2011 IEEE International Parallel & Distributed Processing Symposium

Reader Activation Scheduling in Multi-Reader RFID Systems: A Study of General Case

Shaojie Tang*, Cheng Wang†‡, Xiang-Yang Li*, Changjun Jiang†‡
* Department of Computer Science, Illinois Institute of Technology, Chicago, IL, 60616
† Department of Computer Science, Tongji University, Shanghai, China
‡ Key Laboratory of Embedded System and Service Computing, Ministry of Education, Shanghai, China

***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 guarantee the coverage quality, multiple RFID readers can be deployed in the given region. In this paper, we consider the problem of activation schedule for readers in a multi-reader environment. In particular, we try to design a schedule for readers to maximize the number of served tags per time-slot while avoiding various interferences. We first develop a centralized algorithm under the assumption that different readers may have different interference and interrogation radius. Next, we propose a novel algorithm which does not need any location information of the readers. Finally, we extend the previous algorithm in distributed manner in order to suit the case where no central entity exists. We conduct extensive simulations to study the performances of our proposed algorithm. And our evaluation results corroborate our theoretical analysis.**

***Index Terms*—RFID, Scheduling, Graph Theory.**

I. INTRODUCTION

Radio Frequency Identifier (RFID) technology attracts increasingly interests these days from both academic and industrial areas. It has been considered as a promising technology for tagging and identifying objects. RFID enables simultaneous detection of multiple, distant, and non-line-of-sight objects. Hundreds of millions of RFID tags are deployed until now; and the number is growing even faster currently. The main advantage of RFID systems is its accuracy and low deployment cost. For example, Wal-Mart save billions of dollars by introducing RFID into their goods management system. Typically, RFID systems comprise two components: RFID tags and RFID readers. RFID tags are usually attached to the objects and store the information of corresponding object. Avoiding the interference, readers are capable of reading the information stored on tags placed within its interrogation range. In this work, we consider all tags as passive tags, *i.e.* they do not need to be equipped with battery. Instead, they leverage the energy from the reader's signal to process the store data and communicate with readers.

In this work, we put our focus on improving the *read throughput* of a given multi-reader RFID system. In many locations essential to daily life such as supermarket or post office, multiple RFID readers are needed in a given region to perform tag reading concurrently, which will improve the read throughput greatly. In most existing works, RFID readers are assumed to be static and carefully deployed in a planned fashion. When the location of each reader is known a prior and all the readers has identical interference range, Zhou *et al.* [7] and Tang *et al.* [9] proposed a set of RFID reader scheduling protocols to improve the read throughput of multiple readers RFID system. However, their scheme often suffers from its "ideal" model setting in practical. First, they assume that all readers are deployed in a planned manner and the geometry location of each reader is known a prior. However, in a more realistic model, the position of each reader is often highly dynamic and we can not expect that that their exact geometry location can always be obtained. Second, all of their algorithms are centralized which require a central entity in the system. Unfortunately, this requirement can hardly be satisfied either, especially for large scale RFID systems. Further, even though they claim that all their proposed algorithms can terminate within polynomial time, their analysis heavily depends on the assumption that all readers have the same (or the same order of) interference range. Again, this assumption is indeed too strong in practical where different readers may be equipped with different antennas. Thus, to design an efficient and effective reader activation scheduling by considering all the issues listed above becomes extremely urgent while challenging under large scale RFID systems. In this work, we study the reader activation scheduling problem under a more general model. As main contributions of this paper, we remove those "ideal" assumptions one by one as follows:

1) We first study the model under which different reader may have various interrogation range and interference range. A PTAS (Polynomial-time approximation scheme) has been provided;
2) We next break the restriction that all readers' locations are known a prior. Specially, we propose a centralized scheduling scheme which does not need any location information of the readers;
3) At last, we extend previous centralized algorithm in a distributed manner to suite the case where no central entity exists.

This paper is organized as follows: In Section II, we present a number of basic definitions and background knowledge regarding multi-reader RFID systems; We then formally define the problems addressed in this paper in Section III; Section IV describes a centralized algorithm under the assumption that

1530-2075/11 $26.00 © 2011 IEEE
DOI 10.1109/IPDPS.2011.109

different readers may have different interference and interrogation range; In Section V, we propose another novel algorithm based on the growth bounded property of interference graph, which does not require any geometry location information of the readers. And we further extend this algorithm in a distributed manner; Finally, we evaluate all proposed algorithms in Section VI. We conclude our discussion in Section VIII.

TABLE I
NOTATIONS USED IN PAPER

Symbol	Meaning
V	Set of readers
v	Reader
X	Scheduling set of readers at a single time-slot
$w(X)$	Weight of scheduling set X
R_i	Interference radius of reader v_i
γ_i	interrogation radius of reader v_i
$\mathcal{O}$	Set of interference disks
$\mathbf{O}(v_i)$	Interference disk of reader v_i with radius R_i
$\mathcal{O}(r,s)$	Set of survive disks under (r,s)-shifting
$N(v)^r$	r-hop neighborhood of v in interference graph
MWFS(S,I)	A MWFS contained in S and independent from I

II. BACKGROUND KNOWLEDGE

In this section, we briefly introduce some basic definitions and models used in this work.

Interrogation and Interference Regions: Each RFID reader is associated with an *interrogation radius*, γ_i, which varies from ten centimeters to hundred feets; and an *interference radius*, R_i. Let (x_i, y_i) denote the coordination of reader v_i in the two-dimensional deployment region. We define the *interference region* of v_i, which is denoted by $\mathbf{O}(v_i)$, as a disk centered at (x_i, y_i) with radius R_i. For ease of presentation, we assume $r_i = \beta R_i$ where $1 > \beta > 0$ is a constant. As in [7], given a set of readers V, we define the *region monitored by the readers* V as the union of the interrogation regions of V. These regions can be explored by a RF site survey, *e.g.*, using certain localization device and radio signal strength measurement device to achieve the survey.

