

OTrack: Towards Order Tracking for Tags in Mobile RFID Systems

Longfei Shangguan, *Student Member, IEEE*, Zhenjiang Li, *Member, IEEE*, Zheng Yang, *Member, IEEE*, Mo Li, *Member, IEEE*, Yunhao Liu, *Senior Member, IEEE*, and Jinsong Han, *Member, IEEE*

Abstract—In many logistics applications of RFID technology, luggage 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.

Index Terms—Mobile RFID system, order tracking, probabilistic model

1 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], [2], [3]. Tens of thousands of goods enter large warehouses each day. Considering the manipulation cost and efficiency, the processing of goods is already facilitated through the usage of the RFID technique. A typical application in logistics industry is the airports and the most representative example is the *Hong Kong International Airport*. According to [9], 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 airport luggage sorting system by employing RFID technique. Luggage enter the luggage sorting pool from different entrances. They are identified by an RFID reader and further allocated to different flight according to individual destination. So far as we know, such a task is

mainly completed by intensive labor force or expensive rfid-based systems, they suffer either high error sorting rate or heavy deployment cost.

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. Luggage from multiple counters are gathered to one conveyor belt for sorting. Once been identified by the RFID reader, the luggage will be labeled and ordered. Basing on such order information, the luggage sorting system shunts luggage to the corresponding vehicle at the tail of this conveyor belt. Apart from this, luggage order is also useful for the system manager to track or double check certain luggage. Given the crucial role of luggage sorting to the civil aviation industry, how to accurately and continuously track luggage' order on belts serves as an important component for such RFID-based luggage sorting systems. If their relative positions are determined incorrectly, luggage could be delivered to undesired places. In essence, tracking the order of luggage is equivalent to tracking the order of moving RFID tags, and 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

- L. Shangguan was a research associate at School of Software, Tsinghua University during this study. He is now at CSE Department, Hong Kong University of Science and Technology, Kowloon, Hong Kong. E-mail: longfei@greenorbs.com.
- Z. Li and M. Li are with the School of Computer Engineering, Nanyang Technological University, Singapore 639798. E-mail: {lzjiang, limo}@ntu.edu.sg.
- Z. Yang and Y. Liu are with the School of Software and TNLIST, Tsinghua University, Beijing, China. E-mail: {zheng, yunhao}@greenorbs.com.
- J. Han is with the School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China. E-mail: hanjinsong@mail.xjtu.edu.cn.

Manuscript received 19 Jan. 2013; revised 6 Sept. 2013; accepted 17 Sept. 2013. Date of publication 8 Oct. 2013; date of current version 16 July 2014. Recommended for acceptance by D. Simplot-Ryl. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TPDS.2013.247



Fig. 1. Illustration of the luggage order tracking.

entering the communication range of the reader. Yet such a solution may be highly inaccurate. Due to the hardware heterogeneity and the arbitrary way that luggage is placed on the belt, a tag at the front does not necessarily respond 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 existing localization methods [11], [12], [13]. 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 the pattern as required by [11], [12], [13]. 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 exist 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+ [7] commercial reader and Alien I2 [8] 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 implement. To further guarantee the protocol performance, we mathematically analyze the system parameters of OTrack and provide 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 percent 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 related works of this paper is presented in Section 2. In Section 3,

we introduce the background of our work. The OTrack protocol and system parameter analysis are introduced in Section 4. We illustrate the system evaluation in Section 5 and finally conclude this paper in Section 6.

2 RELATED WORK

Since the emergence, the RFID-based luggage sorting system has been widely deployed in newly-built or updated airports such as Aalborg Airport in Denmark, McCarran International Airport in U.S. and Zaventem airport in Brussels etc. Although automatically, these systems require that all luggage should be physically separated (usually via a carrying case or a luggage try) so as to accurately capture the order of checked luggage. As a result, a great portion of conveyor space is occupied by carrying cases, and sorting efficiency is therefore maintaining in a low level. Different from these systems, our solution targets at tracking the order of luggage without the requirement of physical isolation of checked luggage. Therefore considerably improves the working efficiency of the sorting system. In research area, our concept of order tracking for tags is grounded on both localization and mobile RFID systems. In what follows, we broadly review two categories of research works that are directly related to our work.

2.1 Communication-Based Localization Techniques

The proliferation of wireless devices has fostered the demand for context-aware applications, in which location is viewed as one of the most significant contexts. A variety of protocols have been proposed for localization using the RFID technique or the sensing technique. *SpotON* [11] is a pioneer RFID localization system, which employs received RSSI value as a sensor measurement for estimating inter-tag distance. It adopts aggregation methods for object localization. Ni *et al.* later propose *LANDMARC* [12]. *LANDMARC* uses K-Nearest Neighbor (KNN) to give weight matrix, and integrates the nearest coordinates for location estimation. The system accuracy is around 1 ~ 2 meter. *VIRE* [13] overcomes the unstable issue of RSSI by introducing the concept of virtual reference tags. Specifically, four reference tags form a grid in *VIRE*, and some reference tags will be interpolated in the grid. Then a proximity map is obtained after collecting the results by the RFID readers. Compared with *LADNMARC*, *VIRE* achieves much higher localization accuracy at the cost of more computational overhead. To further reduce the localization error, *Cricket* [14] adopts ultrasound time-of-flight measurement technique to provide location information, which could accurately capture the fine-grained location of each object. *PinPoint* [15] employs the Time-of-Arrival (ToA) information of multiple radio signals for location estimation and compensates the clock difference among each nodes via mathematical approach. Through our study, we find that those existing systems cannot be applied to our problem. At first place, luggage are normally put closely to each other on the moving conveyor, e.g., < 0.5 m. The existing approaches, however, could not provide enough resolution to pinpoint their locations. Besides, all of these localization systems introduce high deployment overhead (i.e., special hardware such as high-accuracy ultrasonic generator/receiver, angle measurement instruments), which

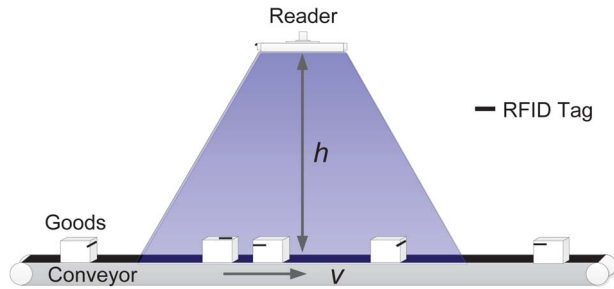


Fig. 2. Tracking the order of luggage on a conveyor belt.

makes it difficult to apply in pervasive applications, such as airport, postal services, food supply chain, etc.

