OTrack: Order Tracking for Luggage in Mobile RFID Systems

Longfei Shangguan[1,2], Zhenjiang Li[2,3], Zheng Yang[1,2], Mo Li[3], Yunhao Liu[1,2]

[1]CSE, Hong Kong University of Science and Technology, Hong Kong

[2]School of Software, TNList, Tsinghua University, China

[3]SCE, Nanyang Technological University, Singapore

{longfei, yang, yunhao}@greenorbs.com, {lzjiang, limo}@ntu.edu.sg

Abstract—In many logistics applications of RFID technology, goods attached with tags are placed on moving conveyor belts for processing. It is important to figure out the order of goods on the belts so that further actions like sorting can be accurately taken on proper goods. Due to arbitrary goods placement or the irregularity of wireless signal propagation, neither of the order of tag identification nor the received signal strength provides sufficient evidence on their relative positions on the belts. In this study, we observe, from experiments, a critical region of reading rate when a tag gets close enough to a reader. This phenomenon, as well as other signal attributes, yields the stable indication of tag order. We establish a probabilistic model for recognizing the transient critical region and propose the OTrack protocol to continuously monitor the order of tags. To validate the protocol, we evaluate the accuracy and effectiveness through a one-month experiment conducted through a working conveyor at Beijing Capital International Airport.

I. INTRODUCTION

As a promising technique, Radio Frequency Identification (RFID) systems have been widely adopted to monitor and classify goods and assets in logistic and supply chain managements [1]–[3]. Tens of thousands of goods enter large warehouses each day. Considering the manipulation cost and efficiency, the processing of goods is highly facilitated through the usage of the RFID technique. A typical application is the airports and the most representative example is the *Hong Kong International Airport*. According to [8], the RFID technique at airports has been used to 1) assist the existing bar-code system to improve its reading accuracy, and 2) find an individual's luggage without removing other luggage in vehicles or aircrafts. Apart from those existing services, in this paper, we exploit one new dimension to further benefit the logistic and supply chain managements by the RFID technique, while it is still lacking. Goods enter the warehouse from different entrances. They are processed on conveyor belts and further allocated to different exists. So far as we know, such a task is mainly completed by intensive labor force in existing systems. Due to the large goods volume, the manual processing inevitably incurs goods loss, which leads to tremendous financial losses for customers and industries.

In this paper, the solution that we have envisioned can be illustrated by Fig. 1. Each piece of checked luggage is attached with a passive RFID tag recording the luggage information.

Fig. 1. Illustration of the luggage order tracking

Luggage from multiple counters are gathered to one conveyor belt and there are tens of belts working in parallel. On each belt, the system tracks the order of tags along the conveyor belt and further delivers each piece of luggage at the tail of the belt to the corresponding vehicle. The luggage order is also useful for the system manager to double check certain luggage. Given the crucial role of supply chains to the economy, how to accurately and continuously track tags' order on belts serves as an important component for such mobile RFID systems. If their relative positions are determined incorrectly, luggage could be delivered to undesired places. We also note that not only airports can benefit from such a component. In fact, it provides a generic service that can be used for a variety of other applications, e.g., postal service, logistic delivery, food supply chain, etc.

However, designing such a RFID tracking system entails a wide range of challenges in practice. The communications between the reader and tags abide by the EPC Class 1 Gen 2 RFID standard [4]. This standard is based on a slotted *ALOHA* scheme to regulate the communications. After the reader interrogates a set of tags, tags' responses follow a random sequence to avoid collisions. As a result, the communications themselves provide mere information to infer tags' relative positions on the belt. One possible solution is utilizing the temporary correlation among a series of communications, e.g., based on the tags' sequence entering the communication range of the reader. Yet such a solution may be highly inaccurate.
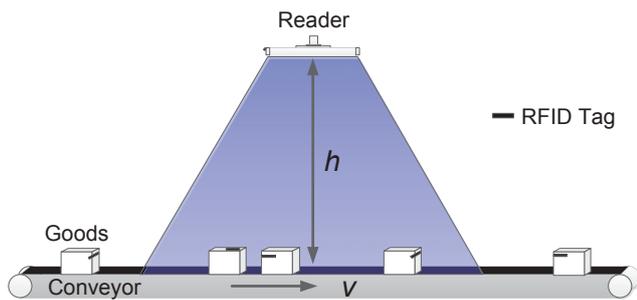
Fig. 2. Tracking the order of tags on a conveyor belt

Fig. 3. Temporary correlation of communications

Due to the hardware heterogeneity and the arbitrary way that luggage is placed on the belt, a tag at the front does not necessarily response to the reader first. Other tags behind may have more sensitive circuits or clearer line-of-sight paths to the reader, leading to the tracking error. Another possible solution is to adopt exiting localization methods [11], [12]. Those methods, however, normally require complicated system deployments (e.g., complex reader or tag arrays) and non-negligible localization inaccuracy (especially in the indoor environments, like warehouses). Due to the space limitation, readers may not be deployed following pattern as required by [11], [12]. In addition, multiple readers can raise the risk to read tags from other belts. On the other hand, as luggage is usually densely placed on the conveyor belt, the localization inaccuracy of even the state-of-the-art methods may cause a significant error in detecting the correct order of tags.

