

# Revisiting Reading Rate with Mobility: Rate-Adaptive Reading of COTS RFID Systems

Qiongzhen Lin<sup>1</sup>, Student Member, IEEE, Lei Yang<sup>1</sup>, Member, IEEE, Chunhui Duan<sup>1</sup>, Student Member, IEEE, and Yunhao Liu, Fellow, IEEE

**Abstract**—Radio-frequency identification (RFID) systems, as major enablers of automatic identification, are currently supplemented with various interesting sensing functions, e.g., motion tracking. All these sensing applications forcedly require much higher reading rate (i.e., sampling rate) such that any fast movement of tagged objects can be accurately captured in a timely manner through tag readings. However, COTS RFID systems suffer from an extremely low individual reading rate when multiple tags are present, due to their intense channel contention in the link layer. In this work, we present a holistic system, called Tagwatch, a rate-adaptive reading system for COTS RFID devices. This work revisits the reading rate from a distinctive perspective: *mobility*. We observe that the reading demands of mobile tags are considerably more urgent than those of stationary tags because the states of the latter nearly remain unchanged; meanwhile, only a few tags (e.g., < 20%) are actually in motion despite the existence of a massive amount of tags in practice. Thus, Tagwatch adaptively improves the reading rates for mobile tags by cutting down the readings of stationary tags. Our main contribution is a two-phase reading design, wherein the mobile tags are discriminated in the Phase I and *exclusively* read in the Phase II. We built a prototype of Tagwatch with COTS RFID readers and tags. Results from our microbenchmark analysis demonstrate that the new design outperforms the reading rate by 3.2× when 5 percent of tags are moving.

**Index Terms**—RFID, internet of things, rate-adaptive reading

## 1 INTRODUCTION

RADIO-FREQUENCY identification (RFID), as one of the top 10 influential technologies in the 21st century [1], is widely used in various fields, such as logistic, supply chain management, asset management, and so on. In contrast to conventional identification technologies (e.g., barcode), RFID offers many attractive advantages, such as non-optical proximity, long transmission range, multiple inventory, and so on. The initial purpose of RFID is to automatically identify (i.e., Auto-ID) objects fast and conveniently. Its potential applications have been explored in the past decade; among which, the most promising is in tracking mobile objects. Many applications benefit from highly accurate motion tracking. For example, supermarkets can thoroughly observe the shopping habits of consumers by monitoring the trajectories of items [2]. Several new battery-free human-machine interactive devices can be developed, such as writing letters in the air by attaching a tag to a finger [3]. Complex multiplayer behavior is also automatically recognized through a tagged football in an international game [4]. Enabling a robot to search for, pick up, fetch, and deliver a particular object from an assembly line has received considerable interest in both the academic community and

various industries [5]. High-precision 3D orientation [6] or rotations [7] of passive objects are monitored through back-scatter signals from tags. The basic concept behind the above tracking applications is to regard each tag reading as one sampling of its motion state (or state of its attached object), which imposes a tacit assumption; that is, the sampling rate (i.e., reading rate, the number of reading times per second) must be sufficiently high, such that an object's state changes can be continuously and faultlessly captured using tag readings for both slow and fast transitions.

A number of recent works have attempted to explore the possibility of improving reading rate from two layers. (1) The first group of studies enables RFID tags to transmit in parallel in the physical layer [8], [9], [10], [11]. However, these work cannot be applied to mobile tags because they require tags remaining the channel coefficients relatively unchanged. Clearly, moving tags change their coefficients rapidly, making physical symbol clusters in the constellation domain difficult to distinguish. (2) The second group of studies attempts to design efficient anti-collision protocols. RFID tags respond completely to a centralized scheduling of a reader for medium access. The current inefficient scheduling mechanism has resulted in a large number of *useless* empty and collided slots. Prior designs [12], [13], [14], [15], however, require substantial modifications to COTS tags or reader to enable collision resolution. The aforementioned two types of schemes can hardly benefit from the existing deployment of billions of COTS RFID tags worldwide with globalized standards.

Unlike the above prior works, we explore tag's reading rate from a third perspective: *mobility*. We observe that only a few tags (e.g., < 20%) are actually in motion despite the

- Q. Lin and L. Yang are with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. E-mail: {lin, young}@tagsys.org.
- C. Duan and Y. Liu are with School of Software, Tsinghua University, Beijing 100084, China. E-mail: duan@tagsys.org, yunhao.liu@gmail.com.

Manuscript received 28 Feb. 2018; revised 1 July 2018; accepted 6 Aug. 2018. Date of publication 13 Aug. 2018; date of current version 31 May 2019.

(Corresponding author: Lei Yang.)

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/TMC.2018.2864936

existence of a massive amount of tags in practice. For example, tens out of thousands of goods in a supermarket or warehouse are simultaneously picked up by a customer in an instant. Our real trace data acquired from a medium-sized sorting system also suggest that less than 10 percent of package pieces are being transported on conveyors (see Section 3). The remaining package pieces are sorted well and always remain stationary. Stationary tags do not urgently require a higher reading rate because their states remain unchanged (i.e., similar to the state obtained in the previous time). However, *their participations for the channel competitions would significantly drag down the reading rates of mobile tags due to signal collisions*. This inspires us to treat the individual reading rate for each tag differently and perform a *rate-adaptive reading*, which allocates more time for reading mobile tags but less time for reading static tags.

In this work, we present a holistic system, called Tagwatch, a rate-adaptive reading system for COTS RFID devices. This system offers higher reading rate for mobile tags, which have high priority in surveillance applications. Transforming this high-level idea into a practical system requires addressing two main challenges. (1) *How do we know which tags are moving?* Identifying mobile tags is challenging because their movements are unpredictable, particularly in a rapidly changing environment with non-ideal communication conditions. Even worse, a static tag may suddenly start to move at any time, or vice versa. (2) *How do we exclusively read a specific set of mobile tags without reading stationary tags?* The naive solution of dividing space for multiple access fails because the spatial distribution of mobile tags are totally unknown, and the mobile tags may frequently go across multiple divisions. Inappropriate space division may lead to massive unexpected readings of stationary tags, which are collaterally covered.

To address these challenging issues, Tagwatch adopts a two-phase reading design. In the first phase, Tagwatch begins a relatively shorter inventory procedure to read all tags. Thereafter, Tagwatch performs a motion assessment using backscatter signals from tags to identify mobile tags (see Section 5). In the second phase, Tagwatch performs selective reading (i.e., using `Select` command), to intensively and exclusively read mobile tags for a relatively longer interval (see Section 6). In this manner, the average individual reading rates of mobile tags are significantly increased.

*Summary of Results.* Tagwatch works with COTS RFID devices and is a purely software solution. We built and evaluated a prototype of Tagwatch using ImpinJ readers and numerous Alien tags (see Section 8). We conducted extensive testbed experiments in our laboratory (see Section 9) and obtained the following findings:

- We accurately model the reading rate for current COTS readers. Both the model and our empirical study indicate that the mean individual reading rate will drastically decrease by 84 percent when the total number of tags is over 30.
- Tagwatch can successfully identify mobile tags with a mean probability of 80 percent as long as a tag moves beyond 1 cm. It also achieves 95 percent of accuracy for motion detection, whereas the false positive rate (FPR) is maintained at below 10 percent.

- Tagwatch can outperform the individual reading rates of mobile tags by a median of  $3.2\times$  and  $1.9\times$  when there are 5 and 10 percent mobile tags.

*Contributions.* In this work, we made the following contributions: To the best of our knowledge, Tagwatch is the first RFID system that performs rate-adaptive reading on COTS devices by considering tag's mobility. It solves a practical problem in the domain of motion surveillance using RFID tags. Specifically, we introduce the algorithm of motion assessment to automatically identify mobile tags without an ideal communication model. Second, by converting selective reading into the set-covering problem, we present a scheduling algorithm for the selective reading of mobile tags. Third, we systematically evaluate the system and present its usage in our pilot study.

## 2 A PRIMER ON RFID SYSTEM

An RFID system is composed of an reader and many tags. The purpose of reader is to collect all EPCs stored in tags. RFID systems adopt the reader-talks-first mode, in which the reader dominates communication and all the tags follow its commands. The battery-free tags cannot hear from each other and compete for the channels. The corresponding protocol is called as anti-collision protocol. This section reviews the mainstream anti-collision protocol used in COTS RFID systems.

*Framed Slotted ALOHA (FSA).* FSA is the basic anti-collision protocol. In FSA, the reader divides time into several slots, which are further organized into frames. The reader broadcasts the `Select` command to start an inventory round, which consists of several frames. In the beginning of a frame, the reader broadcasts the command `Query` which takes the parameter of the frame length  $f$  (i.e., the number of slots present in the current frame). After receiving the `Query` command, each tag randomly selects an integer  $\in [0, f - 1]$  and stores it in the local variable `SC`. Afterwards, the reader starts a time slot by broadcasting `QueryRep`, which lets the tag decrease its `SC` by one. If the `SC` of the tag is equal to 0, then it immediately replies with a 16-bit random signal (i.e., `RN16`). The tag that chooses a collision-free slot is acknowledged with the `ACK` command and allowed to transmit its EPC in the subsequent time slice. Otherwise, the reader sends `QueryRep` to proceed to the next time slot. This process continues until collision is no longer detected.

*Q-Adaptive.* The probability that a given time slot of FSA will make a single reply, denoted by  $q$ , is given by

$$q = \binom{n}{1} \frac{1}{f} \left(1 - \frac{1}{f}\right)^{n-1}, \quad (1)$$

where  $n$  is the total number of tags. When the derivative of the preceding equation is taken, the maximum probability is obtained  $q_{\max} = 1/e$  where  $f = n$ . That is, to achieve the maximum reading rate, the frame length should always be set to the total number of tags participating in the current frame. The ideal FSA can be designed as follows. (1)  $f = n$  is initialized. (2) Each time a tag is successfully identified, the current frame is terminated and a new frame starts with  $f = f - 1$ , such that each slot maintains the maximum probability of a successful reply. The frame length is dynamically adjusted;

hence, we call the scheme *Dynamic FSA* (DFSA). However, the reader does not have a *priori* knowledge about  $n$ .

To address this issue, a COTS reader (i.e., aka Gen2 reader) adopts the Q-adaptive protocol (i.e., a variant of FSA.), which can adaptively estimate  $n$  according to history readings, and is based on an *award-punish* mechanism. In particular, the command Query contains a non-negative integer  $Q$ , which indicates the frame length  $f = 2^Q$ . The reader dynamically increases or decreases  $Q$  at the end of each collided slot or empty slot.