Collisions in Multi-Reader Systems: We first introduce three typical collisions that may cause interference among simultaneous transmissions in RFID systems.

Tag-tag collision (TTc): TTc happens when multiple tags located in the interrogation region of the same reader transmit data simultaneously. See Figure 1(a). TTc can be successfully resolved through certain link-layered protocol *i.e.*, framed Aloha [20] or tree-splitting [16], [18]. In this work, we will not put extra efforts to dealing with TTc.

Reader-tag collision (RTc): When a reader is in the interference region of the other one, as illustrated in Figure 1(b), reader B 's transmission can affect the response from $\mathtt{Tag}_1$ to A. Under this case, any reader suffering RTc can not successfully read any tag. We must carefully schedule the activation of different readers in order to avoid RTc.

Reader-reader collision (RRc): When some tags are positioned within the overlapping interrogation regions of two readers, those tags are not able to tell the difference between the signals from those two readers. Different from RTc, RRc will only disable simultaneous transmissions within the overlapping interrogation region, however, the readers can still read other tags that are inside only its own interrogation region. As illustrated in Figure 1 (c), both $\mathtt{Tag}_2$ and $\mathtt{Tag}_3$ except $\mathtt{Tag}_1$ can be read successfully.

III. PROBLEM DEFINITION AND PRIMARY APPROACH

Consider a set of n readers $V = \{v_1, v_2, \cdots, v_n\}$ and a set of m tags $\{\mathtt{Tag}_1, \mathtt{Tag}_2, \cdots, \mathtt{Tag}_m\}$ deployed statically at a deployment region. Each tag can be covered by the interrogation range of one or more readers. We assume that here is no further request for $\mathtt{Tag}_i$ as long as it has accessed some reader. In the following contents, we name the tag which has not been served as *unread tag*. Except for specific notification, “tag” always refer to “unread tag” throughout this paper.

We next introduce a number definitions which will be used frequently throughout this paper:

Definition 1: (**Well-Covered Tag**) [7]: A tag $\mathtt{Tag}_i$ or its location is said to be well-covered by a reader v_i, wherein $X \subseteq V$ is the set of activated readers, if the below conditions hold.

- The reader v_i is in X, and the tag $\mathtt{Tag}_i$ is in the interrogation region of v_i.
- The reader v_i is not in the interference region of any other reader $v_j \in X$ in the given time-slot. This condition ensures that there are no RTc.
- There is no other reader v_j in X such that the tag $\mathtt{Tag}_i$ is in the interrogation region of v_j. This condition ensures that there are no RRc.

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

Definition 2: (**Feasible Scheduling Set**) Throughout this paper, we denote the euclidian distance between any two readers v_i and v_j as $||v_i - v_j||$, *e.g.* $||v_i - v_j|| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. For any pair of readers $v_i, v_j \in V$, we say they are independent if and only if neither v_i nor v_j is located within the other's interference disk, *e.g.* $||v_i - v_j|| > \max\{R_i, R_j\}$. We define the *feasible scheduling set* as a subset of readers $X \subseteq V$ such that for any two readers $v_i, v_j \in X$, v_i and v_j are independent with each other.

Clearly, a *feasible scheduling set* should not contain any subset of readers which may cause RTc. Next we introduce another key definition which will be used frequently throughout this paper.

Definition 3: (**Weight of Feasible Scheduling Set**) [9] Given a *feasible scheduling set* X, the weight $w(X)$ of X is defined as the total number of well-covered unread tags under the feasible scheduling set. It reflects the amount of tags that may have chance to get served from X, without causing RRc. By excluding those tags which are within the overlapping area of more than one activated readers's interrogation region, $w(X)$ can also be defined as the number of unread tags that are located at the interrogation region of *exactly one* reader in X.

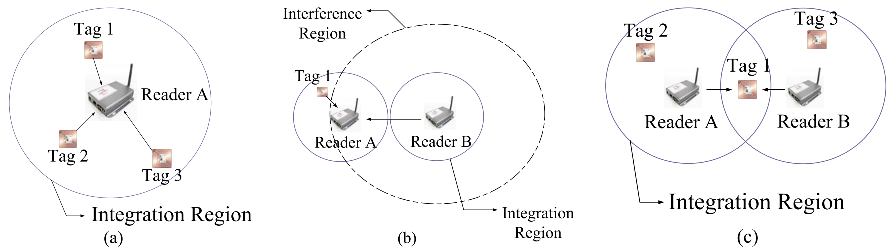

Fig. 1. [9]Collisions in RFID systems. (a) Tag-Tag collision: Tags 1, 2 and 3 respond to reader A simultaneously and causing collision at A; (b) Reader-Tag collision: Response from tag 1 to Reader A is "drowned" by the signal from the reader B; (c) Reader-Reader collision: Signal from reader A and B collide at tag 1

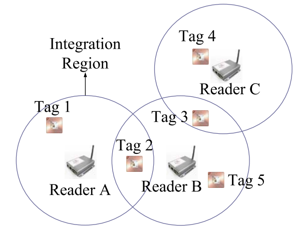

Fig. 2. [9]Illustration of an example in which scheduling less readers will read more tags.

Consider a example in Figure 2, assume there are three independent readers A, B and C. By selecting A, B, C as a feasible scheduling set, *e.g.* $X = \{A, B, C\}$, the resulted total weight is $|\{\mathtt{Tag}_1, \mathtt{Tag}_2, \mathtt{Tag}_3, \mathtt{Tag}_4, \mathtt{Tag}_5\}| - |\{\mathtt{Tag}_2, \mathtt{Tag}_3\}| = 3$. However, if we only activate A and C, *e.g.*, $X = \{A, C\}$, the weight is is increased to $|\{\mathtt{Tag}_1, \mathtt{Tag}_2, \mathtt{Tag}_3, \mathtt{Tag}_4\}| - |\{\emptyset\}| = 4$.