In this paper, we target at a light weight solution tailored to the order tracking problem. We observe that communications between the RFID reader and tags are associated with certain attributes and there exists a strong temporary correlation among those attributes. By taking advantage of such a correlation, we can accurately track the order of tags on conveyor belts in mobile RFID systems. The contributions of this paper are as follows. First, we observe that multiple attributes of the communications between the reader and tags solely do not demonstrate any clear clues. However, by intelligently combining them together, we can obtain a stable indication to determine the physical position of each tag with respect to the reader. We conduct extensive experiments to validate the effectiveness of such a combination by using an ALR-9900+ [6] commercial reader and Alien I2 [7] passive tags. Based on our observation, we then propose the Order Tracking (OTrack) protocol to accurately and efficiently track tags' order on conveyor belts. The proposed protocol is easy to be implemented. To further guarantee the protocol performance, we mathematically analyze the system parameters of OTrack and provide it a set of appropriate settings. We implement OTrack and evaluate its performance through a one-month experiment conducted at Beijing Capital International Airport. The experiment results show that our protocol achieves up to 97% tracking accuracy on average and the protocol is robust to the variance of the belt's velocity.
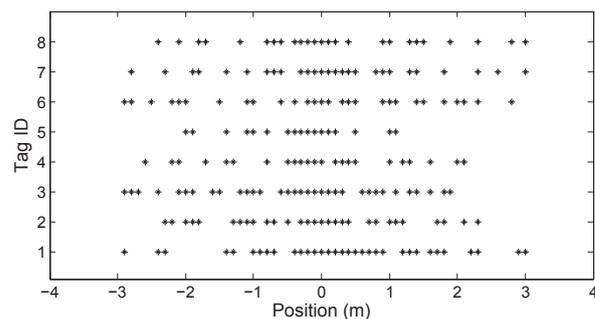
The rest of paper is organized as follows: the problem specification of this paper is presented in Section II. In Section III, we introduce our OTrack protocol and analyze system parameters. The performance evaluation is illustrated in Section IV. We review related works in Section V and finally conclude this paper in Section VI.

II. PROBLEM SPECIFICATION

In this section, we present the formal definition of our luggage order tracking problem.

As shown by Fig. 2, the proposed mobile RFID system consists of three components: a moving conveyor belt with a velocity v, a RFID reader fixed over the belt at a height of h, and a sequence of luggage attached with RFID tags on the belt. Tags will be accessed multiple times during the movement along the belt. The only information that we can use to track the order of tags is the received responses when tags are within the communication range of the reader. We know that due to the randomness from tags' replies, the communications by themselves do not contain any clue to infer tags' relative positions. However, we observe that the communications are associated with multiple attributes (e.g., RSSI, RRR, the temporary sequence to receive each response, etc.). Although in the next section we will show that attributes solely do not provide strong hints to determine their orders either, their combination is actually viable enough for tracking. Thus, the objective to design the OTrack protocol is to intelligently integrate attributes together for obtaining a stable indication, and further explore the implication from such an indication to position each tag on the belts.

III. PROTOCOL DESIGN AND ANALYSIS

In this section, we first present design challenges and our initial attempts. Then, we elaborate the design of OTrack in detail based on the insights obtained from our initial methodologies. Finally, we analyze the setting of system parameters in OTrack.

A. Initial attempts and design challenges

(a) RSSI trend of tag #3   (b) RSSI trend of tag #5   (c) RSSI trend of tag #9
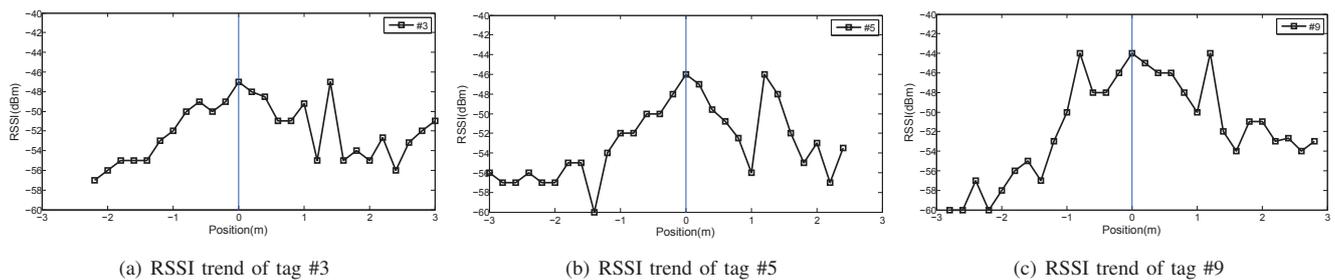
Fig. 4.   RSSI trends of three tags during the movement

1) Utilizing temporary correlation: Our first attempt is to utilize the temporary correlation among successive communications between the reader and tags. More precisely, we rely on the recorded time stamp for each tag entering the communication range of the reader to determine their order on the belt. For the sake of a clear presentation, we denote such a method as *First Come First Sort (FCFS)*. We conduct a 24-hour experiment to examine the performance of FCFS. We equip an ALR-9900+ reader with two Alien ALR-9611-CR antennas over one conveyor belt. The antenna works within the 890∼930 MHz frequency. Each piece of luggage is attached with an Alien I2 passive RFID tag and the velocity of the conveyor belt is $0.4m/s$. We virtually embed an axis along the belt and the **original point** is the vertically projected position of the reader to the belt. Tags are shipped from the negative part to the positive part. To ease the illustration, we place a pressure sensor under the belt at the position $-5m$. When the luggage passes the pressure sensor, the sensor will trigger the system to record the current time then. With this time reference, we can determine each position on the belt[1], at which the tag replies to the reader. For example, we suppose that one piece of luggage passes the pressure sensor when $t = 2s$. If the reader later receives a response from its attached tag when $t = 4s$, we can estimate that the tag replies at position $-5m + (4s - 2s) \times 0.4m/s = -4.2m$.

In Fig. 3, we randomly select 8 consecutive tags to verify the effectiveness of FCFS and Tag #1 is in the front among 8 tags. In Fig. 3, each star represents the position where a tag responses to the reader. From the figure, we can see that different tags reply to the reader within different regions. For instance, tag #5 is within $(-2m, +1.2m)$ yet tag #6 is within $(-3m, +3m)$. In this experiment, the distance between tags #5 and #6 is around $0.6m$ and the FCFS strategy thus causes an ordering error. To quantify the performance of FCFS, we adopt FCFS for one hour to determine the order of 3000 pieces of luggage with the ground truth about their orders. The statistics show that only 55% of luggage has been ordered correctly. It indicates that FCFS fails to track the order with adequate accuracy.