### 3 UNDERSTANDING THE READING RATE

The Individual Reading Rates (IRR) of tags, defined as the number of readings obtained from a particular tag per second. As aforementioned, each reading is considered as a sampling of a tag's location. Thus, the IRR becomes a crucial parameter for many applications like tracking and surveillance. IRR highly depend on the efficiency of the anti-collision protocol, which is used to avoid signal collisions that occur when multiple tags reply their IDs simultaneously. To understand the reading rate, this section discusses the IRR from both theoretical and practical perspective.

#### 3.1 Theoretical Analysis of Reading Rate

Suppose there are  $n$  tags currently, we use ALOHA-like protocol to collect these  $n$  tags. What is the optimal frame length  $f$ ? Since the tag randomly selects one slot to transmit its ID, the probability that the tag selects any one of the  $f$  slots to transmit its ID equals  $1/f$ . Correspondingly, the probability that a time slot is uniquely selected by a tag is given by

$$P = n \left( \frac{1}{f} \right) \left( 1 - \frac{1}{f} \right)^{n-1}. \quad (2)$$

By taking the derivation of  $P$  with respect to  $f$ , we could obtain

$$\frac{\partial P}{\partial f} = n \left( \frac{1}{f^2} \right) \left( 1 - \frac{1}{f} \right)^{n-2} \frac{(n-f)}{f}. \quad (3)$$

Enabling the above equation to be 0, we would maximize  $P$  and find the optimal frame  $f = n$ . This is the reason why we set the frame length to the number of tags. The maximum  $P_{\max} = (n \frac{1}{n} (1 - \frac{1}{n})^{n-1}) \approx 1/e$ . Now, let us consider what happens when one tag is successfully identified. Then, the above probability (i.e., a time slot is uniquely selected by a tag) turns to

$$P = (n-1) \left( \frac{1}{f} \right) \left( 1 - \frac{1}{f} \right)^{n-1}. \quad (4)$$

Obviously, the old frame length (i.e.,  $n$ ) cannot maximize the above equation any more. The optimal frame length changes to  $n-1$ . As an analogy, when  $i$  tags are identified, the optimal frame length equals  $n-i$ . Therefore, we should dynamically change frame length as follows. Initially, the frame length is set to  $n$ . Once a tag is identified in the  $i$ th frame, the system ends the  $i$ th frame immediately and starts a new frame (i.e.,  $(i+1)$ th frame) with a length of  $f_i = n-i$ .

Assume that the frame length can always be adjusted to the optimal value. How much time will the reader consume in an inventory round? This question can be reduced into the classical *Coupon Collector's Problem*, which states that there are  $n$  distinct objects that are repeatedly drawn (with replacement) from an urn with a probability of  $1/n$  of picking an object at each trial. What is the minimum number of trials needed to pick each of the  $n$  objects at least once? We sketch the analysis as follows. When  $f = n$ , and thus, the probability that the  $i$ th tag is successfully identified in a given slot is equal to

$$p = \frac{1}{n} \left( 1 - \frac{1}{n} \right)^{n-1} \approx \frac{1}{ne}. \quad (5)$$

Let the random variable  $\mathcal{F}$  denote the number of slots required by the reader to collect  $n$  tags, and  $f_i$  denote the length of the  $i$ th frame,  $0 \leq i \leq n-1$ , which starts when the  $i$ th tag is identified and ends when the  $(i+1)$ th tag is identified. Thus,  $n-i$  tags are yet to be read in the  $i$ th frame, and each of these tags has a probability  $p$  of being read in a time slot. The frame length,  $f_i$ , is a geometric random variable with parameter  $(n-i)p$ . Thus, when  $\mathcal{F} = f_0 + \dots + f_{n-1}$ , we obtain

$$E[\mathcal{F}] = \sum_{i=0}^{n-1} E[f_i] = \sum_{i=0}^{n-1} \frac{1}{(n-i)p} = \frac{1}{p} \sum_{i=1}^n \frac{1}{i} = \frac{1}{p} H_n \approx ne H_n, \quad (6)$$

where  $H_n$  denotes the  $n$ th harmonic number and is given by  $\ln n + O(1)$ . Therefore,

$$E[\mathcal{F}] = ne(\ln n + O(1)) \approx ne \ln n + O(n). \quad (7)$$

$E[\mathcal{F}]$  approaches  $ne \ln n$  time slots for a large value of  $n$ , where each tag can be read once within  $ne \ln n$  slots.

In addition to the time for collecting tags, each inventory round will introduce an extra time cost for other necessary tasks, e.g., broadcasting Select, synchronization, and clearing history states. We call such time consumption as *start-up cost*, which is denoted by  $\tau_0$ . Suppose that the mean duration<sup>1</sup> of each slot is equal to  $\bar{\tau}$ , then the entire inventory cost is defined as follows:

**Definition 1 (Inventory cost).** *Inventory cost, denoted by  $C(n)$ , is defined as the total time consumed on identifying  $n$  tags once. It is given by*

$$C(n) = \begin{cases} \tau_0 + ne\bar{\tau} \ln(n) & \text{if } n > 1 \\ \tau_0 + \bar{\tau} & \text{otherwise.} \end{cases} \quad (8)$$

With regards to the inventory cost, IRR is then given by

$$\Lambda(n) = \frac{1}{\tau_0 + ne\bar{\tau} \ln(n)}. \quad (9)$$

This equation provides a fundamental estimation model for calculating the scheduling cost (later used in Section 6). To the best of our knowledge, we are the first to *accurately*

1. The actual successful slot is longer than empty and collided slots due to the transmission of EPC. Thus, we use an average approximation.

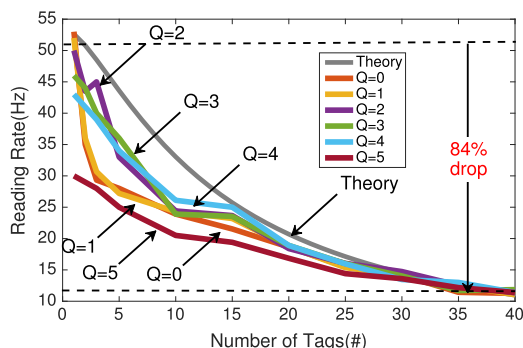


Fig. 1. Empirical study on IRR. Gen2 protocol allows upper application to set an initial  $Q$  as figure shows. However, the reader will gradually and automatically adjust the actual  $Q$ , making total inventory time approach optimal.

model the IRR for COTS RFID system, particularly the impact of start-cost, which was never considered before.

### 3.2 Empirical Analysis of Empirical Study

We then practically measure the reading rates of the Impinj R420 reader across 1 ~ 40 Alien tags, with various frequencies (920 ~ 926 MHz), including 16 channels. We try different initial settings of  $Q$  in the experiments. For each setting, we repeat the experiment 50 times and report the average value. We also utilize the least-squares algorithm to estimate the two unknown parameters, namely,  $\tau_0$  (19 ms) and  $\bar{\tau}$  (0.18 ms), involved in Eqn. (9). The average IRR is depicted in Fig. 1. The following two findings are obtained from the studies:

- (1) Our theoretical formula for IRR agrees well with the measurement results in terms of the trend, except for a slight difference due to the approximation in Eqn. (7). It is seen that the current anti-collision algorithm, i.e.,  $Q$ -adaptive, is approaching the optimal solution, leaving very limited room for us to improve the reading rate by designing better anti-collision protocols.
- (2) IRR is a purely decreasing function of  $n$ , i.e., inversely proportional to  $n \ln(n)$ . Specifically, as shown in Fig. 1, IRR decreases from 63 to 12 Hz (i.e., an 84 percent drop) when  $n$  increases to near 40. This finding hints that companionate tags would seriously and negatively affect the IRRs of the tags of interest.

## 4 OVERVIEW

Although many prior work attempt to improve the reading rate in physical layer by dealing with collisions, they cannot be directly used in practice and real systems. We target at the COTS RFID systems like Impinj R420 or Alien Reader, which follows the world-widely adopted EPCglobal Gen2 standard released since 2013. This section presents a real case which motivates our study, and then sketches the overall solution.

### 4.1 Motivation

*Practical Dilemmas.* We consider the issue of reading rate in real situations where billions of COTS RFID tags have been deployed and covered by RFID readers. We can neither

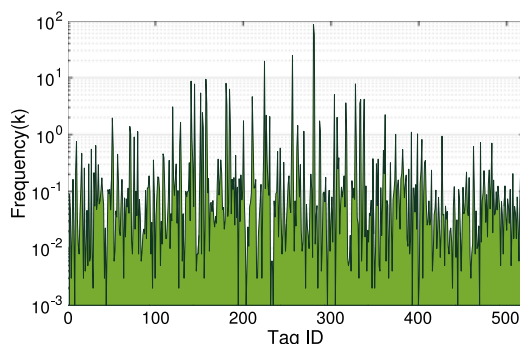


Fig. 2. Trace distribution. We find that 20 percent of the tags are read over 205 times, whereas 10 percent of the tags are read over 655 times. In fact, each individual tag is supposed to be read about 50 times when it passes through the TrackPoint.

change the device settings nor improve the efficiency of the current anti-collision protocol. We are not allowed to communicate tags with different frequency channels simultaneously (so that reading some tags more faster at one pace and the others at a slower pace), because tags do not equip with transceivers and must backscatter their IDs using the same frequency as emitted by the reader. In short, there is no way to apply frequency division multiple access (i.e., FDMA) in current RFID systems. With such rigorous rules, how can we improve the IRR of mobile tags? Before sketch our solution, let us look into a real case.

*A Real Case.* We collect a real reading trace from our previous tracking system, TrackPoint [16], i.e., a gate composed of three reader antennas is mounted above the conveyor to monitor moving baggage. We expect that each tag can be read 10 times *immediately during* it is moved through the device, so as to localize each baggage at a high-precision level. We choose an approximately 4 hour real trace from a TrackPoint. Totally, our RFID reader acquired up to 367,536 readings from 527 tags. We also show the distribution of reading times in Fig. 2. We visually observe that 30 tags ( $\approx 5.7\%$  tags) at most are simultaneously conveyed through the TrackPoint each second. However, we find that 20 percent of the tags are read over 205 times, whereas 10 percent of the tags are read over 655 times. In particular, Tag #271 has been continuously read 90,000 times totally! Why were these tags read so many times? After the further investigation, we surprisedly found that all these exceptional readings come from those tags, which are actually not moved on the conveyor but have been put on the places nearby the TrackPoint. By contrast, the real moving tags are typically read less than 5 times when being moved across the gate, due to the channel contentions from nearby stationary tags. In other words, these repetitive readings of static tags seriously affect the reading of our targets, i.e., mobile tags.