Definition 4: (**Covering Schedule of Readers**) Consider a set of readers V. Let M be the region monitored by V(*i.e.* the union of their interrogation regions). A covering schedule of readers for V is an assignment of readers to each time-slot (being active), such that each location in M is well-covered by some reader in one of the time slots. Here, the total number of required time-slots is called the size of the scheduling. Again, after tag $\mathtt{Tag}_i$ accessing some reader, we say that it leaves the system, that is, we will simply ignore all those tags in the following schedule.

Now we are ready to formally define the problems studied in this paper. Note that the time-slot size is chosen such that each active reader v_i is able to read at least one tag within the time-slot, if there is at least one tag well-covered by v_i. Thus, if we iterate over a covering schedule of readers, then we are guaranteed to read all tags which are covered by our system. This can be easily achieved by rendering a tag passive (using a lower layer protocol) when it has been served; thus, any tag that already been served will not participate in later iterations. The number of iterations required to read all the tags depends on the maximum number of tags well-covered by a reader in any time-slot of the given covering schedule. We first define the minimum covering schedule problem as follows:

Definition 5: (**Minimum Covering Schedule Problem**): Given a set of readers V and its associated regions, the *minimum covering schedule* problem (MCS) is to find the minimum-size covering schedule of readers for V.

Notice that most geometric set-cover problems can be shown to be NP-hard. Since the above defined MCS problem can be regarded as a special case of geometric set-cover problem by setting each reader's interference range sufficiently large, it is also NP-hard. If the NP-Hardness for this problem depresses us somehow, our approximation results proposed in this paper should bring us good news. We next introduce a simple greedy algorithm which acts as the backbone of our scheduling scheme:

1) At the q-th time-slot, we choose a feasible scheduling set with maximum weight and let them be active at time-slot q;
2) It terminates when there are no unread tags remained.

Clearly, we attempt to use one time-slot to its full capacity in the greedy algorithm.

Theorem 1: [7] This simple greedy algorithm can achieve $\log n$-approximation for MCS problem. Here n is the number of readers.

We still use the same greedy algorithm as the backbone of our scheduling scheme. Clearly, if we can successfully find a maximum weighted feasible scheduling set at each time-slot, we can still achieve $\log n$-approximation through above greedy algorithm for MCS problem. The proof is same as the one provided in [7] and omitted here to save space. Then the main challenging issue arising in this paper is how to select a feasible scheduling set with maximum weight under the general model. Specifically, we are interested at finding a maximum weighted feasible scheduling set at each time-slot by assuming no central entity exists, no geometry location information is pre-known and different readers may have various interference range. Since this problem is already non-trivial even under the "ideal" model [7], indeed, tackling the same problem under a more general model becomes much more challenging.

In this work, we mainly concentrate on finding a maximum weighted feasible scheduling set under general model. We next formally define the *One-Shot Scheduling Problem* as follows:

Definition 6: (**One-Shot Schedule Problem**): The objective of *One-Shot Schedule Problem* is to find a *Maximum Weighted Feasible Scheduling Set* (MWFS) at a single time-slot. Notice that we only consider the unread tags when computing the weight of any feasible scheduling set at any time-slot.

IV. One-Shot Schedule Problem With Location Information

We first study a simple case when each reader's position is known a prior and this assumption will be removed later. Here we propose an centralized algorithm by assuming different reader may have various interference/interrogation radius. And we theoretically prove that the weight of the scheduling set found by our algorithm is arbitrarily close to the optimum one.

Assume we are given a set of readers $V = \{v_1, v_2, \cdots, v_n\}$ and their corresponding interference disks $\mathcal{O} = \{\mathbf{O}(v_1), \mathbf{O}(v_2), \cdots, \mathbf{O}(v_n)\}$. In the following contents, we say two disks $\mathbf{O}(v_i)$ and $\mathbf{O}(v_j)$ are independent with each other if and only if v_i and v_j are independent from each other. For clarity of presentation, we will use $\mathbf{O}(v_i)$ and v_i interchangeably in the following contents to represent the same reader v_i. Hereafter we use *disk/radius* to represent *interference disk/radius* for short.

We build our PTAS for approximating MWFS based on the approach of [4] [22] [2], *i.e.* to divide the readers into different levels according to their interference radii. All readers at the same level have similar interference radii, *i.e.* they are within a constant factor of each other. The main idea of our algorithm is similar as the one proposed in [22] [2], the main challenge in generalizing the result of [22] [2] arises due to the fact that in our context $w(X_1 \cup X_2)$ may be less than $w(X_1) + w(X_2)$ for two sets of readers X_1 and X_2. We scale the interference radius of each reader so that the largest interference radius is $\frac{1}{2}$ and the smallest interference radius among all readers is denoted as R_{min}.

We partition the corresponding disks into $l+1$ levels such that level j, $0 \le j \le l$, consists of all disks with radius satisfying that $\frac{1}{(k+1)^{j+1}} < 2R_i \le \frac{1}{(k+1)^j}$. Let $l(\mathbf{O}(v_i))$ denote the level of disk $\mathbf{O}(v_i)$, *i.e.* $l(\mathbf{O}(v_i)) = \lfloor \log_{k+1} \frac{1}{2R_i} \rfloor$. For each level j, we subdivide the plane into grid by using a set of vertical lines $L_{j,v} : x = v\frac{1}{(k+1)^j}, v \in Z$ and a set of horizontal lines $H_{j,h} : y = h\frac{1}{(k+1)^j}, h \in Z$. Hereafter j is named the level of the lines $L_{j,v}$ and $H_{j,h}$; v (and h) is called the index of the vertical (and horizontal) line $l_{j,v}$ (and $h_{j,h}$) at level j. A (r,s)-shifting of the subdivision is the grid defined by the set of vertical lines whose indices modulo k equal r and the set of horizontal lines whose indices modulo k equal s. Different from the UDG case where there is only one (r,s)-shifting, we have different (r,s)-shiftings in different levels. *However, it was proved in [3] that a vertical line at level j of a (r,s)-shifting subdivision is also a vertical line at level $j+1$ of the (r,s)-shifting subdivision.* Then, we compute MWFS using the dynamic programming as discussed later.

