G_i),$$

[4]This rectangular region, which includes the interrogation regions of all the given readers, can be arbitrarily large since the time complexity of our algorithm does not depend on the region's size.

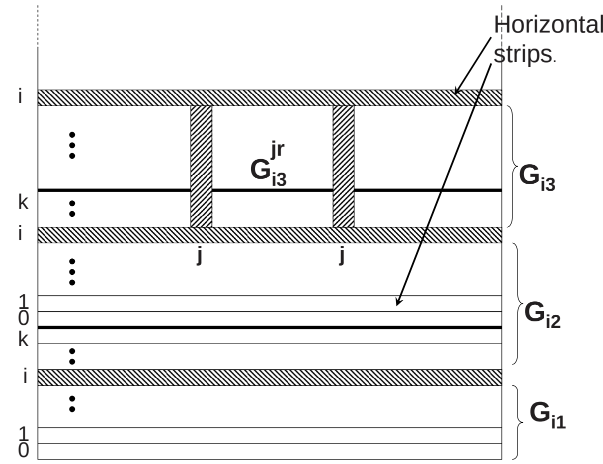

Fig. 3. Division of graph G into subgraphs G_{ip}: First, the whole region is divided into horizontal strips, which are numbered iteratively from 0 to k as shown above. Then, for each i ($0 \le i \le k$), strips numbered i (shaded in the figure) are removed to yield subgraphs $G_{i1}, G_{i2}, \ldots, G_{il}$ for some finite l. Similarly, each G_{ip} is vertically partitioned into G_{ip}^{jr} (for use in Lemma 2).

i.e., O_i is the set of nodes from the optimal solution O in the shaded horizontal strips of Figure 3. Thus, $O = \bigcup_{0 \le i \le k} O_i$.

For any $U \subseteq O$, let $e(U)$ denote the number of unread subelements that are well-covered by U in presence of O. In other words, $e(U)$ is $w(U)$ minus the number of subelements that are contained in the region monitored by U as well as $O - U$ (and hence, not well-covered by U in presence of O, due to reader-reader collisions). Thus, we have

$$e(U) \le w(U).$$

Also, since $O = \bigcup_i (O_i)$, we have $w(O) = \sum_{0 \le i \le k} e(O_i)$. Thus, there exists a t, $1 \le t \le k$, such that $e(O_t) \le \frac{1}{k+1} w(O)$. Now, since $O = O_t \cup (O \cap G_t)$, we have $w(O) = e(O_t) + e(O \cap G_t)$ and thus,

$$e(O \cap G_t) \ge \frac{k}{k+1} w(O) = \frac{k}{k+1} W.$$

For the rest of the proof, note that

$$\max_{0 \le i \le k} W_i \ge W_t \ge w(O \cap G_t) \ge e(O \cap G_t) \ge \frac{k}{k+1} W.$$

■

For clarity of presentation, let us use β to denote the upper bound on the size of an independent set of readers in a square of size $\max(T, 2S) \times \max(T, 2S)$. If θ is the minimum area of an interference region, then

$$\beta = (\max(T, 2S))^2 / \theta. \quad (1)$$

Note that β is bounded by a constant, since each reader must have a non-empty interference region (and thus, θ is bounded from below). We now show the approximation ratio of the DWIS-PTAS algorithm.

Lemma 2: Consider a subgraph G_{ip} (as defined above) where $1 \leq p \leq l$ and $1 \leq i \leq k$. In $|G_{ip}|^{O(k^2\beta)}$ time, we can construct an independent set in G_{ip} whose weight is at least $\frac{k}{k+1}$ times the optimal.

PROOF. We construct subgraphs G_{ip}^{jr} in G_{ip} for $1 \leq j \leq k$ and $1 \leq r \leq l_p$ (for some l_p) by vertical division of G_{ip}, just as G was divided horizontally into subgraphs G_{ip}. See Figure 3. Using a simple packing argument, we can see that the maximum size of an independent set in G_{ip}^{jr} is at most $O(k^2\beta)$. Thus, we can compute the maximum independent set in G_{ip}^{jr} by exhaustive search, and take a union over all r to yield a maximum independent set in $G_{ip}^{j} = \bigcup_r G_{ip}^{jr}$. Then, we pick the best independent set among G_{ip}^{j} over all j, which gives a $\frac{k}{k+1}$-approximate independent set for G_{ip} (based on arguments similar to Lemma 1). ∎

The proof of the below theorem follows from the above two lemmas.

Theorem 1: The DWIS-PTAS algorithm runs in $|\mathcal{R}|^{O(k^2\beta)}$ time and returns an independent set whose weight of at least $(\frac{k}{k+1})^2$ times the optimal.

PROOF. The proof follows from the above two lemmas, viz., Lemma 1 and 2, and the fact that the optimal independent set for any G_i is the union of the optimal independent sets for G_{ip} (for all p). The above fact is true due to chosen width $(\max(T, 2S))$ of the horizontal strips and the fact that $w(\{A_1, A_2\}) = w(A_1) + w(A_2)$ for two readers A_1 and A_2 that are at least $\max(T, 2S)$ distance away. ∎

IDWIS-PTAS: Improved DWIS-PTAS. As suggested in [13], we can improve the performance of DWIS-PTAS by computing the weighted independent set in G_{ip} *optimally* using a dynamic programming approach. The improved DWIS-PTAS (IDWIS-PTAS) runs in $|\mathcal{R}|^{O(k\beta)}$ time and delivers a solution with an approximation ratio of $(k/k+1)$. We state the below without proof, as it follows directly from a dynamic programming technique similar to the one used in [13].

Theorem 2: The IDWIS-PTAS algorithm runs in $|\mathcal{R}|^{O(k\beta)}$ time and returns an independent set whose weight is at least $\frac{k}{k+1}$ times the optimal. ∎

IDWIS-PTAS in 3D. The above described IDWIS-PTAS can be easily generalized to three dimensions. Essentially, we further divide G_{ip} vertically into G_{ip}^{jr} as shown in Figure 3. Then, using dynamic programming, we can compute the optimal independent set in the hyper-rectangle G_{ip}^{jr} in $|\mathcal{R}|^{O(k^2\beta)}$ time. Here, β is the bound on the maximum size of an independent set in a *cube* of size $\max(T, 2S) \times \max(T, 2S) \times \max(T, 2S)$. Using similar arguments as before, we get the following result.