On the other hand, there is research that relies on a set of collected wireless fingerprints for the localization. Represented works include [16], [17], [18], [19]. In [16], the authors compare a suite of wireless-radio-based positioning techniques, and propose a wide-area 802.11-based localization system. It works by having a client device listen for radio beacons in its environment and uses a pre-computed map of radio sources in the environment to localize itself. The authors in [17] propose to achieve an organic indoor localization by considering the error and uncertainty of the input data when constructing the fingerprint database. Bhasker in [18] presents a method for user-assisted location based on wireless signal strengths. Such method is self-maintaining with respect to changing access point deployments. SurroundSense [19] performs a logical location estimation based on ambience features including sound, light, color and WiFi signal in the indoor environments. In our application scenario, however, due to the device heterogeneity and the environmental dynamics, such fingerprinting based approaches cause a significant detection error in practice, which hardly satisfies the application requirements.

2.2 Mobile RFID Systems

Another relevant topic to our work is the performance optimization in mobile RFID systems. Xie *et al.* in [2] consider improving the reading efficiency for RFID tags along a moving conveyor belt. Basing on the probabilistic model for RFID tag identification, the authors propose a dynamic programming-based solution to select optimized frame sizes to successfully identify moving RFID tags. In [20], Welbourne *et al.* design and implement an RFID-based pervasive computing applications with an infrastructure for specifying, extracting and managing meaningful high-level events from raw RFID data in mobile environment. Tran *et al.* in [21] address the problem of translating noisy, incomplete raw streams from mobile RFID readers into clean, precise event streams by building a probabilistic model which captures the mobility of the reader, object dynamic and noisy readings. [3] considers the automatic RFID-based detection for missing-tag events. They propose a monitoring method which does not require the reader to collect IDs from each RFID tag but is still able to answer whether there is missing tags. Later in [22], they further design a protocol that allows a RFID reader to pinpoint exactly which tags are missing. Yang *et al.* [23] present an

identification-free authentication protocol for efficiently pinpointing counterfeit tags by employing collision analysis techniques. Although those existing protocols may complement our design to further improve its efficiency, they are not designed for the tracking purpose, thus apart from the research focus of this paper.

3 BACKGROUND

In this section, we first briefly introduce the communication process between RFID readers and tags, and then present the problem specification of this paper.

3.1 Communication Process between Readers and Tags

OTrack abides by the Class-1 Generation-2 (C1G2) MAC-layer standard [4], which utilizes a *slotted ALOHA* [5] scheme to regulate the communication. In this standard, the communication procedure between a reader and multiple tags is composed of frames and each frame is further divided into time slots. When a reader wants to access tags, it first broadcasts a *beginning round* command containing the frame size f , indicating that one frame consists of f time slots. Upon receiving the command, each tag will randomly select one slot to reply a response to the reader. As tags independently select the slot for replying to the reader, there are three types of slots from the reader's perspective. First, there is no response from tags in a slot, which is termed as an *idle* slot. Second, the reader receives the response from only one tag. In this case, the reader can successfully decode the response when the signal strength is strong enough. We denote such a response as a *clear* response. Third, when more than one tag replies in a slot, a collision occurs (i.e., a *collision* slot) and the reader cannot distinguish those responses. Therefore, the tracking mainly relies on the clear responses from each tag.

3.2 Problem Specification

As shown in Fig. 2, the proposed mobile RFID system consists of three components: a moving conveyor belt with a velocity v , an 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. From the communication paradigm mentioned above, we can see that due to the randomness of 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., Received Signal Strength Indicator (RSSI), Reading Reception Ratio (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.

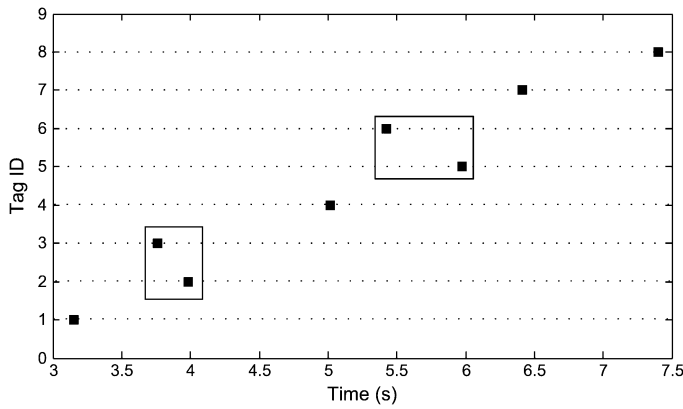


Fig. 3. Temporary correlation of communications.

4 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.

4.1 Initial Attempts and Design Challenges

4.1.1 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 first identified time stamp of each tag 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 conducted 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 0.4 m/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.

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, followed by tag #2, #3 ... until tag #8. In Fig. 3, each black square represents the corresponding time stamp when each tag is identified for the first time. From the figure, we can see that FCFS identifies tag #4 before identifying tag #5, the similar phenomenon also happens on tag #5 and tag #6, which 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 percent 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.

4.1.2 Utilizing the RSSI Trend

As a tag moves 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 4.1.1.

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 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 lack 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.

4.1.3 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 reception¹, for any tag 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 and we term such amount of time as *period* in this paper. According to eq. (1), we can refer to the RRR trend of tags to determine their relative order, and we name this greedy method *G-RRR*.

We set d to be 0.2 s 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 1.5 m 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

1. 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 signal-to-noise ratio (SNR) to the reception of responses merely.