Any two consecutive vertical lines at level j whose indices modulo k equal r, and any two consecutive horizontal lines at level j whose indices modulo k equal s, define a j-square in the (r,s)-shifting subdivision. Clearly, any j-square S can be subdivided into $(k+1)^2$ $(j+1)$-squares (by lines $L_{j+1,v}$ and $H_{j+1,h}$ at level $j+1$). Denoted by $S' \prec S$, these $(j+1)$-squares are called the children of S, and S is called the parent of S'.

A disk $\mathbf{O}(v_i)$ is said to hit a vertical line at $x = a$ if $a - R_i < x_i \le a + R_i$. Similarly, we say the disk $\mathbf{O}(v_i)$ hits a horizontal line at $y = b$ if $b - R_i < y_i \le b + R_i$. For MWFS problem, a disk $\mathbf{O}(v_i)$ at level $l(\mathbf{O}(v_i)) = j$ is said to be survive (respecting to (r,s)-shifting) if it does not intersect the boundary of any j-square of the (r,s)-shifting subdivision. For each level j, let $\mathcal{O}_j(r,s)$ be the set of survive disks at level j respecting to (r,s)-shifting. Define $\mathcal{O}(r,s) = \bigcup_{j=0}^{l} \mathcal{O}_j(r,s)$, *i.e.* the union of survive disks at all levels respecting to (r,s)-shifting. Then a j-square S is named relevant if $\mathcal{O}(r,s)$ contains at least one disk of level j that is inside S. Please see Figure. 3 for an example where $r = s = 0$ and $k = 3$.

We first restrict that an optimum solution $OPT(\mathcal{O}(r,s))$ under (r,s)-shifting should not contain any disk that hits a line at level 0, *e.g.*, each disk in $OPT(\mathcal{O}(r,s))$ must be contained inside some 0-square. For ease of analysis, $OPT(\mathcal{O}(r,s))$ can be further divided into two subsets: (1) some independent disks at level 0, denoted by I_0; (2) independent disks at higher level that are independent with all disks from I_0. The above partition of disks can be performed recursively down to the squares at lower levels.

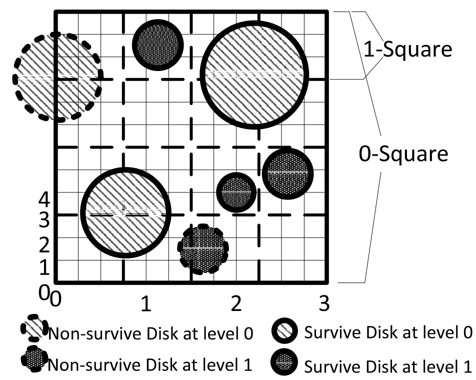

Fig. 3. A illustration of the survive disks at level 0 and level 1. Note that a disk is survive if and only if it does not intersect the boundary of any square at the same level. In this figure, $r = s = 0$ and $k = 3$.

Based on above partition idea, we next compute MWFS(S, I) using the dynamic programming. We first assume that the entry MWFS(S, I) for all squares S (with level at least $j+1$) and all appropriate feasible sets I intersecting S is already computed. The interference disks in MWFS(S, I) can be divided into two subsets: (1) D contains all independent disks, that are independent with disks from I, inside S with level j; (2) independent disks that are independent with any disk from I and D with level larger than j.Notice that each disk in the second subset is contained inside some $(j+1)$-square S'. We then compute MWFS(S, I) by scanning all possible D as follows,

$$MWFS(S, I) = \max_{D} \left(\left(\bigcup_{S' \prec S} MWFS(S', I_{S'} \cup D_{S'}) \right) \bigcup D \right)$$

Here $I_{S'}$ is the subset of disks from I who intersect S'. $D_{S'}$ is defined similarly.

The algorithm processes all relevant squares in order of non-increasing levels. For each j-square S and some appropriate independent set I, MWFS(S, I) is computed by dynamic programming, as shown in Algorithm 1.

Theorem 2: Given a set of readers and their geometry locations, there is at least one (r, s)-shifting, $0 \leq r,\ s < k$ such that $w\left(OPT(\mathcal{O}(r, s))\right) \geq (1 - \frac{1}{k})^2 w(OPT)$ where $w(OPT)$ denote the weight of optimum solution(*e.g.* maximum weighted feasible scheduling set).

Theorem 2 basically shows that the weight of the feasible scheduling set computed by our algorithm can be arbitrarily close to the optimum solution. The proof is omitted here to save space.

Algorithm 1 Approximate Maximum Weighted Feasible Scheduling Set (MWFS) With Location Information

---

Input:All n readers, their geometry locations, interference area, and interrogation region.
Output: Feasible scheduling set $X \subseteq V$.

for all $j = l + 1$ downto 1 **do**
  for all square S with level j **do**
    Let Y be survive readers under (r, s)-shifting of level $\leq j$ and intersecting S.
    for all $J \subseteq Y$ with at most Λ disks **do**
      if J is a feasible scheduling set **then**
        Let X be independent disks in J with level j.
        for all child square S' of S **do**
          Let $J_{S'}$ be disks in J intersecting S'.
          Set $X = X \cup$ MWFS$(S', J_{S'})$.
        end for
        Let I be readers in J with level less than j.
        if $w(X) > w($MWFS$(S, I))$ **then**
          MWFS(S, I) = X
        end if
      end if
    end for
  end for
end for
$X \leftarrow \cup_S$:

V. One-Shot Schedule Problem Without Location Information

In the previous section, we propose an efficient centralized algorithm under the assumption that the geometry location of each reader is known a prior. In this section, we first design a centralized algorithm which does not need any location information. And we further extend this algorithm in a distributed manner where no central entity is required.