According to our study, we find that the inaccuracy of FCFS is mainly due to the heterogenous circuit sensitivities of different tags and environmental dynamics. A tag might be physically farther away from the reader. However, it could have a more sensitive circuit or clearer line-of-sight path to the reader. In such a case, this tag may reply to the reader earlier than some tags in front of it. Such a challenge prohibits FCFS from being used directly to distinguish the order of tags.

2) Utilizing the RSSI trend: As a tag moves ahead along the belt, its physical distance to the reader decreases first and then increases. Therefore a natural hypothesis is that the detected RSSI at the reader side should follow the same trend. When the tag is near the original point on the belt, the detected RSSI value should be the maximum one. We implement such a greedy method on our test-bed and name it G-RSSI. We examine the effectiveness of G-RSSI by using the experiment of the same setting with Section III-A1.

In Fig. 4, we randomly select 3 tags and depict the RSSI values of their responses to the reader. From Fig. 4, we can see that our hypothesis holds only in a statistical sense. If we take a fine-grained look at the RSSI trace, we will find that the RSSI trace fluctuates significantly such that there are multiple peaks with comparable amplitudes. On the other hand, the trace is not symmetric with reference to the original point due to the temporary lacking of the line-of-sight path to the reader. As a result, it is hard to determine which peak is actually the one when the tag is closest to the original point. The problem can become even worse as the maximum peak may appear when the tag is relatively far away from the reader. As the environmental dynamics disturb the stability of the RSSI trend along the tags' movement, G-RSSI fails to directly capture the order of tags.

3) Utilizing the RRR trend: Although RSSI may fluctuate, it statistically becomes larger when a tag is getting close to the reader. It implies that when the tag is close to the reader, its responses should be with sufficiently high signal strengths and they are prone to be received by the reader successfully. To measure the quality of the response reception[2], for any tag

---

[1]Note that it only works when the ground truth (tags' order) is known in advance. In practice, our protocol does not need pressure sensors since they cannot be used to determine tags' positions on belts without the ground truth.

[2]Since the *ALOHA* protocol can automatically adjust to the best frame size to minimize the number of collision slots, in this paper, we focus on the impact of SNR to the reception of responses merely.
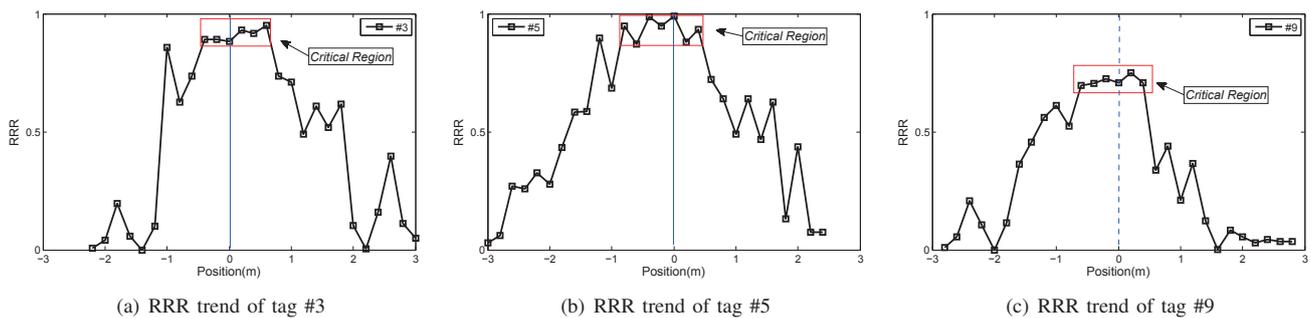
(a) RRR trend of tag #3   (b) RRR trend of tag #5   (c) RRR trend of tag #9

Fig. 5.   RRR trends of three tags during the movement

i, we define its Response Reception Ratio (RRR) as follows:

$$RRR_i = \frac{\text{\# of responses received from } i \text{ in } d}{\text{\# of expected responses from } i \text{ in } d}, \quad (1)$$

where d is a given certain amount of time. According to Eq. (1), we can refer to the RRR trend of tags to determine their relative order, and we name this method *G-RRR*.

We set d to be 0.2s and plot the RRR trends of three tags #3, #5, and #9 in Fig. 5 and find that when a tag just enters the reader's communication range, the RRR is generally low, but as the tag moves ahead, its RRR rapidly increases and stabilizes at a certain value. For example, the RRR value of tag #5 is only 0.1 initially, while its RRR suddenly jumps to 0.6 after it is less than $1.5m$ away from the original point. If we further take a fine-grained look at the high RRR value portion of tag #5, we can observe that within a certain range near the original point, the variance of RRR is quite small, and we can observe similar ranges in the RRR traces of other two tags as well. As a matter of fact, such a region always exists in the RRR trace of each tag and we call such a stable region as *RRR critical region* or **critical region** for short. Fig. 5 implies that the RRR trend does not provide high-granularity location information for tags. It is because the high RRR values stay for a large portion of time when a tag moves along the belt. Similar to previous two solutions, G-RRR cannot directly distinguish the order of tags either.

4) Observed insights: Although the aforementioned attributes cannot be solely used to track the order of tags, we can still obtain three useful observations as follows:

- The critical region for each tag normally covers the original position on the belt.
- RSSI normally exhibits a (local) maximum value when the tag gets close to the original position on the belt.
- Within the critical region for each tag, the trend of the RSSI changing normally demonstrates a concave shape.