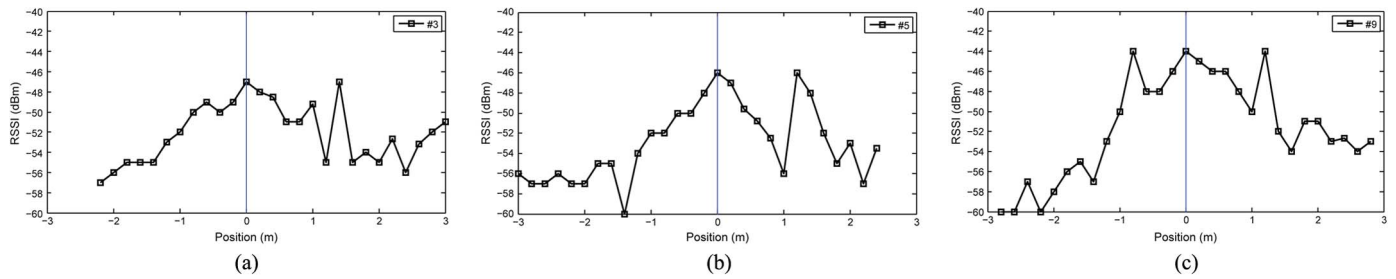


Fig. 4. RSSI trends of three tags during the movement: (a) RSSI trend of tag #3; (b) RSSI trend of tag #5; and (c) RSSI trend of tag #9.

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.

4.1.4 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 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.

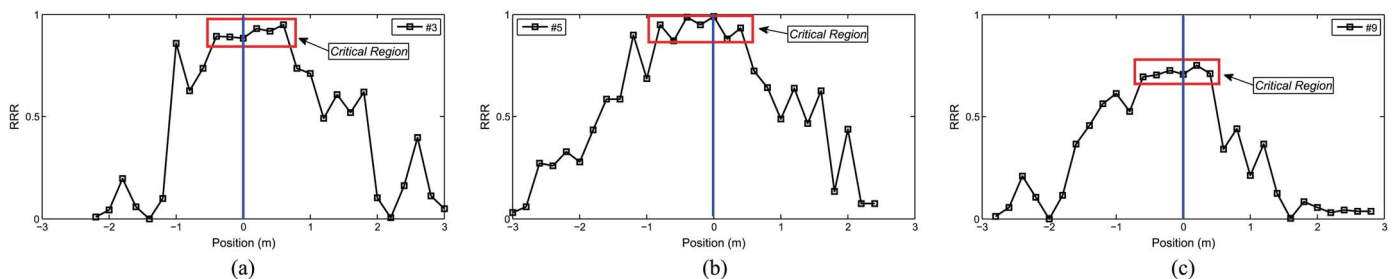


Fig. 5. RRR trends of three tags during the movement. (a) RRR trend of tag #3; (b) RRR trend of tag #5; (c) RRR trend of tag #9.

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

4.2 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 percent in the critical region. Nevertheless, RRR can be only around 60 percent if the tag is blocked by the luggage during the movement. Therefore, in practice, we cannot rely on any pre-defined threshold to detect the critical region. In OTrack, we 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 the most consecutive periods with a minimal variance. Such a range will be considered as the critical region for the tag in OTrack. To accurately quantify such closeness, we avoid using any threshold-based heuristic (e.g., $\leq \pm 5$ percent 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 noise, we use the *quadratic fitting* technique [6] to minimize the influence from noise. With the above two steps, we 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 first introduce notations:

- *period*: is certain amount of time; In our paper, time is partitioned into period;
- *window*: is composed of multiple periods;
- *d*: is the length of period;

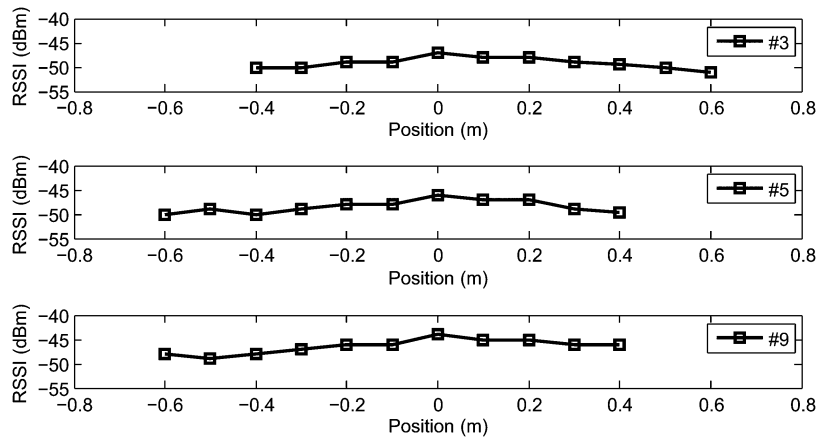


Fig. 6. Combination of RSSI and RRR.

- $p_{i,j}^k$: is the RRR of the k -th period in the j -th window of tag i ;
- $\bar{p}_{i,j}$: is the effective average RRR over the j -th window of tag i ;
- $w_{i,j}$: is the j -th window for tag i , and $|w_{i,j}|$ is the corresponding length in terms of period;
- W_i : is the window set of tag i ;
- w_i^* : is the window covers critical region of tag i ;
- $s_{i,j}$: is the number of periods, in which RRR is greater than zero within the j -th window of tag i ;
- S : is the set of identified tags;
- S^* : is the set of critical region of each tag;
- t_i^* : is the time stamp when RSSI peak of tag i appears.

Given a period, we can calculate RRR for a tag i . Then by grouping several consecutive periods, we form a window. As we may create multiple windows for the same tag in our design, we use $w_{i,j}$ to denote the window of the j -th window for tag i . Within a window, 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 $s_{i,j}$ to denote the periods, in which RRR is greater than zero and $|s_{i,j}|$ is the number of these periods. For example, $|s_{1,3}| = 7$ indicates that there are seven periods during which the RRR is greater than zero for tag i within its j -th window. Based on those definitions, an instrumental explanation of our design is as follows. For any window $w_{i,j}$, we define its effective average RRR as:

$$\bar{p}_{i,j} = \frac{\sum_{k \in s_{i,j}} p_{i,j}^k}{|s_{i,j}|}. \quad (2)$$

Where $p_{i,j}^k$ to denote the RRR of the k -th period of tag i in its j -th window. In this study, we find that by statistically comparing $|w_{i,j}| \times \bar{p}_{i,j}$ with $|s_{i,j}|$, we can conclude whether $w_{i,j}$ 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 $w_{i,j}$ is a subset of a critical region. A more formal specification is given by the following lemma.