*Insight.* In summary, our empirical data suggests that only a few number of tags from a massive number of tags (e.g., 10 percent) are concurrently moving in practice, and their reading demand are urgent. Meanwhile, The IRRs of moving tags are seriously affected by the nearby static tags. Thus, reducing the total number of participating tags in the inventory is a good way of improving the IRR of mobile tags. This inspire us to revisit IRR from the perspective of mobility: *real-time adjustment of the IRR according to tag's current motion state.*

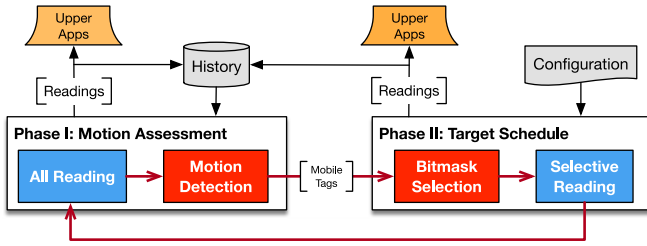


Fig. 3. System architecture. Tagwatch adopts a two-phase reading design, wherein the motion state of each tag is assessed in the first phase and only moving tags are read in the second phase.

## 4.2 Solution

Inspired by the previous insight, in this work, we design a rate-adaptive system, called as Tagwatch. It is a middle layer that runs between the reader and upper applications. It aims to *adaptively* change the reading rate of tags in terms of their motion states. As shown in Fig. 3, Tagwatch adopts a two-phase reading design:

*Phase I: Motion Assessment.* In this phase, Tagwatch reads *all tags* within a short period and then leverages the reading results to assess the motion state of each tag. Afterwards, Tagwatch selects *mobile tags* to be scheduled in the next phase according to the history-based immobility models.

*Phase II: Target Schedule.* In this phase, Tagwatch first selects a group of bitmasks to cover target tags (e.g., mobile tags) and then conducts bitmask-enabled selective reading on target tags for a relatively long period.

The two phases constitute a basic cycle, which occurs alternatively and periodically as shown in Fig. 4. The scheduling phase is longer than the assessment phase, which guarantees target tags gain sufficient time to be read for several cycles. A periodical motion assessment is necessary to capture the state transitions of tags (e.g., from a moving state to a static state, or vice versa). Target tags are read for a longer period, and thus, their average IRRs are higher than those of other tags. Regardless of the phase in which tags are read, all readings should be delivered to upper applications and contribute to the history database.

## 4.3 System Scope

Our system is driven by the assumption that quite small percent of tags are moving in a moment. It is possible that Tagwatch cannot obviously improve IRRs for mobile tags if the percent is over than a threshold (e.g.,  $> 20\%$ ). Keep in mind that as our baseline is to read them all, it is easy for us to switch back to the old fashion (i.e., reading them all) when the assumption does not hold true.

## 5 PHASE I: MOTION ASSESSMENT

This section discusses the first phase, wherein all the tags are continuously read once. The purpose of this phase is to identify all mobile tags.

### 5.1 Modeling Tag's Immobility

To determine which tags are moving, we utilize the physical RF signals of tags. The reading result of a tag contains two basic physical metrics, RF strength and phase. Most COTS RFID systems support milli-degree resolution in detecting RF phase (i.e., hypersensitive to the movement of the tag),

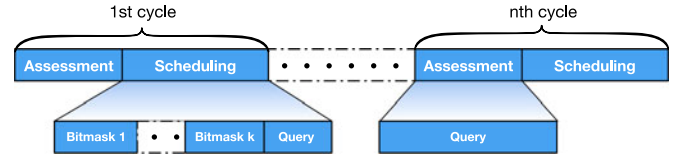


Fig. 4. Reading cycles. Each reading cycle is composed of two phase and occurs alternatively and periodically.

thus, we adopt the RF phase as a main measurement for motion detection.

*Challenges.* The naive method of perceiving the motion of a tag is to compare its incoming RF phase with the last one. If the two phases are same, then the tag is determined to be static; otherwise, it is in motion. However, such method suffers from a high rate of false positives because of the following reasons. (1) The phase estimate is derived from the received signal where the thermal noise from reader receiver is always present, thereby leading to measurement errors. (2) The RF signal propagation does not only occur along the direct path, i.e., line of sight (LOS), but also reflected by surrounding objects (particularly mobile objects). This phenomenon is known as the multipath effect. The final signal received at the receiver is a superposition of multiple copies of the original signals from all paths. Even if the original tag remains stationary, the movements of surrounding objects will create or cancel multipath propositions, thereby resulting in significant jumps of the RF phase.

*Gaussian Model.* A number of prior empirical works [16], [17] have shown that the RF phase measurement results contain random errors, following a typical Gaussian distribution. Thus, we consider the phase measurement for each tag in the scene as a Gaussian random variable instead of an accurate value. Assume that we currently have the  $m$  RF phase values, i.e.,  $\{\theta_1, \theta_2, \dots, \theta_m\}$  for a particular tag. Then the incoming new phase  $\theta_{m+1}$  should follow a Gaussian model, i.e.,

$$\theta_{m+1} \sim \mathcal{N}(\mu_m, \delta_m), \quad (10)$$

where  $\mu_m$  and  $\delta_m$  are the expectation and the standard deviation that estimated through the history readings as follows.

$$\mu_m = \frac{1}{m} \sum_{i=1}^m \theta_i \text{ and } \delta_m = \sqrt{\frac{1}{m} \sum_{i=1}^m (\theta_i - \mu_m)^2}. \quad (11)$$

If the tag's position remains unchanged, the incoming RF phase  $\theta_{m+1}$  should follow the above Gaussian model, whose probability density function is given by

$$\eta(\theta_m, \mu_m, \delta_m) = \frac{1}{\delta_m \sqrt{2\pi}} \exp\left(-\frac{(\theta_m - \mu_m)^2}{2\delta_m^2}\right). \quad (12)$$

Initially, we assume all the tags are in motion (i.e.,  $\mu_0 = 0$  and  $\theta_0 = 0$ ) and then immediately learn their immobility. Correspondingly, a tag is static if the incoming phase value matches the Gaussian model, i.e.,  $|\theta_{m+1} - \mu_m| < \xi \delta_m$  where  $\xi$  is a user-defined parameter.

*Gaussian Mixture Model (GMM).* RF phase is known for its sensitivity to the motion of a tag but notorious for being too excessively sensitive to the movements of surrounding objects due to the multipath effect. Fig. 5a illustrates a toy

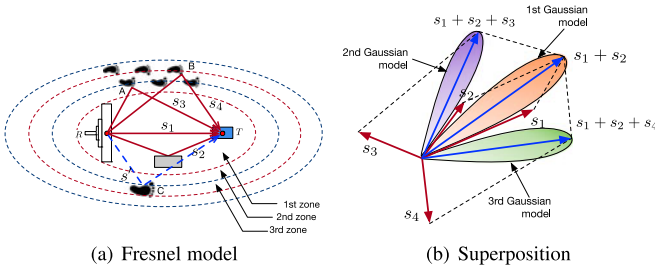


Fig. 5. Mixed Gaussian model. (a) An example showing how a surrounding mobile object affects the final RF signal. (b) The final RF signal received by the reader in three cases.

example in which a person walks by a pair of reader and tag. Two constant signal propagations are available:  $s_1$  and  $s_2$ . By default, the final signal received by the reader is  $s_1 + s_2$ , as shown in Fig. 5b. When the person passes by position  $A$  and  $B$ , he/she respectively introduces two new propagations,  $s_3$  and  $s_4$ , causing the final signal to jump from  $s_1 + s_2$  to  $s_1 + s_2 + s_3$  and  $s_1 + s_2 + s_4$ . Evidently, single Gaussian model fails to depict such jumps, producing two false positives.

We can divide a space into many Fresnel zones for every  $\lambda/2$  given a wavelength of  $\lambda$ , as shown in Fig. 5a. Let  $R$  and  $T$  be a pair of reader and tag. The Fresnel zones contains  $K$  ellipses can be constructed as follows:

$$|RQ_k| + |Q_kT| - |RT| = k\lambda/2, \quad (13)$$

where  $Q_k$  is a point on the  $k$ th ellipse. The preceding equation indicates the distance from points on the ellipse to two foci is  $k\lambda/2$  longer than the distance between the transmitter and receiver. The innermost ellipse is defined as the first Fresnel zone, and the  $k$ th Fresnel zone corresponds to the elliptical annuli between the  $(k-1)$ th and the  $k$ th ellipse. Clearly, the reflections caused by the objects located at odd zones superimposes LOS signal in phase, whereas the superpositions with reflections from even zones are out of phase. The previous research [18], [19] shows that the significant zones for RF transmission are the first 3 ~ 8 zones, more than 70 percent of the energy is transferred via the first Fresnel zone.

The Fresnel zone model<sup>2</sup> inspires us to re-build the immobility of a tag using multiple Gaussian models, i.e., Gaussian mixture model (GMM), wherein each combined propagation (i.e., multipath) result corresponds to a Gaussian model. As shown in Fig. 5b, three Gaussian models exist for the RF phase:  $\mathcal{L}(s_1 + s_2)$ ,  $\mathcal{L}(s_1 + s_2 + s_3)$  and  $\mathcal{L}(s_1 + s_2 + s_4)$ . Fig. 6 shows the distribution of the phase values collected from a stationary tag in a dynamic environment (i.e., asking a person to work around.). The value of the RF phase follows a group of Gaussians models instead of a single model. In particular, suppose we have learned  $K$  Gaussian models, denoted by  $\{\mathcal{N}_1(\mu_{1,m}, \delta_{1,m}), \dots, \mathcal{N}_K(\mu_{K,m}, \delta_{K,m})\}$ , for a particular stationary tag. During detection, the best matched Gaussian model (e.g.,  $k'$ th) is chosen to depict the

2. Note we apply Fresnel theory to explain why we use GMM to build the immobility of a tag. Finding Fresnel zones is unnecessary because our detection algorithm is self-learning and adaptive to changes in the environment.