Three observations can be more clearly illustrated by Fig. 6. Those observations imply that after determining the critical region for a tag, we can track the trend of its RSSI changing within the critical region. The time stamp when the RSSI peak appears within the critical region can be used to approximate the time of that tag passing the original point on the belt, and
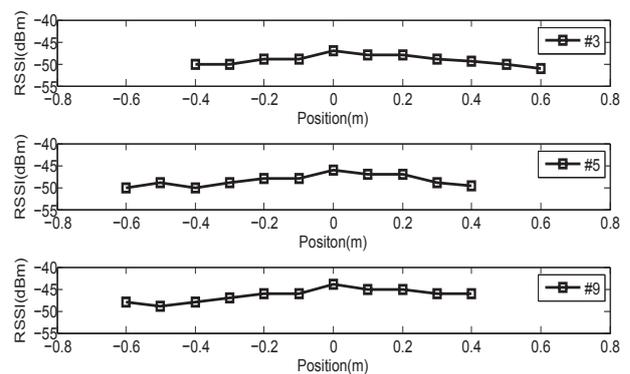

Fig. 6.   Combination of RSSI and RRR

we will refer to such a time stamp to determine the relative positions of each tag. The interpretation of this phenomenon is that after a tag gets sufficiently close to the reader, the communication SNR becomes high enough. The critical region actually indicates that the tag is of good SNR to the reader. As a consequence, the received signal becomes more steady and suffers less impact from surrounding noises.

In the next subsection, we will make use of above observations and design our OTrack protocol.

B. Protocol specification

From Fig. 5, we have found that for different tags, RRR might be significantly different in their own critical regions. When a tag has a clear line-to-sight path to the reader, RRR is more than 90% in the critical region. Nevertheless, RRR can be only around 60% if the tag is blocked by the luggage during the movement. Therefore, in practice, we hardly rely on any pre-defined threshold to detect the critical region. In OTrack, we propose to examine the consistence of the RRR values among consecutive periods . When a period is outside the critical range, its RRR value exhibits significant difference compared with neighboring periods. On the contrary, the RRR values within a critical range are sufficiently close to each other. Therefore, for each tag, we measure the closeness of RRR values and search for a range containing most consecutive periods with a minimal variance. Such a range will be considered as the critical region for the tag

Algorithm 1: *criticalRegionSearch(i)*

Input : *is_win_size_increasing = False*;

Output: w_i^* containing the critical region of tag *i*.

1 **if** *is_win_size_increasing = True* **then**

2 Additively prolong the window size of w_i^*;

3 **if** w_i^* *fails to pass Lemma 1* **then**

4   return w_i^*;

5 **else**

  Create a new window $w_{t,k}^i$ and insert it into the window set W_i;

6 **for** *each window $w_{t,k}^i$ in W_i* **do**

7   **if** $t + |w_{t,k}^i| == k$ **then**

8   **if** $w_{t,k}^i$ *passes Lemma 1* **then**

9   Mark *is_win_size_increasing* to be *True*;

10 $w_i^* \leftarrow w_{t,k}^i$;

11   Delete all other windows in W_i;

12   **else**

     Delete $w_{t,k}^i$;

13 return *NULL*;

Algorithm 2: The OTrack Protocol

Input : *Identified tag set S*, initially, $S = \varnothing$;
      Critical region set S^*, initially, $S^* = \varnothing$;

Output: The ordered tag sequence;

1 **while** *Broadcasting a beginning round command* **do**

2 **if** *a new tag is detected* **then**

3   Inserting it into *S*;

4 **for** *each tag i in S* **do**

5 $S^* \leftarrow criticalRegionSearch(i)$;

6 **for** *each tag i in S^** **do**

7   Performing the *Quadratic Fitting* technique on the window w_i^* of tag *i*;

8   Obtaining the timestamp t_i^* when RSSI peak appears;

9   Conducting the *Insertion Sorting* technique to order the sequence of tags in S^*;

10 Report the ordered tag sequence;

---

in OTrack. To accurately quantify such closeness, we avoid using any threshold-based heuristic (e.g., $\leq \pm5\%$ of a baseline value) to ensure the detection accuracy. Instead, we utilize the *central limit theory* to provide a more precise detection. After determining the critical region, we further explore the RSSI peak and obtain the time when the peak appears. As the collected RSSI values are usually mixed with noises, we use the *quadratic fitting* technique [5] to minimize the influence from noises. With the above two steps, we can obtain an accurate time reference for each tag such that their relative positions can be ordered on the belt. To formally describe our protocol, we introduce several notations at first:

- o_t^i is a *period* starting from time *t* for tag *i*. $|o_t^i|$ indicates the length of o_t^i, in terms of *seconds*.
- $w_{t,k}^i$ is the *k*-th *window* for tag *i* starting from time *t*. $|w_{t,k}^i|$ indicates the length of $w_{t,k}^i$, in terms of *periods*.
- $p_{w_{t,k}^i}^j$ is the RRR in the *j*-th period of $w_{t,k}^i$, $1 \leq j \leq |w_{t,k}^i|$.
- $\overline{p}_{w_{t,k}^i}$ is the *effective* average RRR over $w_{t,k}^i$.

Given a period, we can calculate RRR for a tag *i*. If the starting time of the period is *t*, we denote such a period as o_t^i. Then by grouping several consecutive periods, we form a window. As we may create multiple windows for a same tag in our design, we use $w_{t,k}^i$ to denote the *k*-th window of tag *i*, where *t* is the starting time of the first period in this window. Within a window $w_{t,k}^i$, the reader might receive no response in certain periods, i.e. RRR is zero in those periods. It is usually true when the tag is far away from the reader. Then, we introduce $s_{t,k}^i$ for $w_{t,k}^i$ to denote the periods, in which RRR is greater than zero. Based on those definitions, an instrumental explanation of our design is as follows. For any window $w_{t,k}^i$,

we define its effective average RRR as:

$$\overline{p}_{w_{t,k}^i} = \sum_{j \in s_{t,k}^i} p_{w_{t,k}^i}^j / |s_{t,k}^i|, \qquad (2)$$

where $p_{w_{t,k}^i}^j$ represents the RRR value within period *j* of $w_{t,k}^i$. In this study, we find that by statistically comparing $|w_{t,k}^i| \times \overline{p}_{w_{t,k}^i}$ with $|s_{t,k}^i|$, we can conclude whether $w_{t,k}^i$ is completely within the critical region of tag *i*. The basic principle is that those two values should be sufficiently close to each other if $w_{t,k}^i$ is a subset of a critical region. A more formal specification is given by the following lemma.