A. Centralized Scheme

We first present a centralized approach for One-Shot Schedule problem where no geometry information of each reader is required. We assume that a interference graph $G = (V, E)$ is available through network measurement[1]. Here we give a formal definition of G as follows:

[1]This can be done by a RF site survey using a localization device and radio signal strength measurement device.

Definition 7: **Interference Graph** of the readers V is a graph where every reader in V has a corresponding node, and any two nodes have an edge between each other if and only if one reader is located in the interference region of the other. In other words, any two adjacent readers in interference graph cannot be active simultaneously due to RTc.

Our proposed method is based on the algorithm for maximum weighted independent set problem proposed in [15]. The basic idea of our approach is that, we first select a reader v with maximum weight (by activating v alone); then we compute maximum weighted feasible scheduling set $\Gamma^r(v)$ in the r-hop neighborhood $N(v)^r$ of v which includes v. $N(v)^r$ of node v is defined as:

$$N(v)^r := \{u \in V | u \text{ has hop distance at most } r \text{ from } v\}$$

Hereafter, we use Γ^r and N^r to represent $\Gamma^r(v)$ and $N(v)^r$ respectively. We repeat the process when the weight of Γ^r meets the following requirement:

$$w(\Gamma^{r+1}) \geq \rho \cdot w(\Gamma^r) \tag{1}$$

where $\rho = 1+\epsilon$ and $\epsilon > 0$. The process stops when r increases to $\bar{r}$ and inequality (1) is violated for the first time. We will prove the existence of $\bar{r}$ later.

We then "remove" $N^{\bar{r}+1}$ of node v including v. We repeat the above process until all the nodes in the network are "removed". Assuming that the nodes we have picked are $v_1, v_2, v_3, \cdots, v_m$, the candidate solution for X is the union of $\Gamma^{\bar{r}_i}(v_i)$. Note that we remove $(\bar{r}_i + 1)$-neighborhood of v_i instead of $N(v_i)^{\bar{r}_i}$ in order to ensure that the union of $\Gamma^{\bar{r}_i}(v_i)$ is a feasible scheduling. Please see Figure. 5 for an example.

Algorithm 2 Centralized Reader Activation Scheduling without Location Information

---

Input: $G = (V, E)$ and ρ.
Output: Feasible Scheduling Set X.

repeat
  Pick a reader v with maximum weight by activating it alone;
  Compute $\Gamma^{\bar{r}}(v)$;
  $X = X \cup \Gamma^{\bar{r}}(v)$;
  $V = V \backslash N(v)^{\bar{r}+1}$;
until $V = \varnothing$;

Now we prove that $\bar{r}$ does exist and is bounded by a constant (depending on ρ).

Theorem 3: [15] There exists a constant $c = c(\rho)$ such that $\bar{r} \leq c$.

The correctness and approximation guarantee of the algorithm follows from the following theorem, and the proof is similar as [15] thus omitted here.

Theorem 4: X generated by Algorithm 2 is an feasible scheduling set of weight at least $\frac{1}{\rho} = \frac{1}{1+\epsilon}$ of maximum weighted feasible scheduling set.

B. Distributed Scheme

Compared with centralized scheduling algorithms, distributed algorithms are more realistically and efficiently, specially for large scale RFID systems. The main idea of our distributed algorithm without knowing location information is that we let every reader in the network collect information from other nodes within a limited hops in G; when it finds itself with maximum weight (by activating itself alone), it starts the local computation of MWFS. Notice that in centralized scheduling, to guarantee the correctness, we start from a reader v with the largest weight and then grow the region until a certain criterion (inequality (1)) is violated. Given ρ, we know that we will explore at most $\bar{r}$-hops neighborhood $N^{\bar{r}}$ of v in the interference graph. Thus, other readers that do not have the global largest weight can also start to explore its neighborhood $N^{\bar{r}}$ and find an MWFS simultaneously. Let $c \geq \bar{r}$ be the control parameter depending on ρ. As proved in Theorem 5, c is some constant. To ensure the consistency of these two simultaneous explorings, we require any two initiating readers to be separated by at least $2c+2$ hops in the interference graph G. Please see Figure. 5 for illustration. Our main idea is as follows: **Step 1:** Initially, every reader collects the information from its $(2c+2)$-hop neighborhood of interference graph G. Here constant c is a system parameter. The reader with maximum weight in its $(2c+2)$-hop neighborhood will become a coordinator for local MWFS computation. We choose the value of c in a way such that every node initiating a local MWFS computation terminates the computation within c hops. We can choose a appropriate c according to Theorem 5. Note that we let each node collect information in its $(2c+2)$-hop neighborhood to ensure that MWFSs computed by simultaneous coordinators will always be interference free.

Step 2: Based on the collected information, if a reader v has maximum weight among all its neighbors within $(2c+2)$ hops, v starts to compute local MWFSs Γ^0, Γ^2, ..., $\Gamma^{\bar{r}}$ by enumeration. Same as the centralized algorithm, we find a $\bar{r}$ such that $\rho w(\Gamma^r) \leq w(\Gamma^{r+1})$ when $r < \bar{r}$ and $\rho w(\Gamma^{\bar{r}}) > w(\Gamma^{\bar{r}+1})$. Let $\Gamma^{\bar{r}}$ denote the local result computed by reader v.