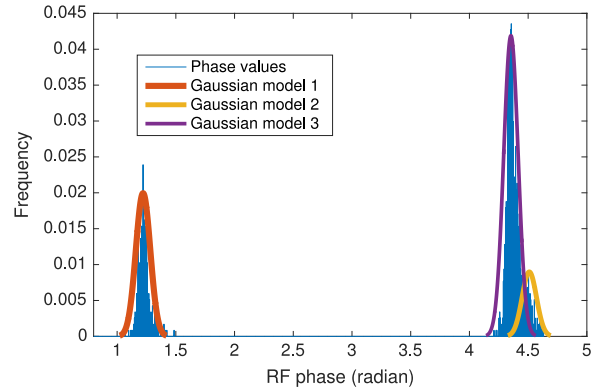


Fig. 6. Use of GMM to model the immobility of a tag.

current immobility of a tag. The determining rule is thereby updated to  $|\theta_{m+1} - \mu_{k',m}| < \xi\delta_{k',m}$ .

## 5.2 Self-Learning Motion Detection

Self-learning motion detection based on GMM is illustrated. First,  $K$  Gaussian models are assigned an initial large deviation  $\delta_k$  and low prior weight  $w_k$ . These  $K$  Gaussian models are ordered by the priority of  $r_k = w_k/\delta_k$ , because a Gaussian model with a high weight but a small deviation is preferred. The Gaussian model with the lowest priority will be gradually eliminated.

*Algorithm.* We build a stack of Gaussian models for each tag independently. On the incoming reading, a new RF phase value  $\theta_{m+1}$  is provided. The Gaussian model is “matched” if  $\theta_{m+1}$  is within  $\xi \times$  standard deviation of the Gaussian model, i.e.,  $|\theta_{m+1} - \mu_k| < \xi\delta_k$ . We search for the first matched model from top to down in terms of priority. Then one of the following two cases is observed:

- *Case 1:* If a match is found with one of the  $K$  Gaussian models in the stack, then the tag is classified as stationary. Correspondingly, we increase the weight of this model, adjust its mean closer to  $\theta_{m+1}$ , and decrease the variance as follows:

$$\begin{cases} w_{k,m+1} = (1 - \alpha) \times w_{k,m} + \alpha \\ \mu_{k,m+1} = (1 - \rho)\mu_{k,m} + \rho\theta_{m+1} \\ \delta_{k,m+1} = \sqrt{(1 - \rho)\delta_{k,m}^2 + \rho(\theta_{m+1} - \mu_{k,m})^2} \end{cases}, \quad (14)$$

where  $\alpha$  is a learning rate and  $\rho = \alpha\eta(\theta_{m+1}, \mu_m, \delta_n)$ <sup>3</sup>. For the unmatched models, we maintain their mean and deviation, but decrease their weights to  $w_{k,m+1} = (1 - \alpha)w_{k,m}$ .

- *Case 2:* A match is not found with any of the  $K$  Gaussian models. In this case, the tag is classified as being in motion. A new model with  $\mu_{k,m+1} = \theta_{m+1}$ , a large  $\delta_{k,m+1}$  (e.g.,  $2\pi$ ) and a small  $w_{k,m+1}$  (e.g., 0.0001) is pushed on to the stack, or replace the model with least priority if the stack is full.

If the tag moves to a new position, all its previous immobility models cannot characterize its new environment any more. That is the reason we introduce

3.  $\eta(\cdot)$  refers to Eqn. (12).  $w_{k,m}$ ,  $\mu_{k,m}$  and  $\delta_{k,m}$  respectively denote the weight, expectation, and standard deviation in the  $m$ th iteration of the  $k$ th Gaussian model.

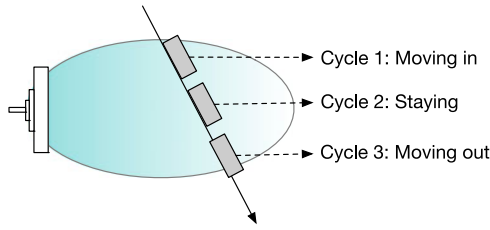


Fig. 7. A toy example. Suppose the tag moves into the region in the first cycle, stays for a cycle, and then moves out in the third cycle. Its state first transits to ‘moving’, ‘static’, and then ‘moving’.

the variable named *weight* to dynamically adjust the order of models. The model with the lowest weight (i.e., outdated) is wipe out when a new model is built up. Referring to the Case 2, it is seen from the algorithm that a new model with the new reading information is pushed into the stack or replace the model with the least priority if the stack is full. Referring to the Case 1, more times the model is matched, the higher priority it has. On the contrary, the models mismatched for many times will be assigned with a low priority.

### 5.3 Model Updating and State Transition

It should be noting that the learning of immobility model for a new tag does not consume extra time. The whole procedure is totally self-adaptive. As Fig. 3 shows, all readings of the mobile tags are inserted to the history database. When moving to the next Phase I, the system updates a tag’s mobility models using its previous history acquired the last Phase II. Thus, the system does not require extra reading time for learning. To better understand this design, we give a toy example in Fig. 7. Suppose a tag moves into the reader’s region in the first cycle, stays for a cycle, and then moves out in the third cycle. We show its state transition as follows:

C1-I: the tag is read once and considered as being mobile because no immobility model matches its current state; → C1-II: this tag is read for multiple times and all its readings are recorded in the history database; → C2-I: its history readings in the last cycle are used to update its immobility model. Its new reading in this phase is used to determine its state. As a result, it is determined to be static; → C2-II: the tag is not read in this phase; → C3-I: since the tags is moving out, its new reading does not match any learned immobility models and the tag is determined as moving; → C4-II: the tag is read in this phase.

Initially, a static tag is ‘falsely’ considered as being mobile and obtain more readings in the first Phase II because its immobility models are not built up yet. This procedure continues until its immobility models reach a convincing levels. Then the tag will not join in the Phase II any more. The procedure is totally self-adaptive.

### 5.4 Practical Issues

In terms of the above self-learning algorithm, it is worth noting the following questions:

*Why do we Model Immobility?* Tagwatch models the immobility of a tag instead of its mobility because potential moving states are too many to predefine. The “self-learning” behaves in two aspects: first, any surrounding object will contribute to the building of Gaussian models regardless of their

components or shapes. The learned model can be used for future multipath propagation as long as the tags remain in their positions. Second, when a tag moves from one place to another (i.e., state transition), the priorities of its outdated immobility models built for previous positions will be gradually reduced until they are completely removed from the stack.

*When do we Learn Gaussian Models?* Without need of a particular pre-learning offline, Tagwatch is able to quickly accommodate the influence of a new multipath (i.e., due to changes of environment) online. For example, suppose a new surrounding object comes into the scene, which introduce a new unknown multipath, the stationary tag is mistakenly considered as being in motion in Phase I because none of its learned models can match this new resulted phase value. Then it will be scheduled to be intensively read in the Phase II. The system leverages all recent history readings of the tag including those collected in Phase II, to build its immobility models. In this way, the newly emerged Gaussian model is quickly learned after one cycle (e.g.,  $3 \sim 5$  s, see Section 9). When entering the next cycle, the incoming phase can match the learned model and the tag will be correctly classified as being stationary. Thus, our algorithm does not have a “cold start”.

*How to Deal with Highly Dynamic Environment?* As we aforementioned, any motion of a surrounding object moving within a same Fresnel zone creates an equivalent propagation (i.e., distance from the reader to tag passing through the object is equal.), leading to a same Gaussian model. Thus, the number of multipathes are relatively limited. If the system has learned all immobility models incurred by all potential multipathes, it could deal with any interferences from surrounding objects whatever how frequently the environment changes. Thus, the current self-learning algorithm is immune to highly dynamical changes of environment.

*How to Deal with Phase Jumps?* It is known that the phase value  $\theta = (4\pi d/\lambda + \theta_0) \bmod 2\pi$  where  $\lambda$ ,  $d$  and  $\theta_0$  are wavelength, distance between reader and tag, and initial phase. Due to the operator  $\bmod$ , if the expected value  $\mu$  is around 0 (e.g., 0.02), the measured one  $\tilde{\theta}$  may flip to a value close to  $2\pi$  (e.g.,  $2\pi - 0.01$ ), resulting in  $|\tilde{\theta} - \mu| > \xi\delta$  (e.g.,  $|2\pi - 0.01 - 0.02| = 6.2532 > 3 \times 0.1$ ). Actually, the measured value is very close to the expected. This problem emerges because phase values are represented in the base- $2\pi$  system. To resolve this issue, we would apply the minimum distance for the difference detection in practice. The minimum distance equals  $|\theta_1 - \theta_2|$  if  $|\theta_1 - \theta_2| \leq \pi$ , otherwise equals  $(2\pi - |\theta_1 - \theta_2|)$ , e.g.,  $2\pi - |2\pi - 0.01 - 0.02| = 0.03 < 3 \times 0.1$ .

*How to Deal with Reading Exceptions?* We do not assume all tags are always within the range to the reader throughout. Tags are allowed to come in, go out or be temporarily blocked any time. The system independently creates Gaussian models for each tag. If one tag leaves for a long while, the system will remove its models for saving memory. On contrary, the system immediately creates a Gaussian model stack for any newly emerging tags. If a new tag happens to come in the Phase II, the system will read it in the Phase I of next cycle. There may exist a very extreme case that the tag happens to move into the range every Phase II and move out every Phase I, making the system blind to it. As this tag is definitely a mobile one, we can add its EPC to the configuration file, as described in next section.

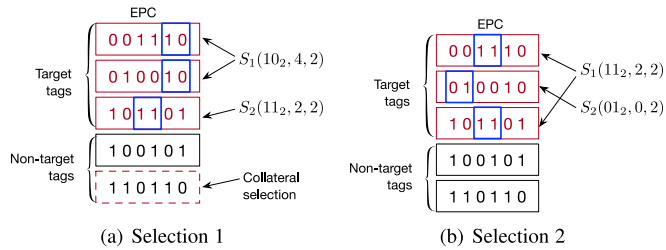


Fig. 8. An example of bitmask selection. (a) Bitmasks  $S_1(10_2, 4, 2)$  and  $S_2(11_2, 2, 2)$  completely cover three target tags. However,  $S_1$  mistakenly covers a non-target tag  $110110_2$ . (b) Bitmasks  $S_1(11_2, 2, 2)$  and  $S_2(01_2, 0, 2)$  cover three target tags without non-target tags.