Lemma 1: Let $\overline{p}_{w_{t,k}^i}$ be the effective average RRR over $w_{t,k}^i$. Then $w_{t,k}^i$ is completely within the critical region with probability $\emptyset(\alpha)$ if $P\{||s_{i,j}| - |w_{t,k}^i| \cdot \overline{p}_{w_{t,k}^i}| \leq \alpha\sqrt{|w_{t,k}^i| \cdot \overline{p}_{w_{t,k}^i} \cdot (1 - \overline{p}_{w_{t,k}^i})}\}$ holds, where $\emptyset(\alpha)$ is determined via the *Standard Gaussian Distribution Chart* [9].

Due to the page limitation, the proof of Lemma 1 is given in our technical report [10]. In Lemma 1, the parameter α is crucial to the accuracy of the critical region detection. If α is large, the inequality in Lemma 1 is easy to hold, while a window $w_{t,k}^i$ is more likely to be mistaken as a part of the critical region. On the contrary, if α is too small, it is hard for the inequality in Lemma 1 to be satisfied. As a direct consequence, the critical region fails to be properly identified. Either case degrades the accuracy of OTrack. In Section IV, we will investigate how to select the parameter α. After α has been properly chosen, it indicates that $w_{t,k}^i$ is within the critical region with a high probability when the inequality holds. If $w_{t,k}^i$ is within the critical region, we can gradually increase its window size such that the final $w_{t,k}^i$ will cover the entire critical region of tag *i*. After figuring out the critical region, we will further apply the quadratic fitting technique to obtain the time when the RSSI peak appears. The order of tags can
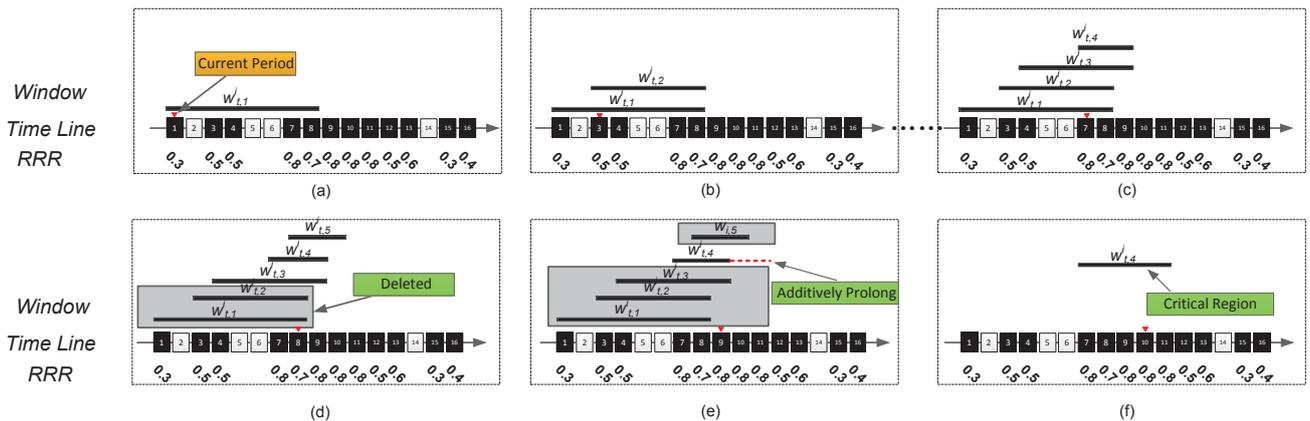
Fig. 7.   Running example of Algorithm 1

be determined based on such a series of time references. The detailed critical region searching protocol and the complete OTrack protocol are given by Algorithm 1 and Algorithm 2, respectively.

The interpretation to operations of Algorithm 1 is as follows. When the reader receives a response from a tag i for the first time at time t, it will create the first window $w_{t,1}^i$ for this tag. In the periods afterwards, if RRR is not zero, the reader will generate a new window $w_{t,k}^i$. By so doing, we will not miss the window aligned with the starting point of the critical region. As time elapses, when the end of a window is reached (line 7 in Algorithm 1), we examine whether this window is a part of the critical region. If the inequality in Lemma 1 holds, this window is a part of the critical region. Meanwhile, it has the same starting point with the critical region. We then gradually increase its window size until the inequality in Lemma 1 becomes invalid and the final window covers the entire critical region of tag i. The whole process can be illustrated by an example shown in Fig. 7.

In the example shown in Fig. 7, along the time line, each white square indicates a period without responses received from tag i. In contrast, one black square represents that the reader receives responses in this period and the number below is the corresponding RRR. When the reader receives responses from tag i for the first time in period 1, it generates $w_{t,1}^i$ as shown by Fig. 7 (a). Later, new windows will be generated if periods are of the black color (Figs. 7 (b) and (c)). In period 8, we reach the ends of $w_{t,1}^i$ and $w_{t,2}^i$. Since the inequality in Lemma 1 does not hold for those two windows, they are deleted as shown by Fig. 7 (d). In period 9, as depicted by Fig. 7 (e), the ends of $w_{t,3}^i$ and $w_{t,4}^i$ are reached. Since only $w_{t,4}^i$ passes the verification of Lemma 1, $w_{t,3}^i$ will be deleted and $w_{t,4}^i$ will be gradually increased in the following periods. Note that at this time, we can delete $w_{t,5}^i$ as well. Although $w_{t,5}^i$ is also within the critical region of tag i, its final size will be shorter than $w_{t,4}^i$ and we prefer a longer one. Eventually, the inequality in Lemma 1 becomes invalid for $w_{t,4}^i$ in period

12 and detected critical region is illustrated in Fig. 7 (f).

After the critical region of tag i is determined by Algorithm 1, the OTrack protocol will sort the order of tag i together with other tags whose critical regions are already determined as shown by Algorithm 2, after which OTrack will output the final order of the passing tags (as well as the luggage they are attached on).