Theorem 3: In three-dimensions, the IDWIS-PTAS algorithm runs in $|\mathcal{R}|^{O(k^2\beta)}$ time and returns an independent set whose weight of at least $(\frac{k}{k+1})^2$ times the optimal weight for any positive integer k. ∎

Performance of GA-1 for the MCS Problem. Recall that in q^{th} step of the GA-1 algorithm, we use the IDWIS-PTAS to select a set of readers to activate in the q^{th} time slot. For a given $\epsilon > 0$, if we choose k as the smallest integer that satisfies

$$\left(\frac{k+1}{k}\right)^2 \leq (1+\epsilon), \tag{2}$$

we have the following result.

Theorem 4: Given set of readers $\mathcal{R}$ in three-dimensions, GA-1 returns a covering schedule of readers $\mathcal{R}$ of size at most $2(1+\epsilon)\ln|\mathcal{R}|$ times the optimal size, for any $\epsilon > 0$. Moreover, GA-1 runs in $|\mathcal{R}|^{O(\beta/\epsilon)}$ time.

PROOF. Since GA-1 iterates until there are no unread subelements, any location in the monitored region is indeed well-covered by an active reader in one of the time slots of the GA-1 solution. Thus, GA-1 returns a covering schedule of readers. Time complexity of GA-1 follows from Theorem 3 and choice of k. We now show the approximation result.

Let $\mathcal{A}_q$ and $\mathcal{O}_q$ be the set of readers selected to be active in the q^{th} time slot by GA-1 and optimal algorithm respectively. Let $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_Q\}$ and $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_P\}$ represent the solution returned by GA-1 and optimal algorithm respectively, where Q and P are the number of time slots used by GA-1 and optimal algorithm respectively. We will show that $Q \leq 2(1+\epsilon)(\ln|\mathcal{R}|)P$.

Let us consider the q^{th} step of GA-1, wherein readers in $\mathcal{A}_q$ are selected to be active in the q^{th} time slot. At each step, we distribute the cost of one (time slot) to all the unread subelements that are well-covered by $\mathcal{A}_q$ in presence of $\mathcal{A}_q$ in the q^{th} time slot. Let c_s denote the cost distributed to the subelement s when its read. If s is unread at q^{th} time slot and is well-covered by $\mathcal{A}_q$ (in presence of $\mathcal{A}_q$), then $c_s = \frac{1}{U_q - U_{q-1}}$, where U_q is the number of unread subelements at the end of (after the) q^{th} time slot of GA-1.

Let $\mathcal{S}$ be the set of all subelements, and $E(\mathcal{O}_p)$ denote the set of subelements in $\mathcal{S}$ that are well-covered by the set of readers $\mathcal{O}_p$ in presence of $\mathcal{O}_p$. Now, since the optimal solution has to read all subelements, we have

$$Q = \sum_{s \in \mathcal{S}} c_s \leq \sum_{\mathcal{O}_p \in \mathcal{O}} \sum_{s \in E(\mathcal{O}_p)} c_s. \tag{3}$$

In the next paragraph, we will show that for any $\mathcal{O}_p \in \mathcal{O}$,

$$\sum_{s \in E(\mathcal{O}_p)} c_s \leq 2(1+\epsilon)\ln|\mathcal{R}|. \tag{4}$$

Now, from Equation 3 and 4, we get $Q \leq 2(1+\epsilon)(\ln|\mathcal{R}|)P$.

Proving Equation 4. Let u_q denote the number of unread subelements in $E(\mathcal{O}_p)$ after the q^{th} time slot of GA-1. Without loss of generality, we can assume that $\mathcal{O}_p$ is an independent set of readers (else, some readers in $\mathcal{O}_p$ would be redundant, as there is only a single channel available). Note that u_0 is the total number of subelements in $E(\mathcal{O}_p)$. Thus,

$$\sum_{s \in E(\mathcal{O}_p)} c_s = \sum_{q=1}^{Q} (u_{q-1} - u_q) \cdot \frac{1}{U_q - U_{q-1}}$$

By Theorem 3 and choice of k, we know that the total weight of $\mathcal{A}_q$ $(= U_q - U_{q-1})$ is at least $(\frac{1}{1+\epsilon})u_{q-1}$, since $\mathcal{O}_p$ is also an independent set of readers with weight at least u_{q-1} in the q^{th} time slot. Thus, we have

$$\sum_{s \in E(\mathcal{O}_p)} c_s \leq (1+\epsilon) \sum_{q=1}^{Q} (u_{q-1} - u_q) \cdot \frac{1}{u_{q-1}}$$

Using some algebra ([6], Chapter 35.3), we get

$$\sum_{s \in E(\mathcal{O}_p)} c_s \leq (1+\epsilon) \ln u_0.$$

Since $u_0 = |E(\mathcal{O}_p)| \leq |\mathcal{R}|^2$, we get

$$\sum_{s \in E(\mathcal{O}_p)} c_s \leq 2(1+\epsilon) \ln |\mathcal{R}|.$$

■

B. Multiple Channels Setting

In this subsection, we consider the MCS problem when there are multiple available channels in the system. For example, in the EPCGlobal Gen2 standard [2], there are about 50 available channels. However, unlike in previous cases, algorithms developed here are heuristics without any performance guarantees. We evaluate the empirical performance of the developed heuristics in Section VII. Note that the MCS problem for the case of multiple channels is a generalization of the single channel case, and hence, is trivially NP-hard.

GA-M: Greedy Algorithm For Multiple Channels. For the case of multiple available channels, we design a greedy algorithm (GA-M) that works as follows. GA-M iterates through time slots, and for each slot, it selects a set of active readers with appropriately chosen associated channel for each reader, such that the set of active readers operating on the same channel form an independent set in the interference graph. The readers with their associated channels are chosen in a greedy manner for each time slot as follows. Consider the q^{th} time slot. We maintain a set RC of reader-channel pairs, such that a pair $(A, c) \in RC$ implies that the reader r has been selected to be active with channel c in the q^{th} time slot. Initially, RC is empty. Then, we iteratively pick the "best" reader-channel (A, c) pair to add to RC. The best reader-channel pair for a given RC is defined as a pair (A, c) that maximizes the total number of unread subelements well-covered by $(RC \cup \{(A, c)\})$ (in presence of $(RC \cup \{(A, c)\})$) in the q^{th} time slot. The above process in continued until no more tags can be read in the q^{th} time slot. At that point, GA-M finalizes RC as the set of reader-channel pairs for the q^{th} time slot, and starts the above process again for the next time slot. GA-M is formally presented below.