Step 3: v announces $\Gamma^{\bar{r}}$ in its $(\bar{r}+1+2c+2)$-neighborhood of G and "removes" $N^{\bar{r}+1}$ from G. As a result, some node $u \in N^{\bar{r}+1+2c+2} \setminus N^{\bar{r}+1}$ might find that it has the maximum weight in its $(2c+2)$-hop neighborhood and it can start to compute its local MWFSs. Please refer to Algorithm 3 for details. A Red node is included in the solution of the current round, while a Black node is excluded. In addition, we use several colors to distinguish the different status of readers. For example, if a reader v marks itself with color *Red*, it means v is selected in the solution for this round. *Black* means the node is not selected as the solution in this round.

Theorem 5: There exists a constant $c = c(\rho)$ such that $\bar{r} \leq c$.

The proof is similar to that of Theorem 3 which is based on the observation that

$$w(\Gamma^r) \geq \rho w(\Gamma^{r-1}) \geq \rho^2 w(\Gamma^{r-2}) \geq \cdots \geq \rho^r w(v)$$

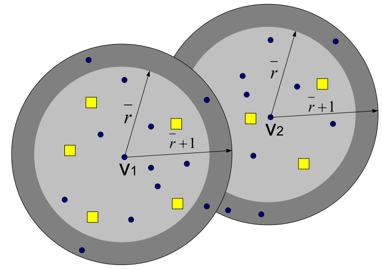

Fig. 4. Illustration of the centralized algorithm. We start from v_1 who has the maximum weight until certain criteria is violated, then we choose v_2 to start its local computing. The light squares denote a possible set of feasible scheduling set and dark nodes denote the remaining readers.

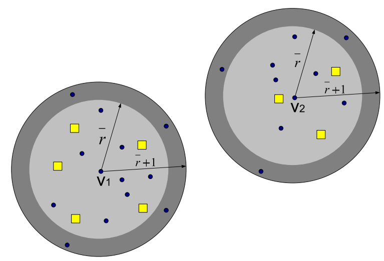

Fig. 5. Illustration of the distributed algorithm. v_1 and v_2 start their local computing simultaneously. Note that v_1 and v_2 need to be departed by at least $2\bar{r}+2$ hops away from each other to ensure the feasibility.

Surprisingly, the performance of our distributed algorithm is also arbitrarily close to the optimum solution.

Theorem 6: X generated by Algorithm 3 is an feasible scheduling set and $w(X) \geq \frac{1}{\rho} w(OPT)$.

Proof: Let $\bar{V} = V \setminus N^{\bar{r}+1}$, and inductively assume that $\Gamma \subset \bar{V}$ is a $\rho-$approximation feasible scheduling set in $G[\bar{V}]$. Obviously, $X = \Gamma^{\bar{r}} \cup \Gamma$ is an feasible scheduling set. Since $\Gamma^{\bar{r}+1}$ is an MWFS in $N^{\bar{r}+1}$, we have $w(OPT \cap N^{\bar{r}+1}) \leq w(\Gamma^{\bar{r}+1}) \leq \rho w(\Gamma^{\bar{r}})$. ■

VI. EVALUATION

In this section, we evaluate the performance of our designed algorithms using a custom simulator. In the simulations, we uniformly and randomly distribute 50 readers and 1200 tags in a square region of side-length 100 units. And we also randomly assign different interference range and interrogation range to each reader following Poisson distribution with parameter (mean) λ_R and λ_r respectively. We may need to modify some assignments to ensure $R_i \geq r_i$. We compare our methods with Colorwave Algorithms (CA) and Greedy Hill-Climbing Algorithms (GHC). Under the greedy Hill-Climbing algorithm, we choose the scheduling set for each time-slot by the following greedy rule: at each step, we select a reader to add to current active reader set, in order to maximize the incremental weight together with other active readers at this time-slot. Then we keep adding the reader to the active set

Algorithm 3 Distributed Scheduling Without Location Information

Input: ρ, c(which is computed from ρ).
Output: A local feasible scheduling set.

1: state = White; active = NO; head = NO;
2: Collects information from $N(v)^{2c+2}$ in G.
3: **if** $w(v) \geq w(u)$, for any $u \in N(v)^{2c+2}$ **then**
4:     head = YES;
5: **end if**
6: **if** head = YES **then**
7:     Computes $\Gamma^0(v), \Gamma^2(v), ..., \Gamma^{\bar{r}}(v)$ such that $w(\Gamma^{i+1}(v)) \geq \rho \cdot w(\Gamma^i(v))$ for $0 \leq i \leq \bar{r} - 1$, and $w(\Gamma^{i+1}(v)) < \rho w(\Gamma^i(v))$ for $i > \bar{r} - 1$.
8:     Broadcasts message RESULT($\Gamma^{\bar{r}}(v)$) among $N(v)^{\bar{r}+1+2c+2}$;
9: **end if**
10: **if** state = White AND head = NO **then**
11:     **if** receives message RESULT($\Gamma^{\bar{r}}(u)$) **then**
12:         **if** $v \in \Gamma^{\bar{r}}(u)$ **then**
13:             state = Red; active = YES;
14:         **end if**
15:         **if** $v \in N(u)^{\bar{r}+1}$ AND $v \notin \Gamma^{\bar{r}}(u)$ **then**
16:             state = Black; active = NO;
17:         **end if**
18:         **if** $v \in N(v)^{\bar{r}+1+2c+2} \setminus N(v)^{\bar{r}+1}$ **then**
19:             If v has no White neighbor within $2c+2$ hops that has larger weight, goto 5;
20:         **end if**
21:     **end if**
22: **end if**

one by one recursively until the weight starts to decrease (the incremental weight becomes negative) due to various collisions.