*Could we Detect Motion with in a Small Region or a Certain Path?* The answer is affirmative. As we will introduce in Section 9, the detection is extremely sensitive to any slight motion. Particularly, 87 and 99 percent of the movement events using RF phase can be successfully detected when the tag is moved away from 2 cm and 3 cm. Thus, our system fails to detect the movement events unless it moves within  $2 \sim 3 \text{ cm}^2$  region or along a  $2 \sim 3 \text{ cm}$  path. Such small error can be ignorable in practice in reference to the tracking application.

*Which Parameters Should User Input in Terms of the GMM Models?* GMM requires to input three user-defined parameters: threshold value  $\xi$ , the learning rate  $\alpha$ , and the number of Gaussian models  $K$ . Other parameters can be determined in the self-learning process. These three user-defined parameters should be manually configured by the engineer based on the actual situation. According to our experience, we advise  $\xi = 3$ ,  $\alpha = 0.1$  and  $K = 5$  by default, which could fit major indoor scenarios.

## 6 PHASE II: TARGET SCHEDULE

In addition to system-determined mobile tags, Tagwatch also allows user to define tags with significant concerns in a configuration file. These tags will be scheduled for reading regardless of whether they are in motion or stationary. We call these tags, including mobile or concerned tags, *targets* or *target tags*. Then, the question becomes: *how can we exclusively read target tags with existences of a large number of stationary tags?* In this work, we leverage a widely-supported Gen2 command, i.e., *Select*, to selectively read target tags.

### 6.1 Selective Reading

The EPC Gen2 air protocol is the most competitive standard that governs RFID systems worldwide. ISO has ratified it as a part of the ISO/IEC 18000 series. Gen2 specifies an important mandatory command, i.e., *Select*, for *selective reading*. Each inventory round starts with a *Select* command. This command is designed to select a subset of those tags that will participate in the upcoming inventory round. The *Select* command contains six mandatory fields and one optional field. The present study focus on four fields: *MemBank*, *Pointer*, *Length* and *Mask*. These fields work together to indicate a selection condition called *bitmask*. In particular, (1) Gen2 divides the tag memory space into four banks for storing password, EPC, TID and customized data. The *MemBank* field specifies the memory bank to which the *Mask* will be compared with. In our system, the *MemBank* is

constantly set to the second bank (i.e., the EPC bank). (2) The *Mask* field contains a bit string for comparison. (3) The *Pointer* field specifies the starting address (i.e., bit number) of the chosen memory bank. (4) The *Length* field specifies the length of the *Mask* used for the comparison. For example, if *MemBank* = 1, *Mask* =  $0101_2$ , *Pointer* = 2 and *Length* = 4, then we select the tags whose bit string starting from 3<sup>rd</sup> bit and ending at (3 + 4)th bit of the EPC equals  $0101_2$ . The tags are divided into two groups: tags that match the bitmask and tags that do not match the bitmask. We only allow the matching tags to respond the query commands (i.e., *Query*) in the subsequent inventory round.

### 6.2 Bitmask Selection

Suppose that  $n'$  target tags from a total  $n$  tags (i.e.,  $n' \leq n$ ) are delivered for selective reading in the second phase. We do not make any assumption on the distribution of the EPCs of target tags as well as those of non-target tags. Any tag can be our target for scheduling. Our task is to seek appropriate bitmasks to cover  $n'$  target tags.

*Challenges.* The search for appropriate bitmasks is non-trivial, and two challenges are encountered. First, a single bitmask is likely unable to cover all target tags in most of time. As shown in Fig. 8a, no common bitmask can concurrently cover the three target tags. We have to choose a group of bitmasks to cover these targets and then perform multiple rounds of selective reading. For simplicity, we use  $S(m, p, l)$  to denote a bitmask with a *Mask*  $m$ , a *Pointer*  $p$ , and a *Length*  $l$ . The *MemBank* field is omitted because this field is fixed to the EPC memory. For example, we select  $S_1(10_2, 4, 2)$  to cover  $001110_2$  and  $010010_2$ , whereas  $S_2(11_2, 2, 2)$  for  $101100_2$ , as shown in Fig. 8a.  $(\dots)_2$  denotes that the number is represented in a binary form. However, such selections are not good because the selected bitmask  $S_1$  collaterally covers a non-target tag with EPC of  $110110_2$ . Thus, the second challenge is to achieve the optimal selection, which can cover all targets and the least non-targets. For example, the second selection (i.e.,  $S_1(11_2, 2, 2)$  and  $S_2(01_2, 0, 2)$ ) shown in Fig. 8b is optimal, which covers three targets without any non-targets.

*Naive Solution.* The naive method directly uses the  $n'$  EPCs of target tags as  $n'$  bitmasks, such that all target tags are covered without including any non-target tag. Therefore, we must start  $n'$  rounds of inventory rounds with  $n'$  *Select* commands. Each inventory round introduces a start-up cost, which affects IRRs. The optimal strategy is to obtain full coverage on target tags with minimum total time cost. We can consider the naive method as the *worst case*. The cost-effective selection may collaterally involve non-target tags as long as their cost is less than in the worst case. We reduce this problem into the *set cover problem*. Before introducing our algorithm, we describe the set cover problem and subsequently define its relation to the bitmask selection problem.

*Classical Set Cover Optimization Problem.* The set cover problem involves selecting a minimum number of sets that contain all the elements in any of the sets in the input. Set cover optimization requires the total cost of the selected sets to be minimal, where each set is assigned a cost. A universal  $\mathcal{U}$  and a family of  $\mathcal{S} = \{S_1, S_2, \dots, S_M\}$  of subset of  $\mathcal{U}$ . Correspondingly, each subset has a weight, denoted by



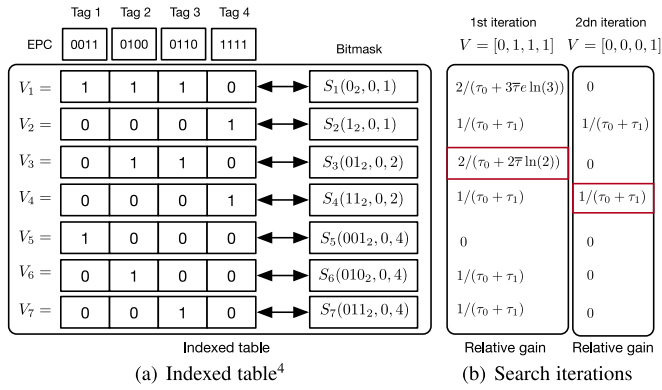


Fig. 9. An example of bitmask selection. (a) The pre-built indexed table<sup>4</sup> over four tags in the scene. (b) An execution of the search algorithm, where the input indicator bitmap is equal to  $[0, 1, 1, 1]$ . Finally,  $S_3$  and  $S_4$  are selected.

$\{c_1, c_2, \dots, c_M\}$ . The objective is to find a group of subsets  $\mathcal{I} \subseteq \mathcal{S}$ , such that minimizing  $\sum_{S_i \in \mathcal{I}} c_i$  subjecting to  $\cup_{S_i \in \mathcal{I}} S_i = \mathcal{U}$ .

**Bitmask Selection as Set Covering.** Within our problem domain, the universal set  $\mathcal{U}$  contains the EPCs of  $n'$  target tags, i.e.,  $\mathcal{U} = \{\text{EPC}_1, \text{EPC}_2, \dots, \text{EPC}_{n'}\}$ . Each bitmask corresponds to a subset  $\subseteq \mathcal{U}$ , which contains all EPCs covered by the bitmask. Thus, we use a bitmask to denote tag set that it covers. To determine the number of bitmasks (or subsets) do we have, let  $L$  be the bit length of the EPC number (e.g., 96 or 128 bits). The bitmasks can be set with different starting addresses and lengths, hence, we have  $\sum_{l=1}^L (L-l+1)2^l = 2^{L+2} - 2L - 4$  candidate bitmasks in global space. The equation obtains a possible bitmask by sliding a mask from the first position to the last for each length  $l$ . For example, If  $L = 96$ , then  $3.1691 \times 10^{20}$  candidate bitmasks (or subsets) are available, which is too large to be accepted. We only focus on  $n'$  target tags rather than on the entire EPC number space. Therefore, the majority of the subset induced by the global bitmasks are empty, i.e., containing none targets. The candidate bitmask  $S_i$  is selected only when it can cover a target tag. Thus, the actual total number of candidate bitmasks is equal to  $\sum_{l=1}^L (L-l+1)n' = n'L(L+1)/2$ .  $n'$  should be small. Thus, the candidates are controllable. Let  $S_i$  denote a bitmask and the tag set that it covers. Then,  $|S_i|$  is the number of tags covered by this bitmask. The time cost to collect the tags covered by  $S_i$  can be approximated to  $C(|S_i|)$ , as indicated in Definition 1. Therefore, our problem can be written as the following optimization formula:

$$\begin{aligned}
 & \text{minimize} && \sum_{S_i \in \mathcal{I}} C(|S_i|) \\
 & \text{subject to} && \mathcal{U} \subseteq \bigcup_{S_i \in \mathcal{I}} S_i.
 \end{aligned} \tag{15}$$

We reverse the subject condition because the  $S_i$  may contain non-target tags. At first glance, we should select a group of bitmasks purely covering all target tags without non-target tags. Such strategy may not be optimal because it may require excessive bitmasks to achieve its purpose. Consequently, start-up cost is increased, which leads to higher

4. This indexed table only considers the bitmasks starting from the first bit due to space limit. The actual table is larger than this one.

total cost. In addition, the worst cost is always equal to  $C(n')$  when taking  $n'$  inventory rounds that use EPCs of target tags as bitmasks. If the cost of “optimal” selection is higher than  $C(n')$ , we should adopt the worst option.

### 6.3 Bitmask Based Schedule

Set cover optimization is an NP-hard problem. Thus we design a greedy algorithm to search for bitmask selection.

**Preprocessing.** Before searching, we build an index table to associate candidate bitmasks with current EPCs, as shown in Fig. 9a. We arrange all the current tags, including target and non-target tags, based on their EPC values. The left column of the table is an indicator bitmap that shows whether the tag is covered by the right bitmask. For example,  $V_1 = [1, 1, 1, 0]$  because the first three EPCs of the tags are covered by the bitmask of  $S_1(0_2, 0, 1)$ . We traverse all possible bitmasks and generate the corresponding indicator bitmaps that involves all the tags in the scene. We abandon the rows with zero indicators, which do not cover any tag. We also merge the rows whose indicator bitmaps are identical by randomly using one of them because the coverage ranges of these bitmasks are equal. When the indexed table is built, only an incremental update is required, i.e., deleting outgoing tags or adding new ones at the end of the assessment phase.