Algorithm 1: GA-M: Greedy Algo. for Multiple Channels.
Input:: A set of readers $\mathcal{R}$ and F, the set of available channels.
Output:: Solution to the MCS problem.
BEGIN

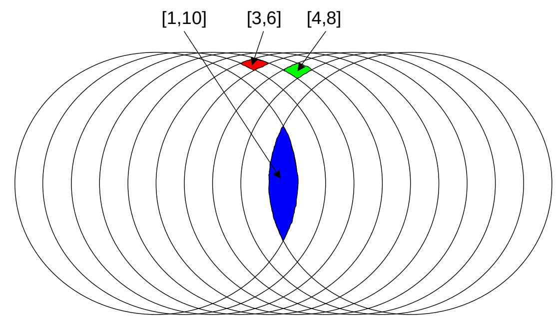

Fig. 4. A set of n readers with interrogation regions, where the MCS problem requires at least $O(\log n)$ time slots.

q= 1; /∗ Time slot number. ∗/
while (there are unread subelements)
$RC = \phi$; /∗ Set of active reader with channels in q^{th} slot. ∗/
while (1)
Let $N(RC)$ be the number of unread subelements well-covered by RC in presence of RC in the q^{th} slot.
if (there is (A, c) s.t. $N(RC \cup \{(A, c)\} > N(RC)$
then
Pick the reader-channel (A, c) pair that maximizes $N(RC \cup \{(A, c)\}$
$RC = RC \cup \{(A, c)\}$;
else
Pick RC for q^{th}time slot.
q++;
break;
end if;
end while;
end while;
END. ◇

Unlimited Number of Channels. When there are unlimited number of available channels, the MCS problem is similar to the conflict-free coloring problem [10]. Given a set of regions in a 2D plane, the *conflict-free coloring problem* is to color the regions in a conflict-free manner using minimum number of colors, where a coloring is said to be *conflict-free* if for every point p there is a region containing p whose color is unique among all the regions that contain p. The above problem is NP-hard even for unit disk regions [8]. However, n pseudo-disks (boundaries intersect at most twice) can be conflict-free colored in $O(n \log n)$ time using $O(\log n)$ colors [10]. Also, there are instances of n unit-disk regions whose conflict-free coloring requires at least $O(\log n)$ colors. If each color is looked upon as a time slot, then the MCS problem is almost equivalent to the conflict-free coloring of readers, except that in the former each reader can be assigned to multiple colors/slots. The above observation yields the following theorem.

Theorem 5: Consider the MCS problem for a given set of n readers with unlimited number of channels available. Assume that the interrogation regions of the readers are two-dimensional pseudo-disks, i.e., regions such that boundaries of each pair of them intersect at most twice. Then, a solution of the MCS problem can be constructed in $O(n \log n)$-time using $O(\log n)$ time slots. Also, there are instances of n readers wherein at least $O(\log n)$ time slots are required.

PROOF. As mentioned above, the MCS problem reduces to the problem of conflict-free coloring of the interrogation regions of the readers, and hence, can be solved using $O(\log n)$ time slots in $O(n \log n)$ time [10].

To show the lower bound [19], we use the same argument as in [8] where each region is assigned exactly one color. Consider the set of readers numbered 1 to n with unit-disk interrogation regions (also numbered 1 to n) as shown in Figure 4. Here, for each interval $[i, j]$ where $i \leq j$, there is a unique subelement s such that the set of interrogation regions containing s is exactly $\{i, i+1, \ldots, j\}$. Now, to read the subelement $[1, n]$, there must be one time slot (say, 1^{st}) wherein only one reader (say j) is active (irrespective of the channel assigned). The above is needed to avoid reader-reader collisions. Without loss of generality, let $j > n/2$. Now, consider the subelement $[1, n/2]$. Using the same argument, there must be a time slot other than the first (say, 2^{nd}) wherein only one reader from $\{1, 2, \ldots, n/2\}$ is active. Note that the above argument holds even if the reader j is allowed to be chosen in multiple time slots. Continuing the above argument, we can see that at least $(\log n)$ time slots are required to read all the subelements in Figure 4. ∎

Three-dimensional Regions. Very little is known about the problem of conflict-free coloring of three-dimensional regions [19], [20]. Note that with unlimited number of available channels, the interference graph has no edges and hence, any subset of readers forms an independent set. Thus, β (the bound on the size of an independent set of readers in a bounded cube) is no longer a constant. But, if we choose a small k and assume that the number of readers in any hyper-rectangle of size $k \max(T, 2S) \times k \max(T, 2S) \times \max(T, 2S)$ is small (or a constant), then we can use exhaustive search to compute the maximum-weighted set in such hyper-rectangles. We can then apply the same arguments as in Section V-A to obtain a $2(1 + \epsilon) \ln |\mathcal{R}|$-approximate solution.

Theorem 6: For the case of unlimited number of channels in three-dimensions, the modified GA-1 (as described above) returns a $2(1 + \epsilon) \ln |\mathcal{R}|$-approximate solution and runs in time polynomial in $|\mathcal{R}|$ and 2^N for any $\epsilon > 0$. Here, N is the maximum number of readers in any hypercube of size $k \max(T, 2S) \times k \max(T, 2S) \times \max(T, 2S)$ and k is as in Equation 2. ∎

VI. Minimum Reading Schedule (MRS) Problem

So far, we considered the scenario where the spatial distribution of tags was unknown. Recall that in this case only the Minimum Covering Schedule problem made sense. This is because it was not possible to learn how much time to allow for various subelements to be read, as time to read all tags in a subelement is proportional to the number of tags in that subelement. However, when the tag distribution is known, we can model this time easily. Thus, we can consider a more meaningful version of the problem, where we try to read all tags as fast as possible for the given tag distribution. We call this the Minimum Reading Schedule (MRS) problem.