1) To test the performance of each algorithm for the MCS problem, we compare the sizes of covering schedules computed by various algorithms. Remember that the size of covering scheduling is the total number of time-slots required to read all tags. As described in Section III, we use the greedy algorithm by recursively finding a maximum weighted feasible scheduling set to active at each time-slot until no unread tags left. Here we use different one-shot algorithms as described in this paper to find the maximum weighted feasible scheduling set at each time-slot and further evaluate their performance;

2) To test the performance for the one-shot scheduling problem, we compare the total number of well-covered tags at a given time-slot under different algorithms.

We first evaluate various algorithms for MCS problem. In the first round of experiments, we fix the parameter λ_R and compare the total number of time-slots required by each algorithm under various parameters λ_r. Alternatively, we next fix λ_r and compare them under different λ_R. The results are demonstrated in Figure.6 and Figure.7 respectively. As we expected, Algorithm 1 has the best performance in terms of least scheduling size. This is because Algorithm 1 is a

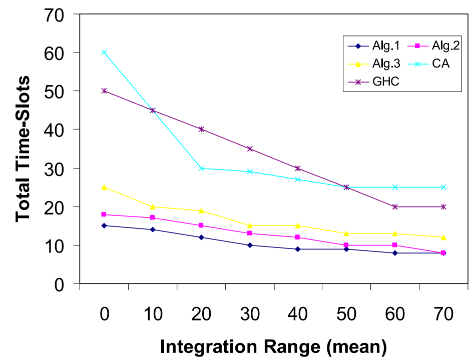

Fig. 6. The size of covering scheduling under various algorithms when λ_r is fixed.

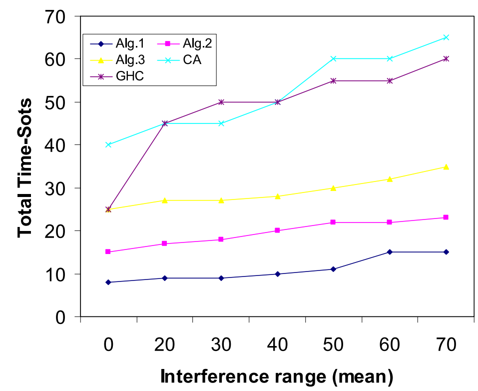

Fig. 7. The size of covering scheduling under various algorithms when λ_R is fixed.

centralized algorithm with global information of each reader's location. Algorithm 2 also performs much better than the rest algorithms, it also results from its centralized manner even though no exact location information is known. Since Algorithm 3 is a distributed algorithm without knowing any location information of each reader, it performs not as good as the previous two algorithms. However, it still beats CA and GHC in all range of values. We also observe that the performance of each algorithm improves as λ_R increases, because larger interrogation region provides a larger coverage area. And the gap between our algorithms and the others becomes even bigger when the interrogation range increases. Same as observed before, the total number of well-covered tags decreases as the interference range increases.

We next test all approaches for One-Shot Scheduling problem. In contrast to the evaluation in MCS where the size of scheduling is the main metric, we are interested at the total number of well-covered tags at one time-slot here. Similar as previous experiment setting, we evaluate each algorithm

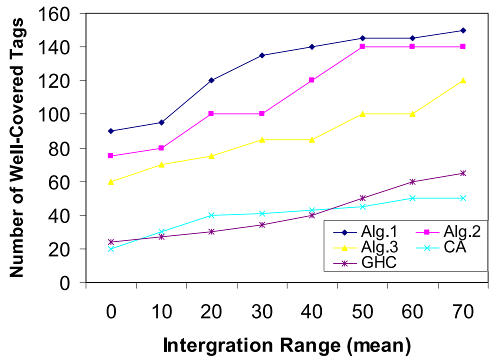

Fig. 8. The total number of well-covered tags under various algorithms when λ_R is fixed.

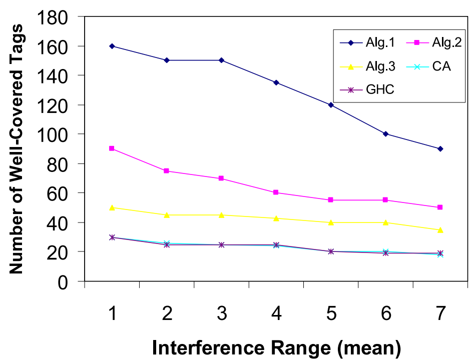

Fig. 9. The total number of well-covered tags under various algorithms when λ_r is fixed.

under different parameter setting for λ_R and λ_r. At the first experiment, we fix the λ_R and increase λ_r. Figure. 8illustrates the total weight of well-covered tags at one time-slot under various algorithms. In the second part, we test the same metric by fixing λ_r. As demonstrated in Figure. 8 and Figure. 9, all of our algorithms perform significantly better than the other algorithms. This is simply because all our approaches are able to find a feasible scheduling set with near maximum weight.

VII. Related Work

Recently, several protocols have been proposed to avoid collisions in RFID systems. We classify these results into three categories according to the type of collisions they addressed. For tag-tag collisions, several link layer protocols, *e.g.* [11], [16], [18], [20], were designed to avoid this collision. A popular solution is the tree walking algorithm (TWA) [16], [18] in which the reader splits the entire ID space into two subsets and tries to identify the tags belonging to one of the subsets, recursing along the way until a subset has exactly one tag or no tag at all. Recently, [19] proposes optimizations to tree traversal. In slotted Aloha protocol [10], a query frame is selected with a sufficiently large number of time slots and each tag sends a response at a random chosen time slot. When the reader hears a response correctly, it sends confirmation. If there is a collision, the colliding tags will choose another random slots to send a response. The reader further changes the frame size according to how frequently the collisions happen in the previous frame. For avoiding Reader-Reader or Reader-Tag Collisions. Colorwave [21] is one of the first work to address reader-reader collisions. In particular, it considers an "interference graph" over the readers, wherein there is an edge between two readers if they could lead to a reader-reader collision when transmitting simultaneously, and 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 time slot), only one of them wins (*i.e.*, sticks to the chosen time-slot), the others pick another time-slots again randomly. In [13], the authors suggest k-coloring of the interference graph where k is the number of available channels. If the graph is not k-colorable under their suggested heuristic, then they will remove 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 [8], 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. For a given network of readers and communication pattern, [14] 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 be allocated to a reader, when a new tag read request comes in. They assume a fixed number of time slots, and aim at maximizing the frequency and time utilization ratio. This work does not provide any performance guarantee. Most recently, [7] proposes a tag access scheduling protocol (EGA) based on STDMA. The authors try to schedule all the tag access such that the total reading time is minimized. In the case where the tag distribution is known, they attempt to solve the problem of assigning time slots to readers, such that each location of the deployment space is well-covered by some reader in one of the time slots. Then the authors extend and optimize their solution for the case where the tag distribution is unknown and also there are multi-channels to be used. Note that in their paper, they assume that the distribution of the tags are static and no new tags will appear in the system dynamically.