Lemma 1. Let $\bar{p}_{i,j}$ be the effective average RRR over $w_{i,j}$. Then $w_{i,j}$ is completely within the critical region with probability

$\phi(\alpha)$ if $P\{|s_{i,j}| - |w_{i,j}| \cdot \bar{p}_{i,j}| \leq \alpha \sqrt{|w_{i,j}| \cdot \bar{p}_{i,j} \cdot (1 - \bar{p}_{i,j})}\}$ holds, where $\phi(\alpha)$ is determined via the Standard Gaussian Distribution Chart [10].

Proof. Window $w_{i,j}$ contains multiple periods. In each period, the reader calculates a response reception ratio for tag i . Since each RRR is calculated independently within its own period, the entire communication process in window $w_{i,j}$ can be described by a Bernoulli process. For each period, we define an event and the probability of its occurrence equals RRR. Then, the reader conducts Bernoulli trials in this window $|w_{i,j}|$ times. According to the *Central Limit Theorem*, we have:

$$\lim_{|s_{i,j}| \rightarrow |w_{i,j}|} P\left\{ \frac{|s_{i,j}| - |w_{i,j}| \cdot \bar{p}_{i,j}}{\sqrt{|w_{i,j}| \cdot \bar{p}_{i,j} (1 - \bar{p}_{i,j})}} \leq \alpha \right\} = \phi(\alpha). \quad (3)$$

When $|s_{i,j}| \rightarrow |w_{i,j}|$, we can rephrase eq. (3) above as follows:

$$P\{|s_{i,j}| - |w_{i,j}| \cdot \bar{p}_{i,j} \leq \alpha \cdot \sqrt{|w_{i,j}| \cdot \bar{p}_{i,j} (1 - \bar{p}_{i,j})}\} = \phi(\alpha).$$

The equation above indicates that if window $w_{i,j}$ is in the critical region, the value of $|s_{i,j}|$ should be within $\pm \alpha \sqrt{|w_{i,j}| \cdot \bar{p}_{i,j} (1 - \bar{p}_{i,j})}$ with a probability close to $\phi(\alpha)$. For example, the value of $|s_{i,j}|$ is within $\pm 2 \sqrt{|w_{i,j}| \cdot \bar{p}_{i,j} (1 - \bar{p}_{i,j})}$ with the probability close to 0.98. \square

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 $w_{i,j}$ 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 5, we will investigate how to select the parameter α . After α has been properly chosen, it indicates that $w_{i,j}$ is within the critical region with a high probability when the inequality holds. If $w_{i,j}$ is within the critical region, we can gradually increase its window size such that the final $w_{i,j}$ will cover the entire critical region of tag i . After figuring out the critical region,

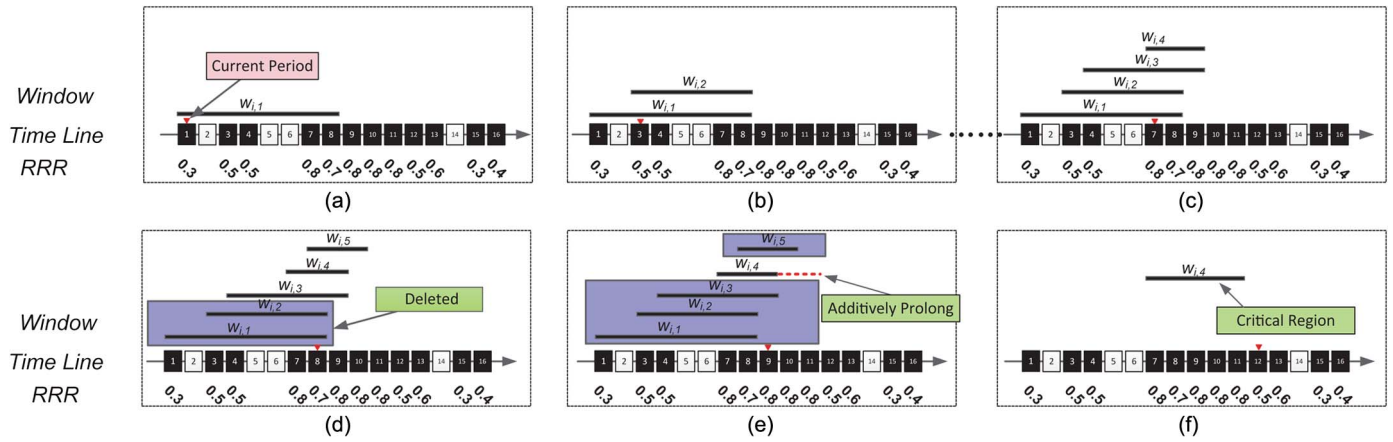


Fig. 7. Running example of Algorithm 1.

we will further apply the quadratic fitting technique to obtain the time when the RSSI peak appears. The order of tags can 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, it will create the first window $w_{i,1}$ for this tag. In the periods afterwards, if RRR is not zero, the reader will generate a new window $w_{i,2}$. By doing so, 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 $w_{i,1}$ as shown by Fig. 7a. Later, new windows will be generated if periods are of the black color (Figs. 7b and 7c). In period 8, we reach the end of $w_{i,1}$ and $w_{i,2}$. Since the inequality in Lemma 1 does not hold for those two windows, they are discarded as shown by Fig. 7d. In period 9, as depicted by Fig. 7e, the ends of $w_{i,3}$ and $w_{i,4}$ are reached. Since only $w_{i,4}$ passes the verification of Lemma 1, $w_{i,3}$ will be deleted and $w_{i,4}$ will be gradually increased in the following periods. Note that at this time, we can delete $w_{i,5}$ as well. Although $w_{i,5}$ is also within the critical region of tag i , its final size will be shorter than $w_{i,4}$ and we prefer a longer one. Eventually, the inequality in Lemma 1 becomes invalid for $w_{i,4}$ in period 12 and detected critical region is illustrated in Fig. 7f.