Offline Algorithms. In our problem setting, we are *given* a set of reader locations and spatial distribution of tags, and we want to compute a "schedule of readers" to read the tags as fast as possible. We focus on design of an *offline* algorithm because spatial distribution of unread tags is unavailable[5] even to an online algorithm. Moreover, incurring computation and communication time after each time slot may defeat the whole purpose of fast reading of tags. *Thus, we do not consider online algorithms.* Also, since the readers are static, the readers' schedule is computed only once.

Probabilistic Model for Reading a Tag/Subelement. In our model, in a given time slot, each active reader reads a *random* well-covered unread[6] tag from its interrogation region. The size of the time slot is chosen to be large enough to allow the above to happen. Due to this randomness in reading, we need to first formulate the reading of a tag/subelement in a probablistic way (as done below). Based on the probabilistic reading of a tag, we will formulate and solve the MRS problem.

Let $\mathcal{R}$ be the set of given readers, and $\mathcal{M}$ be region monitored by $\mathcal{R}$. For each subelement s_j, we maintain two values, viz.,

1) $g(s_j)$, the **initial** *number of tags* in s_j. The value $g(s_j)$ is available from the given distribution of tags, and *it remains constant across time slots.*
2) $p(s_j)$, the *probability* that a tag within s_j has not been read (based on a probablistic model described below) in the previous time slots. The probability $p(s_j)$ is same for all the tags in a subelement.

Initially (in the first time slot), the probability $p(s_j)$ is 1 for each subelement s_j. Now, consider the q^{th} time slot, and let $p(s_j)$ represent the probability of a tag in s_j not been read in the previous $(q-1)$ time slots. Let A be an active reader in the q^{th} time slot, and let $s_1, s_2, \ldots, s_l$ be the "not-fully-read" subelemements (i.e., subelements with at least one unread tag) well-covered by A (in presence of the set of readers active in the given q^{th} time slot). The probability that a tag in s_j has not been read after q time slots is given by:

$$\text{New } p(s_j) = \max(0, p(s_j)(1 - b)), \tag{5}$$

where $b = 1/(g(s_1)p(s_1) + g(s_2)p(s_2) \ldots + g(s_l)p(s_l))$ is the probability of any particular tag (well-covered by A) being read by A in the q^{th} time slot. Note that the above

[5] Note that when a tag is successfully read by a reader A, the reader A still does not know the location of the read tag.

[6] As before, a tag is turned "passive" when it is read. A passive (already read) tag does not respond to any future queries by any reader.

$p(s_j)$ values are based solely on the spatial distribution of readers and subelement, and initial distribution of tags; they are independent of what actually happens within each time slot. Based on the above model, we now define when a subelement is considered fully-read.

Definition 8: (Fully-Read/Not-Fully-Read Subelement.) In a given time slot, a subelement s_j is considered *fully-read* if $p(s_j)$ is zero at the start of the given time slot; otherwise, s_j is considered *not-fully-read* (i.e., if $p(s_j) > 0$ at the start of the given time slot). □

Reading Schedule. Based on the above probabilistic notion of reading a tag, we define a reading schedule of readers.

Definition 9: (Reading Schedule of Readers.) Consider a set of readers $\mathcal{R}$ and a number of tags $|\mathcal{G}|$ distributed uniformly in the region monitored by the readers. Let F be the set of available channels. A *reading schedule of readers* to read all the tags in $\mathcal{G}$ in τ time slots is an assignment $\Psi : (\mathcal{R} \times \{1, 2, \ldots, \tau\}) \rightarrow (F \cup \{\text{Inactive}\})$ of readers to channels (or being inactive) in each time slot, such that all subelements have been fully-read by the end of τ time slots. The number of time slots τ is referred to as the size of the reading schedule of readers. □

Even though our notion of a tag fully-read is probabilistic, a reading schedule of readers is guaranteed to read all tags, under the assumption that each active reader in each time slot of the reading schedule reads at least one well-covered tag. This is true due to the following:

- Initially, the quantity $\sum_j g(s_j)p(s_j)$ is equal to the total number of tags in the system.
- In each time slot, the decrease in the quantity $(\sum_j g(s_j)p(s_j))$ is less than or equal to the number of tags read in that time slot, since each active readers reads at least one tag in each time slot.
- At the end of a reading schedule the quantity $(\sum_j g(s_j)p(s_j))$ has decreased to zero.

Minimum Reading Schedule (MRS) Problem. Given a set of RFID readers $\mathcal{R}$, the number of tags $|\mathcal{G}|$, and the distribution of the tags in the region monitored by $\mathcal{R}$, the *Minimum Reading Schedule Problem* is to find a reading schedule of readers of smallest size. MRS problem is easily NP-hard (reduces to set-cover).

A. Single and Multiple Channels

In this subsection, we first extend the GA-1 algorithm of the previous section (for the MCS problem) to the MRS problem for the case of a single channel. The case of multiple channel is discussed briefly at the end.

EGA-1: Extended GA-1 Algorithm. We use EGA-1 to refer the extended GA-1 algorithm. As in the GA-1 algorithm, the q^{th} step of the EGA-1 algorithm constitutes of selecting a independent set of readers with near-maximum weight to activate in the q^{th} time slot. EGA-1 terminates when all subelements have been fully-read (i.e., the weight of each reader has become zero).

Definition 10: (Weight of Readers (redefined).) Here, we define the *weight* $w(\mathcal{A})$ of a set of readers $\mathcal{A}$ as the reduction in the sum of the $g(s_j)p(s_j)$ of the not-fully-read subelements s_j well-covered by $\mathcal{A}$ in presence of $\mathcal{A}$. □

Observation 1: In a given time slot, the weight of a set of readers $\mathcal{A}$ is equal to the number of readers in $\mathcal{A}$ that well-cover at least one non-fully-read subelement each when the set of active readers is $\mathcal{A}$ in the given time slots. □

The above observation follows from Equation 5. Based on the above observation, we can show that the IDWIS-PTAS algorithm remains a PTAS for the interference graph with the above definition of weight of readers. Essentially, we need to show that Lemma 1 holds for the above definition of weight of readers.