C. Protocol analysis

1) Window size configuration: In the OTrack protocol, multiple windows might be constructed for each tag. We find that their window sizes cannot be determined arbitrarily. If a window $w_{t,k}^i$ is far away from the original point (i.e., tag i is far away from the reader), response reception ratios in this window can exhibit differently. In this case, the window size should be large such that we can observe sufficient RRR heterogeneity and confirm the window outside the critical region. On the other hand, if the tag is close to the reader, the window size should be relatively small. A large window size, in such a case, may cover extra portions outside of the critical region. Either case can cause inaccuracy of the critical region detection. In OTrack, we adopt a simple method to cope with such an issue as follows. The window size is set to guarantee that a clean response from tag i can be received at least once with a high probability in this window. The rationale behind is that if tag i is far away from the reader, the RRR values are low in general and the window size should be large; Otherwise, the window size should be small. The detailed window size setting in OTrack is given by Lemma 2.

Lemma 2: Let $p_{w_{t,1}^i}^j$ be the RRR in the first period of $w_{t,k}^i$. A response can be successfully received at least once from tag i with probability $> 1 - \eta$ if the window size is $|w_{t,k}^i| \geq \lceil ln(1/\eta)/p_{w_{t,1}^i}^j \rceil$.

Due to the page limitation, the proof of Lemma 2 is given in our technical report [10]. In OTrack, the default value of η is set to be 0.05 and Lemma 2 indicates that the window size $\lceil ln(1/\eta)/p_{w_{t,1}^i}^j \rceil$ is large enough to guarantee that a clean

Fig. 8. Accuracy Ratio vs. x

Fig. 9. Accuracy Ratio vs. Period
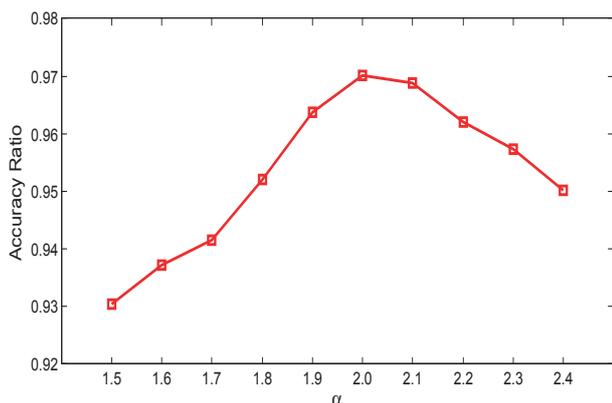
response can be received at least once from tag i with a high probability. By so doing, we guarantee large enough window sizes for those low RRR regions and small enough window sizes for those high RRR regions.

2) Quality of the critical region detection: Lemma 1 in the previous subsection states that the RRR values of different periods within a critical region should be sufficiently similar to each other, and we rely on such a conclusion to detect the critical region for each tag. In practice, however, the RRR values in a critical region might still be relatively different. Therefore, we want to examine how likely such a phenomenon occurs. In Lemma 3, we find that in principle the RRR values cross a critical region can possibly exhibit a large variance, the probability of its occurrence , however, is extremely small and bounded from above.

Lemma 3: We introduce δ to measure the difference between $||s_{i,j}| - |w_{t,k}^i| \cdot \overline{p}_{w_{t,k}^i}|$. The probability $P\{||s_{i,j}| - |w_{t,k}^i| \cdot \overline{p}_{w_{t,k}^i}| \geq \delta \mid w_{t,k}^i$ is within a critical region$\}$ is $\leq 2\exp(-\delta^2/2 \cdot |w_{t,k}^i| \cdot \overline{p}_{w_{t,k}^i})$.

Due to the page limitation, the proof of Lemma 3 is given in our technical report [10]. Lemma 3 indicates that the RRR values are not likely to exhibit a large variance. As a result, the inequality in Lemma 1 can serve as reasonably good criterion to determine whether a window is completely within the critical region for OTrack.

IV. EXPERIMENTAL EVALUATION

In the previous section, we have elaborated the design detail and parameter configuration of OTrack. In this section, we evaluate its performance through extensive experiments in practice.

A. Experiment Setting

We implement OTrack on a workstation equipped with an Intel Core i7 CPU (2.93GHz) and an 8GB RAM. To evaluate its performance in practice, we conduct the experiment on a testing conveyor belt located at Terminal 1 of Beijing Capital International Airport with 10000 pieces of luggage. Each
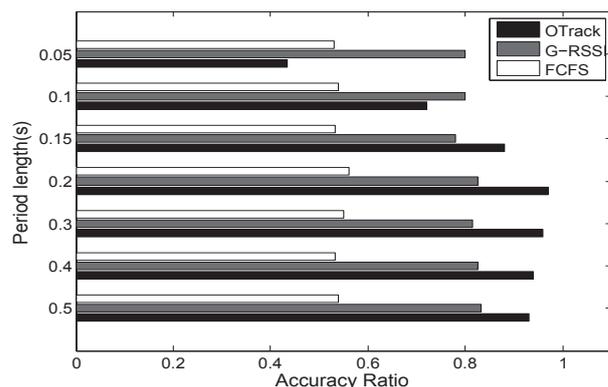
luggage is attached with an passive RFID tag. We equip an ALR-9900+ reader with Alien ALR-9611-CR antennas and place it over the conveyor belt. The vertical distance between the belt and the reader is 1.75 meters and the communication range of the reader is around 7 meters on the belt. All the antennas work within 890~930 MHz. To better understand the protocol performance, we conduct a comparison study for OTrack with other two practical ones, FCFS and G-RSSI, in the experiment. FCFS, as we have explained in Section II, relies on the recorded time stamp (by the reader), when each tag enters the reader's communication range, to determine tags' relative positions on the belt. G-RSSI is a greedy algorithm that decides the order of tags based on the time stamp when each RSSI peak from tags' responses appears. To quantify the performance of each protocol, we mainly refer to the metric *Accuracy Ratio* defined as follows:

$$\text{Accuracy Ratio} = \frac{\text{\# of tags ordered correctly}}{\text{Total \# of tags shipped on the belt}}.$$