After the critical region of tag i is determined by Algorithm 1, the OTrack protocol will sort the order of

Algorithm 1: *criticalRegionSearch(i)*

Input : tag ID # i ;
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**
6 Create a new window $w_{i,j}$ and insert it into the window set W_i ;
7 **for each** window $w_{i,j}$ in W_i **do**
8 **if** $w_{i,j}$ expires **then**
9 **if** $w_{i,j}$ passes Lemma 1 **then**
10 Mark $is_win_size_increasing$ as $True$;
11 $w_i^* \leftarrow w_{i,j}$;
12 Delete all other windows in W_i ;
13 **else**
14 Delete $w_{i,j}$;
15 return $NULL$;

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).

Algorithm 2: The OTrack Protocol

Input : $null$;
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 ; (Identified tag set S , initially, $S = \emptyset$);
4 **for each** tag i in S **do**
5 $S^* \leftarrow S^* \cup criticalRegionSearch(i)$; Critical region set S^* , initially, $S^* = \emptyset$);
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;

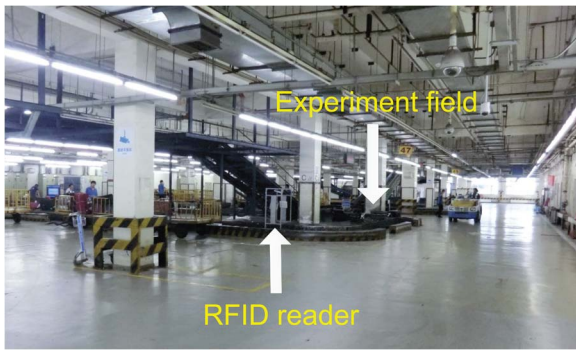


Fig. 8. Experiment field.

4.3 Protocol Analysis

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

Proof. Based on the RRR value observed in the first period of window $w_{i,j}$, the probability of the reader receiving no response in the whole window can be approximated by $(1 - p_{i,j}^1)^{|w_{i,j}|}$. Since the trend of RRR is statistically increasing within the first half of the RRR trace, such an approximation serves as an upper bound of its true value in the statistical sense. Then we require the approximated probability $\leq \eta$ and we obtain $|w_{i,j}| \cdot \ln(1 - p_{i,j}^1) \leq \ln \eta$. Since $p_{i,j}^1 \in (0, 1)$, we have $-p_{i,j}^1 \geq \ln(1 - p_{i,j}^1)$. Combining those two inequalities, we get $-\alpha \cdot p_{i,j}^1 \leq \ln \eta$. Therefore, $|w_{i,j}| \geq \ln(1/\eta)/p_{i,j}^1$. As $|w_{i,j}|$ is an integer, we make $|w_{i,j}| \geq \lceil \ln(1/\eta)/p_{i,j}^1 \rceil$. \square

In OTrack, the default value of η is set to be 0.05 and Lemma 2 indicates that the window size $\lceil \ln(1/\eta)/p_{i,j}^1 \rceil$ is large enough to guarantee that a clean 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.

4.3.2 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 and Lemma 4, 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 of $\|s_{i,j} - |w_{i,j}| \cdot \bar{p}_{i,j}\|$. The probability $P\{|s_{i,j}| - |w_{i,j}| \cdot \bar{p}_{i,j} \geq \delta \mid w_{i,j} \text{ is within a critical region}\}$ is $\leq \left[\frac{e^\delta}{(1+\delta)^{(1+\delta)}}\right]^{|w_{i,j}| \cdot \bar{p}_{i,j}}$.

Due to the page limit, we omit the proof here.

Lemma 4. The probability $P\{|s_{i,j}| - |w_{i,j}| \cdot \bar{p}_{i,j} \leq -\delta \mid w_{i,j} \text{ is within a critical region}\}$ is $\leq \frac{e^{-\delta^2 \cdot |w_{i,j}| \cdot \bar{p}_{i,j}}}{2}$.

Due to the page limit, we omit the proof here.

Lemmas 3 and Lemma 4 indicate 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.

5 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.

5.1 Experiment Setting

We implement OTrack on a workstation equipped with an Intel Core i7 CPU (2.93 GHz) and an 8 GB 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 10,000 pieces of luggage as shown by Fig. 8. Each luggage is attached with an Alien I2 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 3, 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:

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

5.2 Experiment Results

5.2.1 Investigation on α in Lemma 1

Fig. 9 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 $w_{l,k}^i$ is prone 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. To explore an appropriate setting of the system parameter α , we examine ten representative values of α in this experiment. As shown by Fig. 9, 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. 9, we configure α to be 2 in the following experiments.

5.2.2 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

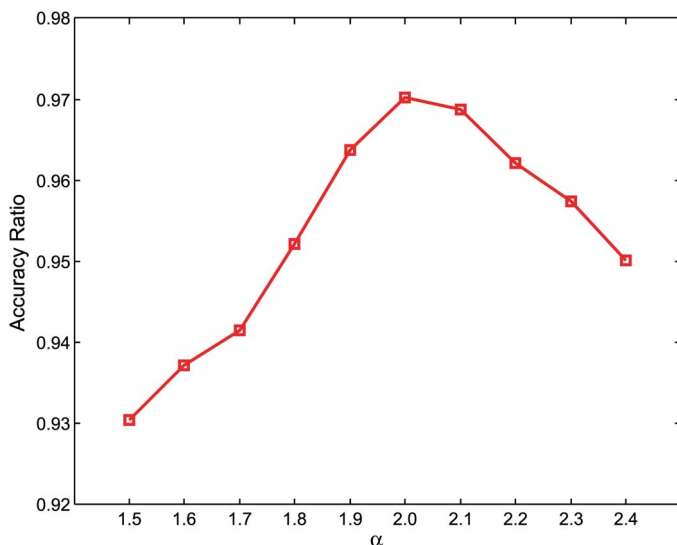


Fig. 9. Accuracy ratio vs. α .