Lemma 3: Using Definition 10 for weight for readers, let the maximum weight of an independent set in G_i be W_i and in G be W. Then,

$$\max_{0 \le i \le k} W_i \ge \frac{k}{k+1} W.$$

PROOF. As in Lemma 1, let O be the optimal solution of DWIS problem, i.e., the maximum-weight independent set in G. Let

$$O_i = OPT \cap (G - G_i),$$

i.e., O_i is the set of nodes from the optimal solution O in the shaded horizontal strips of Figure 3. Thus, $O = \bigcup_{0 \le i \le k} O_i$.

For any $U \subseteq O$, let $e(U)$ denote the number of readers in U that well-cover at least one unread subelement when the set of active readers is O in the given time slot. Thus, by Observation 1, we have

$$e(U) \le w(U).$$

Rest of the proof is same as Lemma 1. ■

Theorem 7: For the Definition 10 of weight of readers, the DWIS-PTAS algorithm runs in $|\mathcal{R}|^{O(k^2\beta)}$ time and returns an independent set whose weight of at least $(\frac{k}{k+1})^2$ times the optimal.

PROOF. The proof follows from the previous Lemma 2 (which is independent of the weight definition) and the above Lemma 3, and the fact that the optimal independent set for any G_i is the union of the optimal independent sets for G_{ip} (for all p). The above fact is true due to chosen width $(\max(T, 2S))$ of the horizontal strips and the fact that $w(\{A_1, A_2\}) = w(A_1) + w(A_2)$ (for Definition 10) for two readers A_1 and A_2 that are at least $\max(T, 2S)$ distance away. ■

As in Section V, the above theorem generalizes to IDWIS-PTAS in three dimensions.

Theorem 8: For the Definition 10 of weight of readers, in three-dimensions, the IDWIS-PTAS algorithm runs in $|\mathcal{R}|^{O(k^2\beta)}$ time and returns an independent set whose weight of at least $(\frac{k}{k+1})^2$ times the optimal weight for any positive integer k. ■

Thus, when we use IDWIS-PTAS in q^{th} step of EGA-1 to select a set of active readers in the q^{th} time slot, the approximation ratio of EGA-1 is preserved, as proved below.

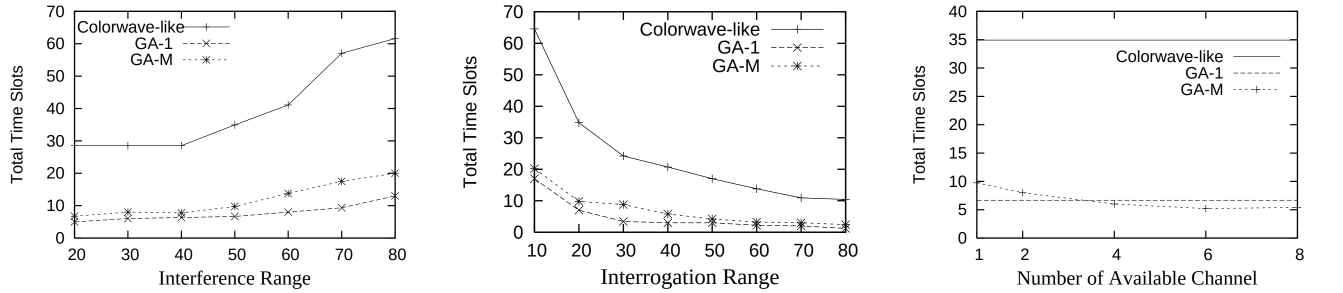

Fig. 5. Performance of the GA-1, GA-M, and Colorwave-like algorithms for the MCS problem. (a) Varying interference range with single channel, (b) Varying interrogation range with single channel, and (c) Varying number of available channels.

Theorem 9: Given a set of readers $\mathcal{R}$ and a distribution of $|\mathcal{G}|$ tags in the (three-dimensional) region monitored by the readers, EGA-1 returns a reading schedule of readers of size at most $(1+\epsilon)\ln|\mathcal{G}|$ times the optimal size, for any $\epsilon > 0$. Moreover, EGA-1 runs in $O(|\mathcal{G}|)|\mathcal{R}|^{O(\beta/\epsilon)}$ time.

PROOF. Since EGA-1 iterates until all subelements are fully-read, EGA-1 indeed returns a reading schedule of readers. We now show the approximation ratio of EGA-1.

Let $\mathcal{A}_q$ and $\mathcal{O}_q$ be the set of readers selected to be active in the q^{th} time slot by EGA-1 and optimal algorithm respectively. Let $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_Q\}$ and $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_P\}$ represent the solution returned by EGA-1 and optimal algorithm respectively, where Q and P are the number of time slots used by EGA-1 and optimal algorithm respectively. We will show that $Q \leq (\frac{k+1}{k})^2(\ln|\mathcal{G}|)P$. Then, we choose k as in Equation 2, we get the theorem result.

By Observation 1, the weight of each $\mathcal{A}_q$ is $|\mathcal{A}_q|$, and each $\mathcal{O}_p$ is $|\mathcal{O}_p|$. Also, note that $\sum_{q=1}^{Q}\mathcal{A}_q = \sum_{p=1}^{P}\mathcal{O}_p = |\mathcal{G}|$, since the initial sum of $p(s_j)g(s_j)$ over all subelements s_j is $|\mathcal{G}|$. In the next paragraph, we will show that

$$w(\mathcal{A}_q) \geq (\frac{k}{k+1})^2(|\mathcal{G}| - \sum_{i}^{q-1} w(\mathcal{A}_i))/P, \qquad (6)$$

i.e., the weight of $\mathcal{A}_q$ is at least equal to $1/P$ of the "remaining weight" ("probabilistic" number of remaining tags) yet to be "covered" by the EGA-1. Since $w(\mathcal{A}_q)$ is at least 1 for all q, the above equation gives $Q \leq (\frac{k+1}{k})^2\log_{(P/P-1)}|\mathcal{G}| \leq (\frac{k+1}{k})^2(\ln|\mathcal{G}|)P$, since $(P/(P-1))^P \geq e$. That last inequality is true as $f(P) = (P/(P-1))^P$ is monotonically decreasing for $P > 1$, and $f(1) = \infty$ and $lim_{P\to\infty}f(P) = e$.