B. Experiment Results

1) Investigation on α in Lemma 1: Fig. 8 depicts the performance of OTrack with various α settings. As mentioned before, α is crucial to the accuracy of the critical region detection. If α is large, the inequality in Lemma 1 is easy to hold, while a window $w_{t,k}^i$ is prone to be mistook as a part of the critical region. On the contrary, if α is too small, it is hard for the inequality in Lemma 1 to be satisfied. As a direct consequence, the critical region fails to be properly identified. Either case degrades the accuracy of OTrack. To explore an appropriate setting of the system parameter α, we examine ten representative values of α in this experiment. As shown by Fig. 8, the trend of the accuracy ratio exhibits a concave shape as we vary α. When α is around 2, the curve reaches the peak value and the corresponding accuracy ratio is as high as 0.97. The accuracy ratio drops when we either increase or decrease α. Suggested by Fig. 8, we configure α to be 2 in the following experiments.

TABLE I
DIFFERENT LEVELS OF WORKLOADS

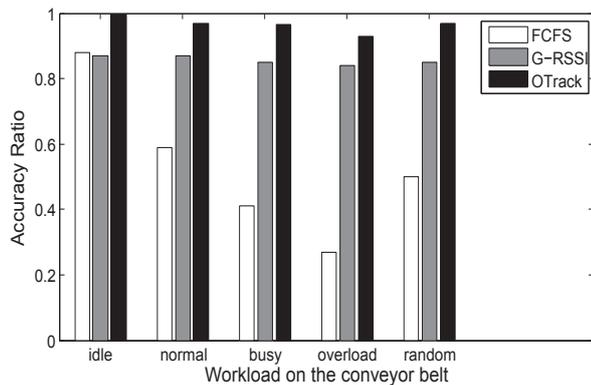| Names of settings	Distance lengths
Idle	$1.0m$
Normal	$0.5m$
Busy	$0.3m$
Overload	$0.2m$
Random	$[0.2m, 1.0m]$

Fig. 10.   Accuracy Ratio vs. Luggage distance

2) Accuracy Ratio vs. period length: According to the definition, the RRR value is a statistic result calculated within one period. The length of the period thus needs to be carefully selected. Otherwise, the obtained RRR value is not stable enough for the critical region searching. If the period is too short, the randomness from the environmental dynamics cannot be completely eliminated. It will cause the inaccuracy to the critical region detection, and thus deteriorates the overall performance of OTrack. On the other hand, the length of the period should not be too large either. Since a window is composed of consecutive periods and the critical region is finally indicated by a window in OTrack, the granularity of the detected critical region will not be high if the period length is too large, which may also impact the accuracy of our protocol.

In Fig. 9, we vary the period length from $0.05s$ to $0.5s$ to examine its impact. As expected, the accuracy of OTrack is poor when the period is short. In particular, the accuracy ratio is only 0.42 when the period length is set to be $0.05s$. As the period length increases, the accuracy ratio of OTrack increases dramatically. When the period length is $0.2s$, the accuracy ratio is up to 0.98. In Fig. 9, we observe that if we further increase the period length, OTrack's accuracy ratio starts to degrade. On the other hand, as both FCFS and G-RSSI protocols do not rely on RRR to determine the order of tags, their performance remains stable cross different period lengths. However, OTrack with a proper period length can outperform those two protocols. From statistics, the performance improvements of OTrack over FCFS and G-RSSI are 40%+ and 20%+, respectively.

3) Accuracy Ratio vs. luggage distance: In this subsection, we investigate the performance of three protocols as we vary the distance between two (successive) pieces of luggage. Since such a distance implies the luggage density and the shipping workload on the belt, to facilitate the presentation, we utilize terms, like *idle*, *busy*, *overload*, etc., to name several typical distance settings in Table I. From Fig. 10, we can see that when the luggage load is low, all three protocols can achieve high accuracy ratios. In particular, the accuracy ratio of OTrack is close to 1.0. This is because the interference from neighboring tags is weak when tags are separated adequately apart from each other. However, with the consideration of the shipping efficiency in practice, luggage cannot be sparsely placed on belts. Fig. 10 shows that when luggage load increases, the performance of all three protocols deteriorate. Compared with FCFS, both OTrack and G-RSSI only slightly decrease. Yet FCFS suffers from a significant dive to 0.3 in the overload scenario. In the same scenario, however, G-RSSI performs with an accuracy ratio smaller than 0.8 and OTrack even achieves as high as 0.93 accuracy. Fig. 10 indicates that our proposed protocol can effectively handle various shipping workloads in practice.

V. RELATED WORK

We review two categories of research works that are directly related to our work.

Communication-based localization techniques: A variety of protocols have been proposed for location using the RFID technique or the sensing technique. *SpotON* [11] is a pioneer RFID localization system, which employs RFID Reader and a batch of active tags for the indoor localization. Ni. et al. later propose *LANDMARK* [12]. It uses two different types of RFID tags (reference tag and tracking tag) for the object tracking. *TASA* [15] relies on RFID tag Arrays for location sensing and frequent route detection. Liu et.al [13] propose to use RF tag arrays for activity monitoring, so as to finding frequent trajectory pattern of human beings. Through our study, we find that those existing works cannot be applied to address our problem. On the conveyor belts, luggage is normally close to each other, e.g. $< 0.5m$. On the other hand, all such localization systems introduce high deployment overhead, which makes it difficult to apply in the applications, such as airport, postal services, food supply chain, etc.