**Searching for Optimal Bitmasks.** Tagwatch uses an iterative searching algorithm. An input indicator bitmap  $V$  shows the target tags that should be covered. Each iteration involves the following steps:<sup>5</sup>

- (Step 1) *The relative gain for each bitmask is calculated.* The relative gain  $R(S_i)$  for the bitmask  $S_i$  is defined by

$$R(S_i) = \frac{|V_i \& V|}{C(|V_i|)}, \tag{16}$$

where  $|V_i \& V|$  is the cardinality of the result of a bitwise “AND” on two bitmaps. It denotes the number of target tags that can be covered using  $S_i$ , and is considered as the absolute gain of  $S_i$ .  $C(|V_i|)$  is the inventory cost if  $S_i$  is used for the selective reading, i.e., inventory cost for reading  $|V_i|$  tags. This cost is considered as the bitmask price.

- (Step 2) *The bitmask with the highest relative gain is selected.* We expect the selected bitmask to identify more target tags with less price (i.e., covering less non-target tags). A draw can be resolved by random selection.
- (Step 3) *The input indicator bitmap is updated.* Suppose we select the  $j$ th bitmask in Step 2, then  $V = V - (V \& V_j)$ , i.e., the new input bitmap is changed to indicate the target tags that are not covered in this iteration.
- (Step 4) *Return to Step 1 for the next iteration.* The search process is terminated when  $V = 0$ .

A search example is provided in Fig. 9b where the original input indicator bitmap  $V = [0, 1, 1, 1]$ , i.e., the last three tags are target whereas the first tag is non-target tag. In the

5. It is worthy noting that the searching is performed at the server side and its calculation cost does not affect the air communication.

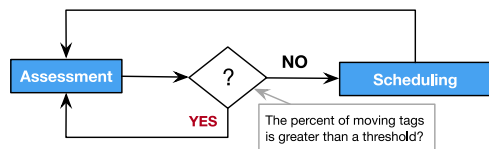


Fig. 10. Processing in special case

first iteration,  $R(S_1) = \underline{2}/(\tau_0 + 3\overline{7}e \ln 3)$  because two common “1”s are between  $V$  and  $V_1$  (i.e., gain is equal to 2), and a total of three tags must be collected (i.e., inventory cost is equal to  $\tau_0 + 3\overline{7}e \ln 3$ ). At the end of the first iteration, we select bitmask  $S_3$  because it has the highest relative gain compared among all the bitmasks. Then, we update  $V = V - (V \& V_3) = [0, 0, 0, 1]$  as the input of the second iteration and select  $S_4$  for the second bitmask. After the second iteration,  $V = 0$  and the search process is terminated.

## 7 DISCUSSION

In this section, we discuss some practical issues when applying Tagwatch into the real environment.

### 7.1 Dealing with Exceptions

Although our scheduling algorithm could significantly improve the reading rate of mobile tags, it still introduces some extra cost to achieve this goal as traditional optimization algorithm does. This extra time costs are consumed on broadcasting multiple `SELECT` commands and the exceptional readings on collateral tags. Our empirical study shown in Section 3 has demonstrated that the percent of moving tags is usually below than 20 percent. However, the exceptional cases might occur for a short while, i.e., a large percent of tags start moving. In such situation, the extra costs might be over than the gains that Tagwatch can provide.

So, how could we deal with the exceptional case? Our baseline is the time consumption taken on reading all tags as the today’s RFID system does. If the percent of moving tags exceeds some threshold such that the expected total time cost is over than the baseline, our system should automatically switch to the old-fashion-reading all tags, keeping the total cost at the baseline level. Thanks to the design of two-phase reading, the switching can be very simply accomplished: skipping the second phase when the system finds that the expected cost exceeds the baseline in the end of the first phase. The final effect is equivalent to prolonging the first phase to ‘infinity’. When the percent falls back, our system re-activates the second phase. Fig. 10 illustrates this algorithm. Note that the time cost can be predicated by using Eqn. (4) where the  $n$  is substituted by the number of moving tags.

### 7.2 Extending to Multiple Readers/Antennas Scenario

Other wide scenes such like supermarkets or warehouse may require to deploy multiple readers for a full coverage. For example, we deploy four CheckPoints to cover the target space in our pilot study, and each CheckPoint contains four antennas. The interference among these readers would be an issue as studied in our prior work [20]. Following the common practice [21], we adopt the most common solution for the multiple reader scenario, namely, using a centralized scheduler running at the server arranges the operations of

readers in different time slots. Any two adjacent readers are scheduled in two different time slots. On the other hand, today’s commercial reader usually supports 4 antennas and can be extended to connect 32 antennas at most with the help of a hub [22]. Each reader schedules its each antenna in a similar time-sharing fashion. Thus, the case of multiple reader can be reduced to the case of multiple antennas. From the high level, we can consider that each antenna operates at its individual time slots without interferences from others.

How could our two-phase reading protocol adapt the multiple antenna scenario? This question can be answered in terms of two kinds of applications in terms of their purpose [20] as follows:

- *Location-sensitive case:* The location-sensitive applications care about tags’ locations in this case. The tags located in the common regions covered by multiple antennas must be read by these antennas individually, because a tag’s channel parameters estimated by different antennas are totally different. To adapt this case, the immobility model must be built for each pair of reader and tag independently during the assessment phase; the scheduling of mobile tags are also performed individually for each antenna.
- *Location-insensitive case:* On the contrary, the location-insensitive applications do not care about tags’ locations in this case. The tags located in the common region only need to be read by one of these multiple antennas. We could give more optimization for this case. Specifically, during the assessment phase, each antenna still builds the immobility mode for each tag individually even if it is located in the common region. The tag is considered as being moving as long as one antenna detects its motion. During the scheduling, these antennas covering a common tag should jointly search the optimal bitmask, making the common tags are read once with minimum cost.

## 8 IMPLEMENTATION

We adopt the ImpinJ Speedway R420 reader [23] without making any hardware or software modification. The reader is compatible with EPC Gen2 standard. Four reader antennas with circular polarizations are used to provide a gain of 8 dB in two directions. Four types of tags from Alien Corporation [24] are used. We adopt the ImpinJ LLRP Tool Kit (LTK) to communicate with the reader. LLRP [25] is another protocol specified by EPCglobal that works with Gen2. It is designed to deliver Gen2 parameters from a client to a reader.

*Prototype with LLRP.* LLRP specifies reader operation using ROSpec, which is an XML document that encapsulates parameters of selective reading (e.g., MemBank, Pointer, Mask, Length, etc). Fig. 11 shows a typical ROSpec, where three bitmasks are configured for three selective readings. Each ROSpec is composed of several AISpecs, each of which is used for an antenna setting. An AISpec consists of more than one C1G2Filters. The filters function as the bitmasks. We can set multiple bitmasks by adding multiple C1G2Filters or multiple AISpecs. We adopt the second method by default. After receiving a ROSpec, the reader sequentially starts selective reading.

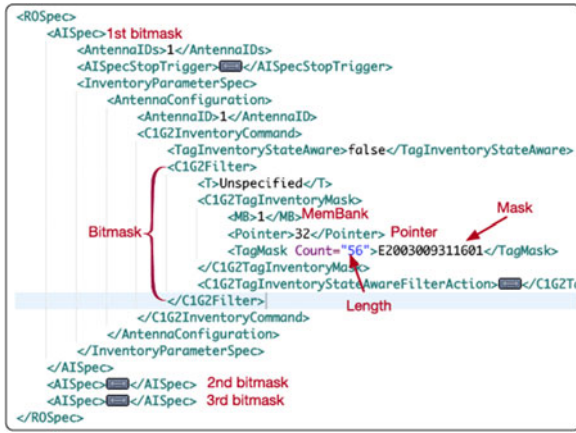


Fig. 11. An example of ROSpec. This spec defines three bitmasks for scheduling.

**Parameter Choice.** Tagwatch has several parameters. The key parameters are employed as follows. (1) Reading model: we utilize the least square method to estimate  $\tau_0$  and  $\bar{\tau}$  for the inventory cost (refer to Eqn. (8)) by conducting the empirical experiments. Consequently, practical  $\tau_0 = 19$  ms and  $\bar{\tau} = 0.18$  ms. (2) Cycle length: all tags are allowed to be read once in the first phase, thereby the interval of Phase I dynamically depends on the total number of tags. We fix the length of Phase II to 5 seconds. The upper applications can adjust the length of Phase II according to their requirements. (3) For motion detection, the learning rate  $\alpha$ , the number of Gaussian models  $K$  and the threshold  $\xi$  are set to 0.001, 8 and 3.0, respectively, by default.

## 9 EVALUATION

We start a few experiments that provide insight into the operation of the system. We evaluate the performance of the two phases in detail in the following sections.

### 9.1 Evaluation of Phase I

The performance of the motion assessment that occurred in the first phase is first evaluated. We focus on accurately detecting movements with a low false positive rate in a timely manner. To represent false positives, we deploy 100 stationary tags in our office. Approximately 10 individuals work in the room. These people will introduce additional multipath propositions when approaching the tags. We monitor these stationary tags for 48 hours using a reader and collect over 2 million readings. The trace is used as the input to build the immobility model of each tag and to test

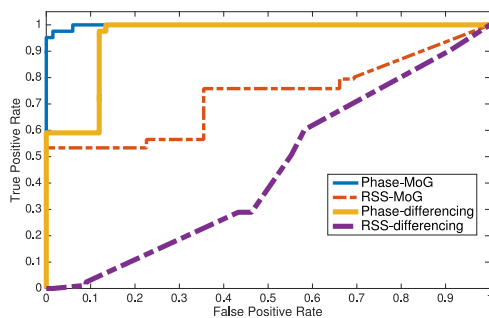


Fig. 12. Detection accuracy.