Proving Equation 6. In short, the Equation 6 is true due to the greedy choice of $\mathcal{A}_q$ for each q. At the q^{th} step of EGA-1, the EGA-1 picks a set of readers whose weight is at least $(\frac{k}{k+1})^2$ times the maximum possible at that stage. We show that the maximum possible weight at q^{th} step of EGA-1 is at least $c = (|\mathcal{G}| - \sum_{i}^{q-1} w(\mathcal{A}_i))/P$. To show this, consider the valid schedule of readers represented by $\{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{q-1}, \mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_P\}$. In the above schedule, let the weight of $\mathcal{O}_p$ at $(q-1+p)^{th}$ slot be w_{qp} for $1 \leq p \leq P$. Note that $\sum_p w_{qp} + \sum_{i=1}^{q-1} w(\mathcal{A}_i) = |\mathcal{G}|$. Thus, there exists a p such that $w_{qp} \geq (|\mathcal{G}| - \sum_{i}^{q-1} w(\mathcal{A}_i))/P = c$. For such a p, the weight w_{qp} of O_p at the $(q-1+p)^{th}$ slot can be at most the weight of O_p at the q^{th} slot (since more subelements are not-fully-read at q^{th} slot than at $(q-1+p)^{th}$ slot). Thus, such an O_p must have a weight at least c at the q^{th} time slot. Thus, the maximum possible weight at q^{th}step of EGA-1 is at least c. ∎

MRS Problem in Multiple Channels. For the case of multiple channels, we use the same heuristic as the one presented in the previous section for the multiple channels, except that we use the weight function as defined in Definition 10.

When there are unlimited number of channels, the conflict-free coloring does not provide a solution for the Tags-Reading problem. In two dimensions, one possible heuristic could be to use a sequence of conflict-free colorings (where color corresponds to a time slot) until all the subelements are completely read. Such a strategy doesn't have any performance guarantee. However, the result of Theorem 6 does generalize to EGA-1.

Theorem 10: For the case of unlimited number of channels in three-dimensions, the modified (as in Section V-B) EGA-1 returns a $2(1+\epsilon)\ln|\mathcal{G}|$-approximate solution and runs in time polynomial in $|\mathcal{R}|$, $|\mathcal{G}|$, and 2^N for any $\epsilon > 0$. Here, N is the maximum number of readers in any hypercube of size $k\max(T, 2S) \times k\max(T, 2S) \times \max(T, 2S)$, where k is as in Equation 2. ∎

VII. Performance Evaluation

In this section, we evaluate the performance of our designed algorithms using a custom simulator. For the MCS problem, we compare the sizes of covering schedules computed by various algorithms, and for the MRS problem, we simulate a tree-splitting based link layer protocol and compare the sizes of reading schedules computed by various algorithms for a given random distribution of tags.

In the simulations, we uniformly and randomly distribute 50 readers in a rectangular region of size 100×100 units. For the MRS problem, we also distribute randomly 1200 tags in the region. For now, we consider interrogation and interference regions to be *circular disks, with the* default *radius/range being 20 units and 50 units respectively. We will consider irregular disks (as described later) for the last set of* *experiments.* For GA-1 and EGA-1 algorithms, we use $k = 2$ (i.e., $\epsilon = 1.25$), since higher values of k did not result in noticeable improvement in performance but were much slower. We compare our algorithms with the Colorwave algorithm [22] for the MRS problem or a Colorwave-like algorithm for the MCS problem. As discussed in Section III, other works on avoiding collisions in RFID systems either consider only tag-tag collisions [14], [17], [18], [21], or have very different objective criteria [7], [11], or assume sophisticated tag technology [2].

MCS Problem. First, we evaluate the performances of GA-1 and GA-M for the MCS problem. In this setting, we do not take the tag distribution into consideration, and compare the covering schedules of readers delivered by various algorithm. For comparison, we use a random algorithm similar to the Colorwave algorithm [22], wherein each reader picks a random time slot, such that interfering readers have different time slots and each subelement in the monitored region is well-covered. In plots, we refer to this algorithm as *Colorwave-like*. Figure 5(a) shows the single channel performance with varying interference ranges. As expected, all algorithms perform worse (takes more time slots) with increasing interference range. The GA-M heuristic performs close to the approximation algorithm GA-1. The performance gap is bigger for larger interference range, because for the given parameter values (region size of 100×100 and $k = 2$) GA-1 solution is actually *optimal* for interference range ≥ 50. Figure 5b shows the single channel performance with varying interrogation range. We observe that the performance of each algorithm improves with increase in interrogation range, because larger interrogation region entails a larger coverage area. For both the above experiments, GA-1 and GA-M perform significantly better than Colorwave-like algorithm for all range values. Since Colorwave-like algorithm is an example of a random access scheme, the above exemplifies the superiority of scheduled access schemes in RFID systems.

Multiple Channels. Figure 5(c) shows multi-channel performance of GA-M for varying number of channels and the default range values. *Note that GA-1 and Colorwave-like algorithms work only for single channel; the plot shows their single channel performances for comparison.* We note that GA-M's performance indeed improves with more channels. However, the improvement is not significant because of a relatively small interference range. Use of multiple channels is expected to make more significant impact when interference range is relatively larger. To validate the last statement, we ran a separate experiment with different parameter values: 200 readers, interrogation range = 8 units and interference range = 60 units. See Figure 6. Note the almost proportionate decline in number of slots with increasing number of channels initially, and then, a saturation effect after about 4 channels. The saturation effect is because at that point, the number of active readers in a time slot is large enough that the reader-reader collisions (which can't be resolved using more channels) become dominant.

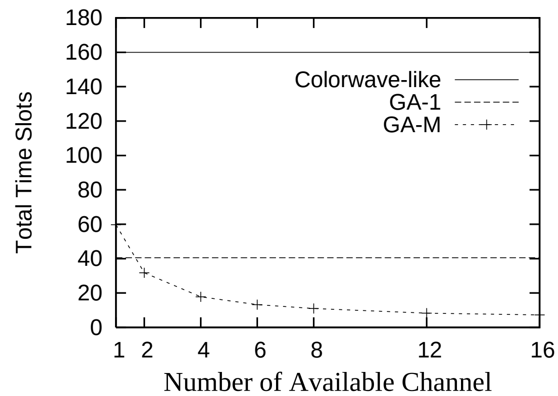

Fig. 6. Varying number of channels with larger interference range for the MCS problem.