On the other hand, there also exists plenty of researches that focus on wireless channel characters for localization. Represented works include [19]–[23]. In [19], the authors survey the localization and localizability techniques in wireless networks and put forward several open questions in this area. [20] proposes an indoor localization technique by utilizing user movement pattern and WiFi signal trace matching. Yang et.al [21] consider the node localizability problems in wireless sensor netoworks, which plays a key role in understanding the localization performance for wireless networks. Further, they leverage user motions to construct the radio map of a

floor plan, which significantly reduce the cost of fingerprinting database construction [22]. Wu et. al [23] leverages Channel Impulse Response (CIR) of wireless channel to localize humans in indoor environment. In our application scenario, however, due to the device heterogeneity and the environmental dynamics, the approaches aforementioned would cause a significant detection error in practice, which hardly satisfies the application requirements.

Performance optimization for RFID system: Another relevant topic to our work is the performance optimization in mobile RFID systems. Xie et al. in [3] consider improving the reading efficiency for RFID tags along a moving conveyor belt. Yang et al. [18] present an identification-free authentication protocol for efficiently pinpointing counterfeit tags. Tan [2] considers the automatic RFID-based detection for missing-tag events. Qian [14] proposes a scheme that provides low latency RFID identification and has stable performance for massive RFID networks. There are also plenty of related works in Database area. Jeffery [16] proposes an adaptive RFID middleware to identify tag motions in RFID system. Tran et al. in [17] address the problem of translating noisy, incomplete raw streams from mobile RFID readers into clean, precise event streams with the location information. Tong [24] consider to extract frequent items from probabilistic data set, which are useful to solve counting and classification problems in mobile RFID system. Generally, although those existing protocols or algorithms may complement our our design to further improve its efficiency, they are not designed for the tracking purpose, thus apart from the research focus of this paper.

VI. Conclusion

In this paper, we study how to design a mobile RFID system to track the order of tags on conveyor belts. Although the communications between readers and tags cannot be directly utilized to determine the relative position of tags on belts, we observe that the combination of multiple attributes of the communications serves as a viable way to achieve such a goal. To translate our observation to a practical protocol, we propose OTrack. OTrack can intelligently integrate attributes of communications such that the order of tags can be accurately tracked. To guarantee the performance of OTrack, we further mathematically analyze and properly set system parameters. Over one-month experiment conducted at Beijing Capital International Airport demonstrates the accuracy and effectiveness of our design.

VII. Acknowledgement

References

[1] B. Sheng, C. C. Tan, Q. Li, and W. Mao, Finding Popular Categories for RFID Tags, In *Proc. of MobiHoc*, 2008.

[2] C. C. Tan, S. Bo, and Q. Li, How to Monitor for Missing RFID tags, In *Proc. of ICDCS*, 2008.

[3] L. Xie, B. Sheng et al., Efficient Tag Identification in Mobile RFID Systems, In *Proc. of Infocom*, 2010.

[4] EPC Class 1 Gen 2 RFID standards, http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2/uhfc1g2_1_2_0-standard-20080511.pdf

[5] W. Sun, Y. Yuan, Optimization Theory and Methods: Nonlinear Programming, Springer; Softcover reprint of hardcover 1st ed. 2010.

[6] http://www.alientechnology.com/readers/

[7] http://www.alientechnology.com/tags/

[8] http://www.hongkongairport.com/gb/

[9] http://www.mathsisfun.com/data/standard-normal-distribution-table.html

[10] L. Shangguan, Z. Li, Z. Yang, M, Li, Y. Liu, Towards Order Tracking for Tags in Mobile RFID Systems, *Technical Report* http://long.srfid.org/files/2413/4339/3051/OTrack.pdf

[11] J. Hightower, R. Want and G. Borriello, SpotON: An indoor 3D location sensing technology based on RF signal strength, *UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering*, Seattle, 2000.

[12] L. Ni, Y. Liu, Y.C. Lau, and A.P. Patil, LANDMARC: indoor location sensing using active RFID, In *Proc. of PerCom*, 2003.

[13] Y. Liu, Y. Zhao, L. Chen, J. Pei, J. Han, Mining Frequent Trajectory Patterns for Activity Monitoring Using Radio Frequency Tag Arrays, *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, Vol 23, No. 11, Pages 2138-2149, November 2012.

[14] C. Qian, Y. Liu, H. Ngan, and L. Ni, ASAP: Scalable Arbitration for Contactless RFID Systems, In *Proc. of ICDCS*, 2010.

[15] D. Zhang, J. Zhou, M. Guo, J. Cao, T. Li, TASA: Tag-Free Activity Sensing Using RFID Tag Arrays, *IEEE Transactions on Parallel and Distributed Systems, (TPDS)*, Vol 22, No. 4, Pages 558-570, April 2011.

[16] S. Jeffery, et al., An adaptive RFID middleware for supporting meta-physical data independence, In *Proc. of VLDB*, 2007.

[17] Thanh Tran, Charles Sutton, Richard Cocci et al., Probabilistic Inference over RFID Streams in Mobile Environments, In *Proc. of ICDE*, 2009.

[18] L. Yang, J. Han, Y. Qi, Y. Liu, Identification-Free Batch Authentication for RFID Tags, In *Proc. of ICNP*, 2010.

[19] Y. Liu, Z. Yang, X. Wang, L. Jian, Location, Localization, Localizability, Journal of Computer Science and Technology (JCST), 25(2): 274-297, Mar, 2010.

[20] C. Wu, Z. Yang, Y. Liu, W. Xi, WILL: Wireless Indoor Localization Without Site Survey, In *Proc. of INFOCOM*, 2012.

[21] Z. Yang, Y. Liu, Understanding Node Localizability of Wireless Ad-hoc Networks, *IEEE Transactions on Mobile Computing (TMC)*, Volume 11, Issue 8, Pages 1249-1260, August 2012.

[22] Z. Yang, C. Wu, Y. Liu, Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention, In *Proc. of Mobicom*, 2012.

[23] K. Wu, J. Xiao, Y. Yi, M. Gao, L. Ni, FILA: Fine-grained indoor localization, In *Proc. of INFOCOM*, 2012.

[24] Y. Tong, L. Chen, Y. Cheng, P. Yu, Mining Frequent Itemsets over Uncertain Databases, In *Proc. of VLDB*, 2012.