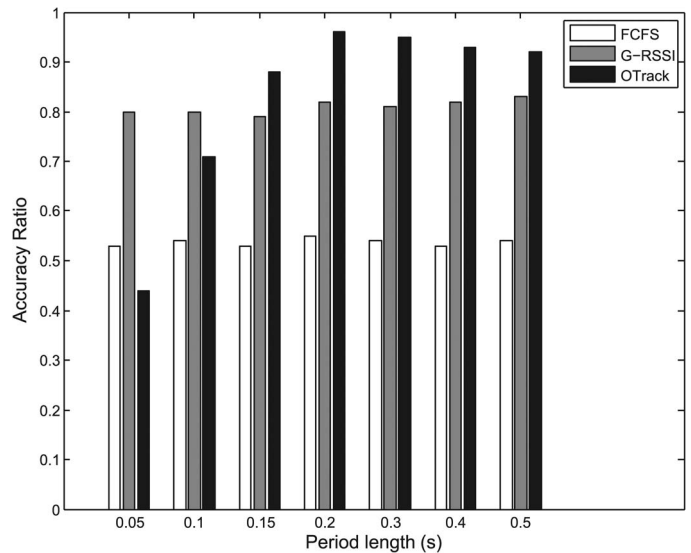


Fig. 10. Accuracy ratio vs. period.

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. 10, we vary the period length from 0.05 s to 0.5 s 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 0.05 s. As the period length increases, the accuracy ratio of OTrack increases dramatically. When the period length is 0.2 s, the accuracy ratio is up to 0.98. In Fig. 10, 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 percent+ and 20 percent+, respectively.

5.2.3 Scrutiny on the Performance of Three Algorithms

To give a concrete example, we choose to look at three typical pieces of luggage as shown by Fig. 11 and demonstrate how each protocol executes. In Fig. 12, we plot the RSSI traces collected from tags attached to those three pieces of luggage. According to the ground truth, their correct order should be #52, #53, and #54, where tag #52 is in front. Based on the time stamp when each tag enters the communication range of the reader, FCFS can successfully determine tag #52 to be in front. However, it mistakes the order of other two tags as the reader hears tag

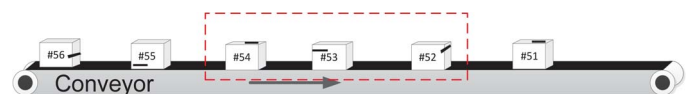


Fig. 11. Relative location of these luggage.

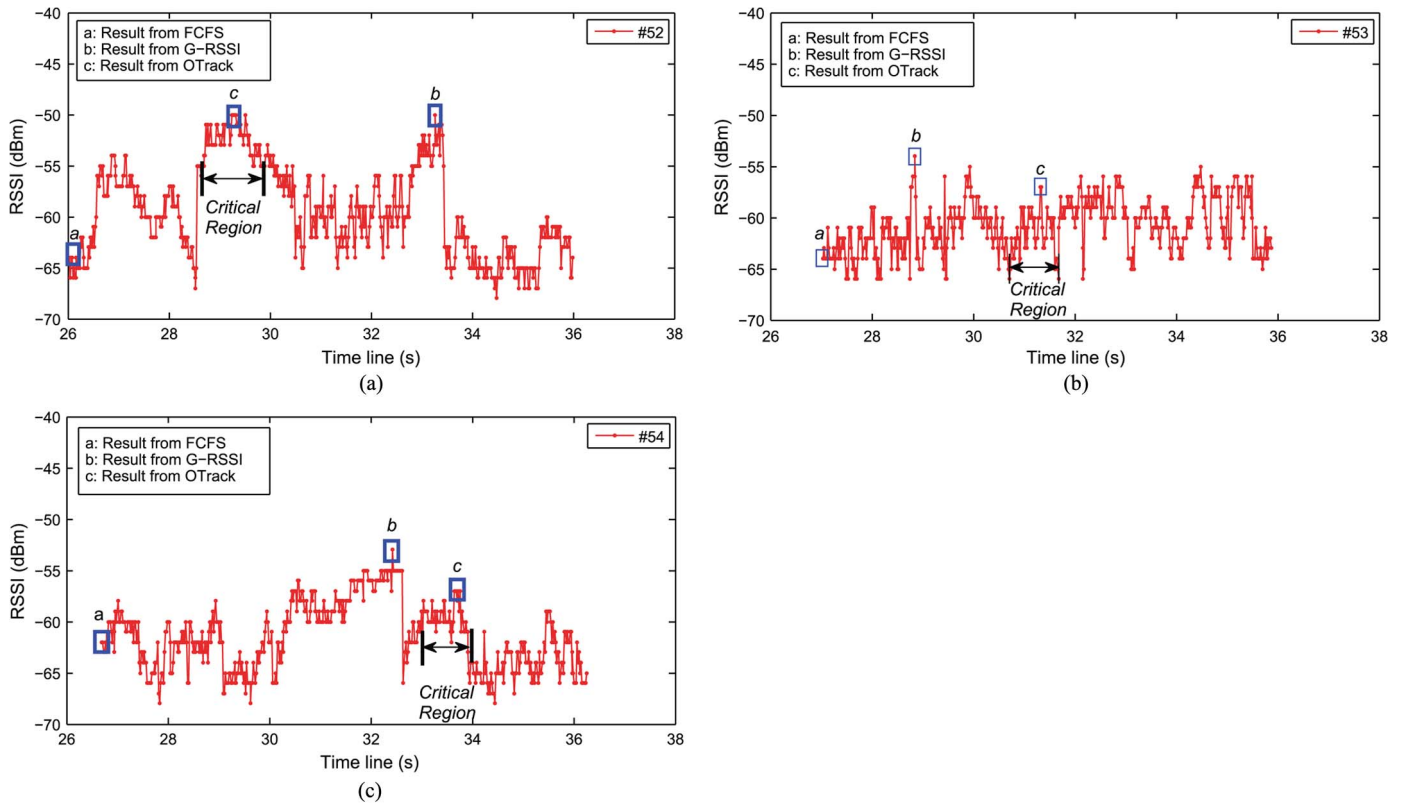


Fig. 12. RSSI traces of three consecutive tags in Fig. 11: (a) RSSI trace of tag #52; (b) RSSI trace of tag #53; and (c) RSSI trace of tag #54.