MRS Problem. In the second set of experiments, we evaluate the performances of EGA-1 and EGA-M algorithms for the MRS problem. Here, we use a random distribution of 1200 tags in the region as part of the input, and use Colorwave for a baseline comparison. As mentioned before, a tree-splitting based link layer protocol is used for our algorithms here. We use the time slot size equivalent to make three edge traversals, since it was found to be most efficient for the given parameters (see below). For the single channel case (Figure 7(a)-(b)), the *relative* performance of various algorithms is similar to that observed in the MCS problem. We note that EGA-M heuristic performs same as the EGA-1 for small values of interference range, and performs close for larger values, for the same reason as discussed in the MCS problem. However, in Figure 7(b), we notice that the performance of Colorwave actually worsens with increase in the interrogation range. This implies that Colorwave algorithm is not effective in handling the reader-reader collisions, and this ineffectiveness seems to far outweigh the advantage of increase in coverage area. Note that Colorwave is indeed incapable of handling reader-reader collisions, since the tags do not participate in the algorithm (collision detection). Similarly, EGA-M heuristic's performance also worsen with increase in interrogation range for smaller values. In contrast, EGA-1's performance always improves with increase in interrogation range, *which implies that EGA-1 is most effective in handling the reader-reader collisions.*

Multiple Channels. In Figure 7(c), we observe that the increase in number of channels has more significant impact (compared to the MCS problem) on the performance of EGA-M.

Time Slot Size. We now present results of our simulations to estimate the "perfect" time slot size. As mentioned before, if tree-splitting protocol is used in each time slot, then the time slot size is chosen as the number of edge traversals that will result in at least one tag being read. Such a number depends on the density of the tag IDs and distribution – higher density would require a larger number of edge traversals – for a fixed interrogation range. For our chosen parameters, viz., 16-bit tag

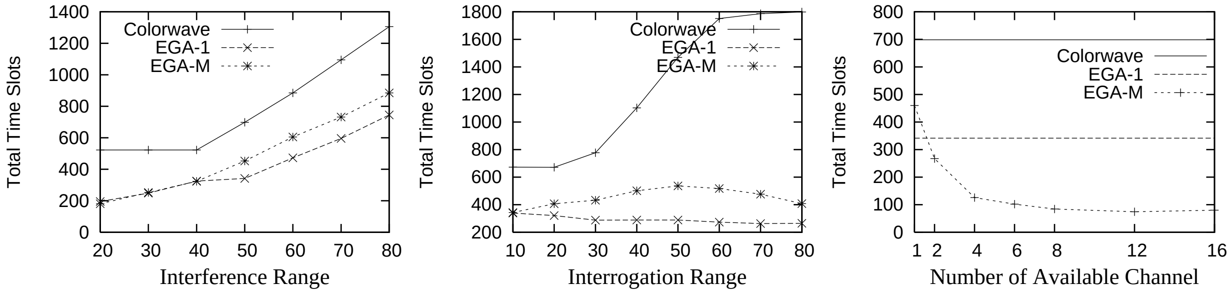

Fig. 7. Performance of the EGA-1, EGA-M, and Colorwave algorithms for the MRS problem. (a) Varying interference range with single channel, (b) Varying interrogation range with single channel, and (c) Varying number of available channels.

IDs, 100 $\times$ 100 region size, 1200 tags, interrogation range of 20 units, we observed that choosing the time slot size of three edge traversals was most efficient. See the below table.

TABLE I
FRACTION OF 1200 TAGS READ

Time Slot Size (l)	EGA-1	EGA-M
2	0.695	0.720
3	0.976	0.986
4	0.999	0.999

We observe in the above Table 1 that increase in the time slot size (l) improves the fraction of tags read. However, this will also proportionately increases the total run time (= number of slots in the reading schedule $\times$ l). Too small a value for l makes the algorithms underestimate the number of slots needed, leading to unread tags. Too large a value, on the other hand, makes slots unnecessarily longer and degrade performance (in terms of absolute time). Note that for later iterations of the reading schedule, we can use smaller time slot size due to reduction in number of unread tags.

Experiments with Regions as Irregular Disks. To illustrate the efficacy of our techniques for general shapes of regions, we conduct the above experiments with the interference and interrogation regions as irregular disks. In particular, for *each* reader I, we generate an irregular region of "range" r, by randoming choosing six points at a distance of more than $0.6r$, but less than r. The chosen six points are then sorted around the reader I, and connected to create a polygonal region. Based on the above way of constructing irregular interrogation and interference regions, we conduct the above experiments. See Figure 8 and 9. In general, we observe similar pattern of performance comparison as before (for regular regions in Figure 5 and 7), except that here in some cases GA-M and EGA-M marginally outperfom GA-1 and EGA-1 respectively. We also note that Colorwave's performance remains relatively unchanged (compared to before); this is because increase in interference *range* of irregular regions does not necessarily result in proportional increase in intersection of the regions due to possible "intertwining" of regions. Finally, note that in Figure 8, the Colorwave Algorithm's performance for interrogation range of 10 is abruptly high, because of minimal reader-reader collissions.

Summary. In summary, our simulation results show the following. (i) For the case of one-channel, our heuristics perform close to the approximation schemes and much better than Colorwave [22]; for the MRS problem, EGA-1 is most effective in handling reader-reader collisions. (ii) For the case of multiple channels, our heuristics perform proportional to the number of channels available (upto the saturation point) for reasonable choice of parameters.

VIII. Conclusions

In this paper, we addressed the problem of efficient reading of RFID tags in a multi-reader system. Multiple readers provide concurrency and also better coverage, but also bring in additional collision problems. We have used a slotted time model, and developed algorithms to compute a near-optimal activation schedule for the readers. We have considered two scenarios – one where the distribution of tags is unknown and the other where it is known. We have considered suitable models of the tag reading problem in these scenarios.

Our algorithms assume a planned deployment of readers where a prior site survey is possible to determine interference and interrogation regions of the readers. This is a departure from more conventional adaptive approaches. However, our approach is able to produce near-optimal schedule in the single channel case. The schedule does not need to be computed dynamically. It can be computed only once, and the readers activated according to the computed schedule to read tags.

Computing a near-optimal schedule for the multiple channels case is still an open question. However, we have developed efficient heuristics. Empirical evaluations suggest that the heuristics perform quite close to the approximation algorithms for the single channel case. Evaluations also suggest that our algorithms are far superior than Colorwave, a random access based protocol targeted for similar multiple reader systems.