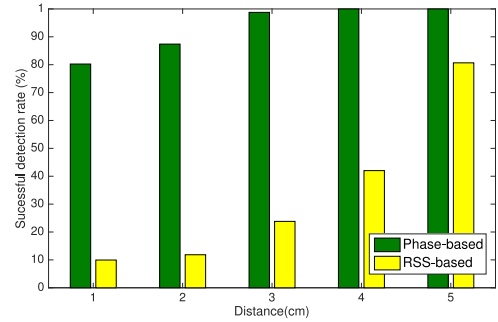


Fig. 13. Detection sensitivity.

for false positive rates. To measure the actual positive rate (i.e., accuracy), we attach a tag on a toy train that moves along an oval track with a radius of 20 cm at a constant speed of 0.7 m/s. All the readings are considered collected from a mobile tag.

**Detection Accuracy.** First, we investigate the detection accuracy. We present accuracy using the *Receiver Operating Characteristic* (ROC) curve, which is composed of True Positive Rate (TPR) and False Positive Rate. We obtain pairs of TPR and FPR by adjusting the detection threshold (e.g.,  $\xi$  in Tagwatch). For comparison, we also use three other methods for baselines. Fig. 12 presents the ROC results, where *Phase/RSS-differencing* is the naive method that simply compares the incoming RF phase/RSS with the last value. *Phase/RSS-MoG* utilizes the Mixture of Gaussian (MoG) of phase/RSS to model the immobility of a tag. Consequently, both RF phase-based detection techniques are better than RSS-based methods. In particular, given a FPR of 0.2, Phase-MoG and Phase-differencing achieve 0.99 above TPRs, whereas RSS-MoG and RSS-differencing achieve only 0.53 and 0.12 TPRs, respectively. The RF phase is more sensitive to the motion than RSS. Thus, using phase as a motion indicator is superior to using RSS. However, regardless of which physical indicator (e.g., RF phase or RSS) is used, MoG-based approaches can always control FPRs at a relatively acceptable level compared with differencing. These results show that the multipath effect significantly influences the final received backscatter signals, and MoG effectively depicts this dynamic nature. For example, we can find an appropriate detection threshold to achieve  $\geq 0.95$  TPR while  $\leq 0.1$  FPR using Phase-MoG.

**Detection Sensitivity.** Second, we further study the sensitivity of Tagwatch to motion. Our objective is the timely and accurate monitoring of tag movement, even if its displacement is minimal. During this stage, we move a tag away in a random direction with a displacement ranging from 1 cm ~ 5 cm. We conduct the experiment 20 times with the same displacement setting. We apply the successful detection rate, i.e., the ratio of the number of successful detections to the total test times as a metric to evaluate sensitivity. Fig. 13 presents the results compared with the results of the RSS-based method. The figure shows that we can successfully detect 87 and 99 percent of the movement events using RF phase when the tag is moved away from 2 cm and 3 cm, whereas only 9 and 18 percent of the events are detected using RSS. Even when the tag is moved away to 5 cm ( $\approx 1/4$  wavelength), the RSS-based method achieves a 76 percent successful rate. In theory, phase value is proportional to

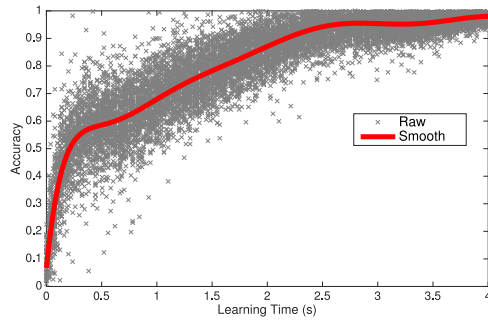


Fig. 14. Learning curve.

twice the distance between the reader and the tag. Thus, when a 1 cm displacement is introduced, an actual change in distance of 2 cm will occur with regard to the RF phase. These characteristics provide a natural amplifier for the RF phase, thereby making it sensitive to movements.

*Learning Curve.* Third, we conduct a group of experiments to answer the question: how long does the system need to build a stable Gaussian model? It seems like the system requires a “slow” and “cold” start. In the experiments, we keep a tag stationary and let a person work around. Total one minute readings are collected. We model the tag’s immobility using the first 10 ms, 20 ms, 30 ms, ..., and 10,000 ms trace data respectively, and then investigate the detection accuracy using the subsequent 100 ms trace as test data. We say the system conducts a correct detection when the test reading matches one of the immobility Gaussian models. Fig. 14 shows the detection accuracy as a function of time. It suggests that we can achieve 70 and 90 percent of detection accuracy when fed with 1.49 s trace (i.e., including 67 readings) and 2.9 s trace (i.e., including 130 readings). Therefore, one-cycle readings (i.e., 5 s) are sufficient to stably create a newly emerging Gaussian model, providing a “quick start” for the self-learning.

## 9.2 Evaluation of Phase II

To understand how Tagwatch operates in the second phase, we deploy a total of  $4 \times 40$  tags with *random* EPCs within the ranges of 4 reader antennas (i.e., each antenna covers 40 tags). In addition, we use the configuration file to directly label target tags to eliminate the influence from the first phase. IRR is used as the metric (i.e., Hz). Each experiment with the same setting is repeated 50 times, and the average result is reported.

*Schedule Feasibility.* To make an intuitive understanding of the scheduling, we show two specific examples in

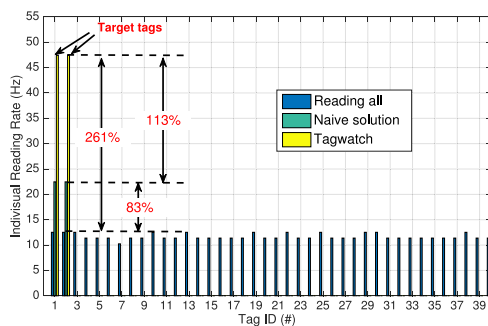


Fig. 15. Schedule feasibility: 2/40.

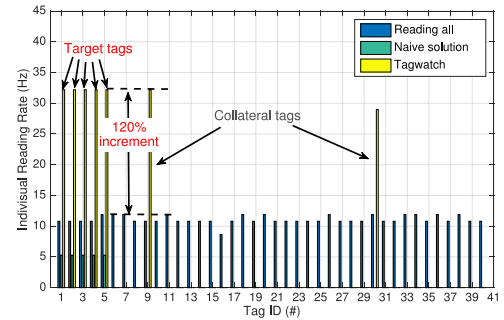


Fig. 16. Scheduling feasibility: 5/40.

Figs. 15 and 16, where 2 and 5 tags out of total 40 tags are selected as our targets. In the figure, each tag is associated with three bars that correspond to the IRRs using reading all, Tagwatch, and naive rate-adaptive solution (which simply selects targets’ EPCs as bitmasks). From Fig. 15, we observe that the mean IRRs of the targets (i.e., tag #1 and #2) are approximately 13 Hz in the solution of reading all. By contrast, if Tagwatch is applied, the IRRs of the targets will increase by 261 percent (i.e., from 13 Hz to 47 Hz), which is thrice higher than in “all reading”. Naive solution also gives 83 percent of increment (i.e., from 13 to 24 Hz). Meanwhile, the IRRs of non-targets drop to zero (note that we focus on the performance of Phase II; thus, the calculation rate excludes the readings obtained in Phase I). From Fig. 16, the similar results are observed except two points: first, tag #9 and #30 are collaterally involved into Phase II. Even so, Tagwatch still offers 120 percent of increment to target tags. Second, the IRRs that naive solution obtained is even lower than that of reading all, because its cost have counteracted its gain. More discussions about IRR will be introduced in the overall evaluation. These two experiments validate that it is completely feasible and effective to perform selective reading through bitmask enabled selection.

*Schedule Cost.* Tagwatch introduces extra time cost on the functions of motion assessment and bitmask selection, i.e., the interval during the end of Phase I and the start of Phase II. This extra time cost may affect the real-time of the system. We slice these extra time consumption from 50,000 cycles, by calculating the time difference between the first reading in Phase II and the last reading in Phase I for each cycle. The CDF of time is illustrated in Fig. 17. We find our additional functions introduce less than 4 ms in the 50 percent of cycles, and 6 ms in the 90 percent of cycles. Compared with the 5 seconds of each cycle, such extra consumption can be ignored.

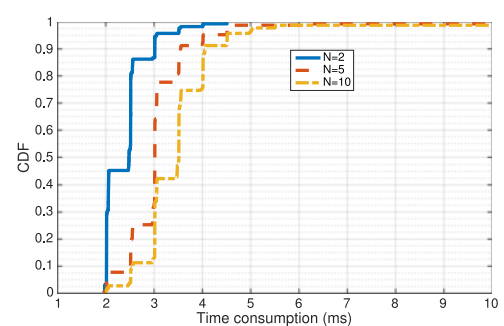


Fig. 17. Schedule cost.

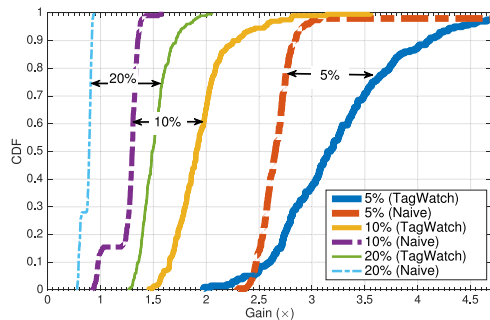


Fig. 18. IRR gain. This figures shows the IRR gains of adaptive-reading in terms of different percents of mobile tags.

### 9.3 Overall Evaluation

Finally, we present the overall performance of Tagwatch, including two phase readings, by studying the IRR gain and a tracking application.

We evaluate the overall performance through the IRRs of mobile tags. We defined *IRR gain* as the average ratio of IRR obtained by rate-adaptive reading to the IRR obtained by reading all. We present IRR gains with respect to the percent of mobile tags, as shown in Fig. 18. In the figure, we compare two rate-adaptive solutions: one is our Tagwatch, while another is the naive solution that directly uses mobile tags' EPCs as bitmasks for selective reading. The results of two solutions are plotted in a pair considering a same percent of mobile tags. For each percent, we vary the total number of tags to 50, 100, 200, 300 and 400, and perform 1,000-cycle reading for each setting. All mobile tags are put on a spinning turntable. From the figure, we have the following findings:

- First, when 5 percent of tags are moving, Tagwatch can improve their IRRs by  $3.2\times$  in 50 percent of trials and by  $4\times$  in 10 percent of trials. Naive solution can also achieve a gain of  $2.6\times$  on median. This result suggests that any kind of adaptive-reading solutions can bring efficiency benefits when a few mobile tags exist.
- Second, when 10 percent of tags are moving, the IRR gain that Tagwatch achieves has a median of  $1.9\times$  with a standard deviation of  $0.29\times$ . In particular, it can achieve a gain of  $2.3\times$  in 10 percent of trials. This result suggests that the IRR gain decreases as the percent of mobile tags increases, because more tags participate in the Phase II, making the contention more intense among mobile tags. By contrast, the IRR gain of naive solution is up to  $1.5\times$ . Clearly the algorithm of bitmask selection used in Tagwatch is superior to broadcasting all.
- Third, when the percent reaches 20 percent, the IRR gain of Tagwatch is getting close to 1 (e.g.,  $1.5\times$  on average), which means nothing improved. In particular, naive solution has a median gain of  $0.8\times$ , namely, the actual IRR is lower than that of reading all. Because too much `Select` commands are broadcasted in Phase II, which totally counteracts its gain.