#54 prior to tag #53. G-RSSI performs comparably with FCFS in this case. Due to the fluctuation of RSSI, the RSSI peak of tag #52 appears quite late and the tag #53's appears quite early. The detected order of G-RSSI is #53, #52, and #54. Different from FCFS and G-RSSI, the RSSI peak detected within the critical region offers a consistent time reference for OTrack to determine tags' order. According to Fig. 12, the order obtained from OTrack is the same as the ground truth.

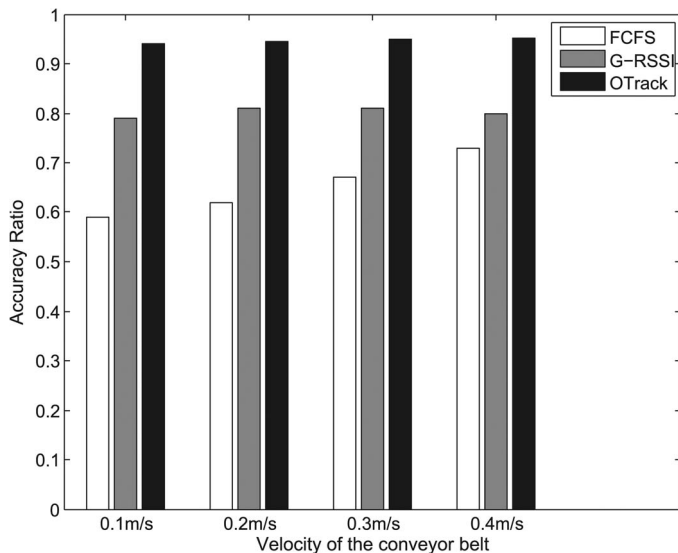


Fig. 13. Accuracy ratio vs. velocity of the conveyor.

5.2.4 Accuracy Ratio vs. Velocity of the Conveyor

In this trial of experiment, we investigate different protocols' performances with different velocities of the conveyor belt. Through our study, we find that the trends of RSSI and RRR are not changed under different belt velocities. Their relative correlation remains as well. As a result, there is no obvious performance variance for both OTrack and G-RSSI. It indicates that our protocol is robust to the velocity setting of the conveyor belt. As tags are prone to be exposed to the reader when the belt velocity is high, the accuracy ratio of FCFS slightly increases with the velocity's increasing. However, its accuracy ratio is still very low, less than 0.7 for most of the time as shown in Fig. 13.

5.2.5 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*,

TABLE 1
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]

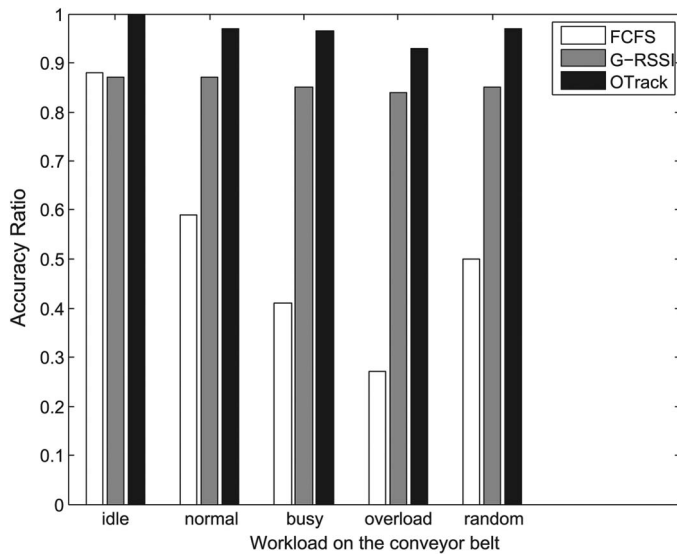


Fig. 14. Accuracy ratio vs. luggage distance.

etc., to name several typical distance settings in Table 1. From Fig. 14, 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. 14 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. 14 indicates that our proposed protocol can effectively handle various shipping workloads in practice.

6 CONCLUSION

In this paper, we studied 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.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for providing valuable comments. This work was supported in

part by the NSFC Major Program 61190110, NSFC under Grants 61171067, 61133016, and 61373175, Grant 2011AA010100, National Basic Research Program of China (973) under Grant 2012CB316200, and the NSFC Distinguished Young Scholars Program under Grant 61125202, the Fundamental Research Funds for the Central Universities of China under Project 2012jdgz02 (Xi'an Jiaotong University).