Acknowledgement

Zhou and Das's work was partially supported by NSF grant CNS-0519734, and Gupta's work was partially supported by NSF grants IIS-0713186, CNS-0721701, and CNS-0721665.

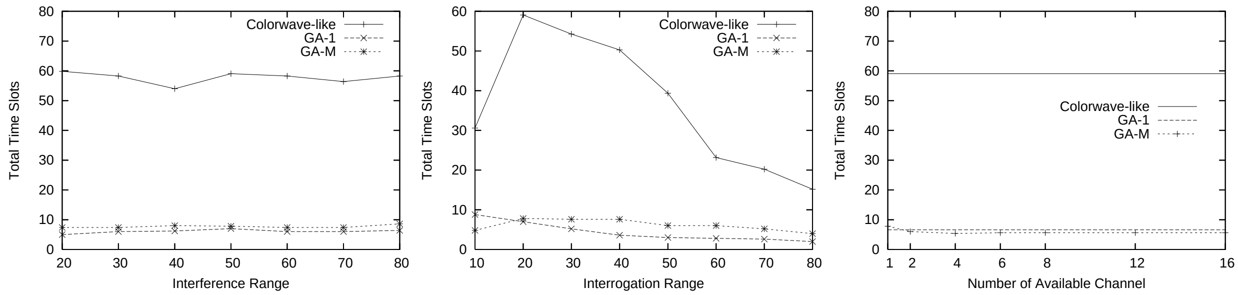

Fig. 8. Performance of the GA-1, GA-M, and Colorwave-like algorithms for the MCS problem *with irregular regions*. (a) Varying interference range with single channel, (b) Varying interrogation range with single channel, and (c) Varying number of available channels.

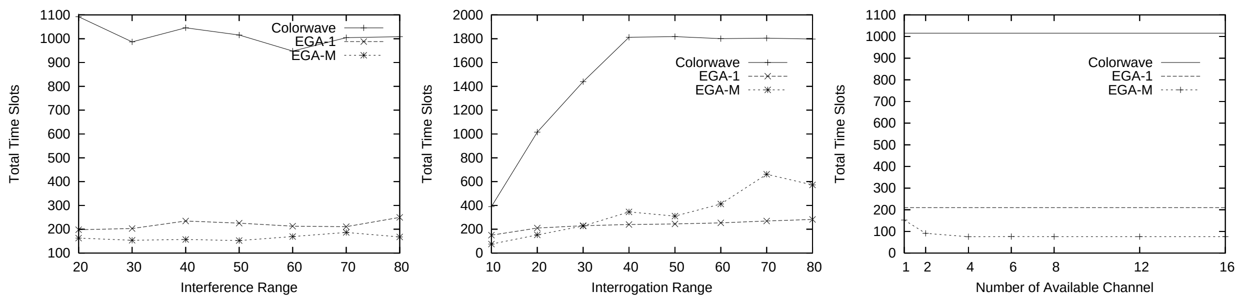

Fig. 9. Performance of the EGA-1, EGA-M, and Colorwave algorithms for the MRS problem *with irregular regions*. (a) Varying interference range with single channel, (b) Varying interrogation range with single channel, and (c) Varying number of available channels.

REFERENCES

[1] Passive RFID basics, 1998. `http://ww1.microchip.com/downloads/en/AppNotes/00680b.pdf`.

[2] Class 1 gen. 2 uhf air interface protocol standard version 1.0.9, 2005.

[3] N. Abramson. The ALOHA system - another alternative for computer communication. In *Joint Computing Conference Proceedings*, 1970.

[4] H. Bronnimann and M. Goodrich. Almost optimal set covers in finite VC-dimension: (preliminary version). In *SoCG*, 1994.

[5] I. Chlamtac, C. Petrioli, and J. Redi. Energy-conserving access protocols for identification networks. In *IEEE/ACM ToN*, 1999.

[6] T. Cormen et al. *Introduction to Algorithms*. McGraw Hill, 2001.

[7] V. Deolalikar et al. Perturbative time and frequency allocations for RFID reader networks. Technical report, HP Labs, 2005.

[8] G. Even, Z. Lotker, D. Ron, and S. Smorodinsky. Conflict-free colorings of simple geometric regions with applications to frequency assignment in cellular networks. In *FOCS*, 2002.

[9] H. Gupta et al. Connected sensor cover: Self-organization of sensor networks for efficient query execution. In *IEEE/ACM ToN*, 2006.

[10] S. Har-Peled and S. Smorodinsky. On conflict-free coloring of points and simple regions in the plane. In *SoCG*, 2003.

[11] J. Ho, D. Engels, and S. Sarma. HiQ: A hierarchical q-learning algorithm to solve the reader collision problem. In *SAINTW*, 2006.

[12] D. S. Hochbaum and W. Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32, 1985.

[13] H. Hunt et al. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *J. Algorithms*, 26, 1998.

[14] D. Hush and C. Wood. Analysis of tree algorithms for RFID arbitration. In *Proc. of ISIT*, 1998.

[15] S. Jain and S. R. Das. Collision avoidance in multi-reader RFID networks. In *WiNTECH*, 2006.

[16] M. Kodialam and T. Nandagopal. Fast and reliable estimation schemes in RFID systems. In *MobiCom*, 2006.

[17] C. Law, K. Lee, and K. Siu. Efficient memoryless protocol for tag identification. In *Intl. Wksp. on Dis. Algos. and Methods*, 2000.

[18] J. Myung and W. Lee. Adaptive splitting protocols for RFID tag collision arbitration. In *MobiHoc*, 2006.

[19] S. Smorodinsky. Personal Discussion.

[20] S. Smorodinsky. On the chromatic number of some geometric hypergraphs. In *SODA*, 2006.

[21] H. Vogt. Efficient object identification with passive RFID

tags. In *Proc. of the Intl. Conf. on Pervasive Computing*, 2002.

[22] J. Waldrop, D. Engels, and S. Sarma. Colorwave: A MAC for RFID reader networks. In *Proc. of WCNC*, 2003.

[23] W. Wang et al. Efficient interference-aware TDMA link scheduling for static wireless networks. In *MobiCom*, 2006.