VIII. Conclusion

In this paper, we address the reader activation scheduling problem under large scale multi-reader RFID systems. For networks with geometry location, we propose a PTAS scheme. For networks without geometry location, we present a centralized algorithm based on the growth bounded property of interference graph. This algorithm is further extended in a distributed manner where no central entity is required. Clearly, all the algorithms proposed in this paper are well suited for practical implementation by removing a number of strong assumptions made in previous works.

ACKNOWLEDGMENTS

The research of authors is partially supported by the National Basic Research Program of China (973 Program) under grants No. 2010CB328101, No. 2011CB302804, and No. 2010CB334707, the Program for Changjiang Scholars and Innovative Research Team in University, the Shanghai Key Basic Research Project under grant No. 10DJ1400300, the NSF CNS-0832120, NSF CNS-1035894, the National Natural Science Foundation of China under grant No. 60828003 and No. 61003277, the Program for Zhejiang Provincial Key Innovative Research Team, and the Program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program).

REFERENCES

[1] C. LAW, K. LEE, AND K. SIU. Efficient memoryless protocol for tag identification. In *Intl. Wksp. on Dis. Algos. and Methods, 2000.*

[2] LI, X.Y. AND WANG, Y. Simple approximation algorithms and PTASs for various problems in wireless ad hoc networks. *Journal of Parallel and Distributed Computing*

[3] T. ERLEBACH, K. JANSEN, E. SEIDEL, Polynomial-time approximation schemes for geometric graphs, In *Proc. of Annual Symposium on Discrete algorithms, 2001.*

[4] I. HARRY B. HUNT, M. V. MARATHE, V. RADHAKRISHNAN, S. S. RAVI, D. J. ROSENKRANTZ, R. E. STEARNS, NC-approximation schemes for NP- and PSPACE- hard problems for geometric graphs, *J. Algorithms 26 (2) (1998) 238C274.*

[5] X. LIN AND N.B. SHROFF The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM Trans. Netw. 14*, 2 (2006), 302–315.

[6] S. SANGHAVI, L. BUI, AND R. SRIKANT Distributed link scheduling with constant overhead. In *ACM SIGMETRICS* (2007), pp. 313–324.

[7] ZONGHENG ZHOU AND GUPTA, H. AND DAS, S.R. AND XIANJIN ZHU Slotted Scheduled Tag Access in Multi-Reader RFID Systems. In *IEEE ICNP 2007*

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

[9] SHAO-JIE TANG, JING YUAN, X.-Y. LI, GUIHAI CHEN, YUNHAO LIU AND JIZHONG ZHAO RASPberry: A Stable Reader Activation Scheduling Protocol in Multi-Reader RFID Systems. In *IEEE ICNP 2009*

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

[11] I. CHLAMTAC, C. PETRIOLI, AND J. REDI. Energy-conserving access protocols for identification networks. In *IEEE/ACM ToN*, 1999.

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

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

[14] J. HO, D. ENGELS, S. SARMA., AND HI Q A hierarchical q-learning algorithm to solve the reader collision problem. In *SAINTW*, 2006.

[15] SHUICHI SAKAI AND MITSUNORI TOGASAKI AND KOICHI YAMAZAKI A Note on Greedy Algorithms for Maximum Weighted Independent Set Problem

[16] D. HUSH AND C. WOOD. Analysis of tree algorithms for RFID arbitration. In *Proc. of ISIT*, 1998.

[17] S. JAIN AND S. R. DAS. Collision Avoidance in a Dense RFID Network. In *WiNTECH*, 2006.

[18] C. LAW, K. LEE, AND K. SIU. Efficient memoryless protocol for tag identification. In *Intl. Wksp. on Dis. Algos. and Methods*, 2000.

[19] J. MYUNG ANDW. LEE. Adaptive splitting protocols for RFID tag collision arbitration. In *ACM MobiHoc*, 2006.

[20] H. VOGT. Efficient object identification with passive RFID tags. In *Proc. of the Intl. Conf. on Pervasive Computing*, 2002.

[21] J. WALDROP, D. ENGELS, AND S. SARMA. Colorwave: A MAC for RFID reader networks. In *IEEE WCNC*, 2003.

[22] T. ERLEBACH, K. JANSEN, E. SEIDEL, Polynomial-time approximation schemes for geometric graphs, *In Proc. of Annual Symposium on Discrete algorithms*, 2001.

[23] S. E. SARMA, S. A. WEIS, AND D. W. ENGELS, RFID Systems and Security and Privacy Implications," In *Workshop on Cryptographic Hardware and Embedded Systems*. LNCS, vol. 2523, 2002, pp. 454-470.

[24] M. KODIALAM AND T. NANDAGOPAL. Fast and reliable estimation schemes in RFID systems. In *ACM MobiCom*, 2006.