REFERENCES

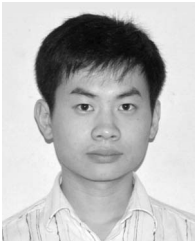
- [1] B. Sheng, C.C. Tan, Q. Li, and W. Mao, "Finding Popular Categories for RFID Tags," in *Proc. MobiHoc*, 2008, pp. 159-168.
- [2] L. Xie, B. Sheng, C.C. Tan, H. Han, Q. Li, and D. Chen, "Efficient Tag Identification in Mobile RFID Systems," in *Proc. IEEE INFOCOM*, 2010, pp. 1-9.
- [3] C.C. Tan, S. Bo, and Q. Li, "How to Monitor for Missing RFID Tags," in *Proc. ICDCS*, 2008, pp. 295-302.
- [4] EPC Class 1 Gen 2 RFID Standards, 2008. [Online]. Available: http://www.gs1.org/gsm/kc/epcglobal/uhf1g2/uhf1g2_1_2_0-standard-20080511.pdf
- [5] L. Roberts and G. Lawrence, "ALOHA Packet System with and without Slots and Capture," *Computer Comm. Rev.*, vol. 5, no. 2, pp. 28-42, Apr. 1975.
- [6] W. Sun and Y. Yuan, *Optimization Theory and Methods: Nonlinear Programming*, 1st ed., New York, NY, USA: Springer-Verlag, 2010.
- [7] [Online]. Available: <http://www.aliotechnology.com/readers/>
- [8] [Online]. Available: <http://www.aliotechnology.com/tags/>
- [9] [Online]. Available: <http://www.hongkongairport.com/gb/>
- [10] [Online]. Available: <http://www.mathsisfun.com/data/standard-normal-distribution-table.html>
- [11] J. Hightower, R. Want, and G. Borriello, "SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength," Dept. Computer Science. Eng., Univ. of Washington, Seattle, WA, USA, UW CSE 00-02-02, 2000.
- [12] L. Ni, Y. Liu, Y.C. Lau, and A.P. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID," in *Proc. IEEE PerCom*, 2003, pp. 407-415.
- [13] Y. Zhao, Y. Liu, and L. Ni, "VIRE: Active RFID-Based Localization Using Virtual Reference Elimination," in *Proc. ICPP*, 2007, pp. 1-8.
- [14] N.B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket Location-Support System," in *Proc. MobiCom*, 2000, pp. 32-43.
- [15] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Arawala, "Pinpoint: An Asynchronous Time-Based Location Determination System," in *Proc. MobiSys*, 2006, pp. 165-176.
- [16] Y.C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy Characterization for Metropolitan-Scale Wi-Fi Localization," in *Proc. MobiSys*, 2005, pp. 233-245.
- [17] J. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. Hicks, S. Teller, and J. Ledlie, "Growing an Organic Indoor Location System," in *Proc. MobiSys*, 2010, pp. 271-284.
- [18] E.S. Bhasker, S.W. Brown, and W.G. Griswold, "Employing User Feedback for Fast, Accurate, Low-Maintenance Geolocation," in *Proc. IEEE Percom*, 2004, pp. 111-120.
- [19] M. Azizyan, I. Constandache, and R.R. Choudhury, "Surround-Sense: Mobile Phone Localization via Ambience Fingerprinting," in *Proc. MobiCom*, 2009, pp. 261-272.
- [20] E. Welbourne, N. Khoussainova, J. Letchner, Y. Li, M. Balazinska, G. Borriello, and D. Suci, "Cascadia: A System for Specifying, Detecting, Managing RFID Events," in *Proc. MobiSys*, 2008, pp. 281-294.
- [21] T. Tran, C. Sutton, R. Cocci, N. Yanming, D. Yanlei, and P. Shenoy, "Probabilistic Inference Over RFID Streams in Mobile Environments," in *Proc. IEEE ICDE*, 2009, pp. 1096-1107.
- [22] T. Li, S. Chen, and Y. Ling, "Identifying the Missing Tags in a Large RFID System," in *Proc. Mobihoc*, 2010, pp. 1-10.
- [23] L. Yang, J. Han, Y. Qi, and Y. Liu, "Identification-Free Batch Authentication for RFID Tags," in *Proc. ICNP*, 2010, pp. 154-163.
- [24] L. Shanguan, Z. Li, Z. Yang, M. Li, and Y. Liu, "OTrack: Order Tracking for Tags in Mobile RFID System," in *Proc. IEEE INFOCOM*, 2013, pp. 3066-3074.



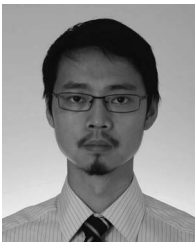
Longfei Shangguan is pursuing his PhD degree at Department of Computer Science and Engineering, Hong Kong University of Science and Technology. He received the MPhil degree from HKUST, in 2013 and BS degree from Xidian University, in 2011. His research interests include pervasive computing, wireless sensor networks and RFID system. He is a Student Member of the IEEE and ACM.



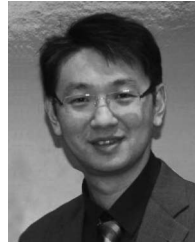
Zhenjiang Li works in the Wireless And Distributed Sensing (WANDS) research group, School of Computer Engineering, Nanyang Technological University, Singapore. He received his BE degree in the Department of Computer Science and Technology from Xi'an Jiaotong University, the Mphil degree in Department of Electronic and Computer Engineering from Hong Kong University of Science and Technology, and the PhD degree in the Department of Computer Science and Engineering of Hong Kong University of Science and Technology. He is a member of IEEE and ACM.



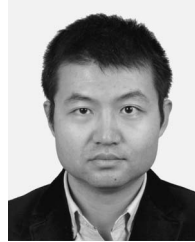
Zheng Yang is currently a post-doc fellow in the Department of Computer Science and Engineering of Hong Kong University of Science and Technology. He received his BE degree in the Department of Computer Science from Tsinghua University, Beijing, China, and his PhD degree in the Department of Computer Science and Engineering of Hong Kong University of Science and Technology. He is a member of IEEE and ACM. He has been awarded the 2011 State Natural Science Award (second class).



Mo Li is a Nanyang Assistant Professor (NAP) in School of Computer Engineering, Nanyang Technological University. He received his BS degree in Department of Computer Science and Technology from Tsinghua University, and the PhD degree in Department of Computer Science and Engineering from Hong Kong University of Science and Technology. He is heading Wireless And Networked Distributed Sensing (WANDS) system group in Parallel and Distributed Computing Centre (PDCC). He is a member of IEEE and ACM.



Yunhao Liu received the BS degree in automation from Tsinghua University, China, in 1995, the MS and PhD degrees in computer science and engineering from Michigan State University, in 2003 and 2004, respectively. He is now a professor at TNLIST, School of Software, Tsinghua University, as well as a faculty member with the Department of Computer Science and Engineering at the Hong Kong University of Science and Technology. His research interests include wireless sensor network, peer-to-peer computing, and pervasive computing. He is a Senior Member of the IEEE, IEEE Computer Society, and an ACM Distinguished Speaker.



Jinsong Han is currently an associate professor at Xi'an Jiaotong University. He received his PhD degree from Hong Kong University of Science and Technology. His research interests include pervasive computing, distributed system, and wireless network. He is a member of IEEE and ACM.

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.