In summary, adaptive reading is completely logical when the mobile tags are a minority (e.g.,  $< 20\%$ ). It would provide nearly  $4\times$  gain without modification on firmware or hardware, which was hardly achieved in prior work.

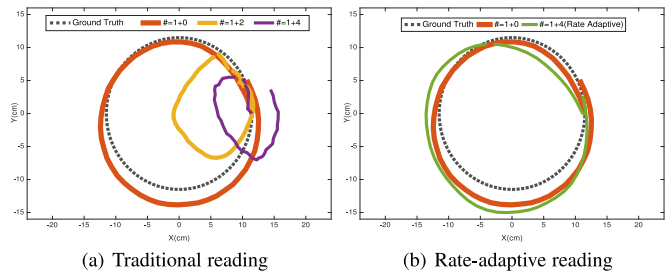


Fig. 19. Tracing a mobile object using one RFID tag in company with different numbers of stationary tags. (a) Traditional reading with zero, two and four stationary tags. (b) Rate-adaptive reading with four stationary tags.

However, as any scheduling will introduce additional cost to start-up and collateral tags, it should notice that the cost may counteract its gains when there are a large number of mobile tags (e.g.,  $> 20\%$ ). In such situation, we should simply switch back to the old fashion: reading them one by one.

### 9.4 Application Study

Finally, we take the tracking of mobile tags as an application to study the effect of Tagwatch. In the scene, we deploy four antennas around the surveillance region at the points of  $(\pm 5\text{ m}, \pm 5\text{ m})$ . We attach a tag on a mobile toy train and allow it to move along a circular track. The trajectories are recovered using our prior tracking solution [16], namely Differential Augmented Hologram. The experiments are conducted in three different cases:  $(1 + 0)$  (i.e., # of mobile tags + # of stationary tags),  $(1 + 2)$  and  $(1 + 4)$ . We fix the initial position at a known point to improve comparison.

To visually illustrate the impact of reading rate on motion tracking, Fig. 19a shows an example<sup>6</sup> in which the trajectory of a tagged toy train moving along a circular track is recovered. Moreover, we place two and four stationary tags beside the track. Correspondingly, without using Tagwatch, the mean accuracy deteriorates from  $1.8 \pm 0.72\text{ cm}$  in case  $(1 + 0)$  to  $6\text{ cm} \pm 3.55\text{ cm}$  in case  $(1 + 2)$  and  $10.6 \pm 5.46\text{ cm}$  in case  $(1 + 4)$  due to the channel competition from the stationary tags. By contrast, the mean tracking accuracy remains at  $3.34 \pm 1.42\text{ cm}$  with rate-adaptive reading even when four interfered stationary tags are placed beside the track. Even with the presence of four other tags, the mean accuracy (i.e.,  $3.34\text{ cm}$ ) of the recovered trajectory remains as good as that obtained without a static tag. The minimal accuracy loss comes from the apportioned readings in the first phase. Thus, this application study shows that Tagwatch can exactly identify the mobile tag as well as adaptively read it.

## 10 PILOT STUDY

This section presents a pilot study that applies Tagwatch to a sorting task in a real scenario.

*Background.* We collaborate with a medium-sized logistics company to develop an assisted manual sorting system. The current sorting system of the company is semi-mechanized, wherein all packages are transported through conveyors,

6. Despite a single moving tag shown in the example, our system can deal with the case where multiple mobile objects present.

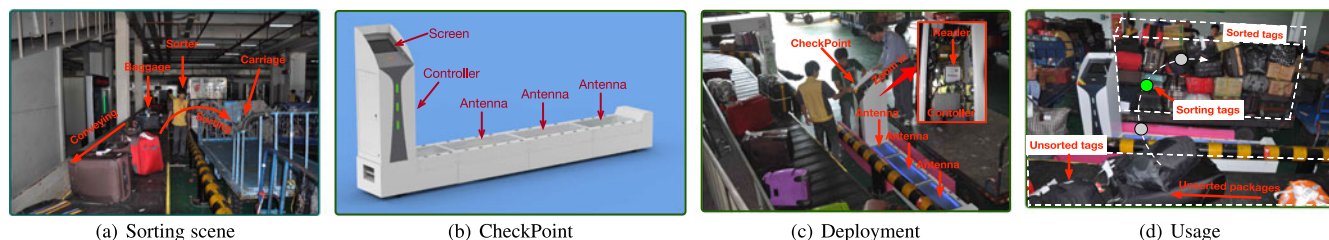


Fig. 20. RFID assisted sorting system. (a) The sorter finds, picks up and moves the packages from the carousel to the carrier vehicle. (b) The blue print of CheckPoint. (c) The deployment of CheckPoint on site. (d) The usage of CheckPoint.

but the system relies on human ability for sorting. Every day, tens of thousands of packages enter the conveyor system to be transported by a shared conveyor system in a sorting station, thereby mixing the different packages. These packages are then manually separated from the conveyor to carrier vehicles based on their destinations. Fig. 20a shows the sorting scene, where a sorter finds and selects packages from the sorting carousel (i.e., a circular conveyor used as a buffer, such as the baggage claim area in an airport arrival hall.). The sorters stand by the carousel along the conveyor to find and carry packages. This process is error-prone, and sorters may select the wrong package. To help improve sorting accuracy and efficiency, we introduce RFID technology to provide package tracking and checking services, and thus, eliminate wrong sorting. This system addresses many practical issues. In this work, we mainly focus on one key issue, that is, to guarantee that the packages are correctly sorted.

*Customized Devices.* To provide a double-checking service, we customize several devices as follows. We first upgrade the tag printer by installing a small RFID reader inside it. Meanwhile, an RFID tag is embedded into each package tag. A unique EPC number is written when the package tag is printed. Second, we design and customize a checking device, called *CheckPoint* (Fig. 20b). This L-shaped device is deployed between the carousel and a carrier vehicle. It contains four reader antennas on the bottom, which are covered by unbreakable glass to allow sorters to stand on them. An RFID reader and an industrial computer are arranged on the side container. Figs. 20c and 20d show the deployment and the usage of CheckPoint that a package is automatically double-checked<sup>7</sup> when it is being moved from the carousel to the carrier vehicle. If the sorted package does not belong to the vehicle, CheckPoint will turn on the read light to alert sorters. The system has been deployed for over 3 years and monthly tests have been conducted in two sorting stations. Each deployment contains 22 tag printers (on check-in counters) and 4 CheckPoints (beside the carousels). To date, our pilot study has consumed over 100,000 RFID tags.

*Checking Service.* An RFID system can read tags within the range of over 10 m without requiring LOS, which makes reading considerably faster than scanning barcodes. However, these two advantageous features create an uncontrollable reading zone, thereby leading to stray reads or unwanted reads [26]. In our case, the four reader antennas of CheckPoint cover many sorted tags that remain statically in the vehicle and a few unsorted tags moving on the

7. The first check is manually conducted by the sorter himself when the package is being picked.

conveyor, as shown in Fig. 20d. We have to identify the actual sorting tags, which are moved across above the device. The movement trends and directions of unsorted and sorted tags are discriminative; thus, they can be easily distinguished by leveraging the physical layer signal patterns. The detailed algorithm can refer to our previous work [16]. However, such approach requires many samplings of the states of mobile tags, e.g., at least 10 readings to represent the moving trend. However, the individual reading rates of sorting tags is extremely low (e.g., 2 ~ 3 reads) because of the channel competition among many sorted and stationary tags.

*Evaluation Results.* To exclude interferences from these sorted tags, we apply Tagwatch to the checking service. Consequently, the error of detecting and sorting behaviors (considering both correct and incorrect behaviors) is reduced from 43 to 6 percent (Fig. 21) due to adaptive reading, which increase the readings of sorting tags by 2 ~ 3 $\times$ . In the current working procedure, the sorters must recheck all sorted packages finally in the end of sorting and before delivery. This step is expected to be eliminated in the future after using Tagwatch. However, during our pilot study, we reserve this step for two reasons: obtaining ground truth and in case of an unexpected failure in our system. The rechecking results suggest that the sorting error decreases from 7 to 0.95 percent. Therefore, the sorters are aware of and can immediately correct the approximately 6 percent erroneous sorting after the warnings from devices are noticed.

## 11 RELATED WORK

A variety of works have been proposed to improve reading rate from two layers. Actually, Tagwatch can work with any of the following designs.

*Physical Layer Study.* The first group of studies aims to transmit RFID tags in parallel and separate collided

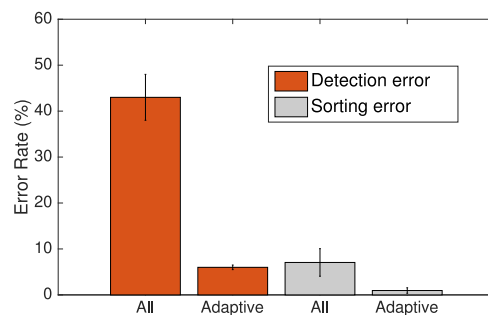


Fig. 21. Sorting error rate. The labels of 'all' and 'adaptive' mean reading all and adaptive reading respectively.

transmission in the physical layer. For example, Buzz [9] decodes tag collisions bit by bit. It assumes the linear combination of the *static* channel coefficients of reflecting tags independent of coexisting tags. It cannot apply to moving tags which change their coefficients every second. The linear additional-based scheme proposed in [27] also holds the same assumption. BST [10] enables concurrent transmission by leveraging intra-bit multiplexing of on-off keying signals. It detects signal edges when the distances between consecutive symbols exceed a predefined threshold. BiGroup [11] recovers collisions without modifying COTS tags, but instead, changes the readers. Other designs [8], [28], [29] also recover the collisions of up to two concurrent tags using predefined preambles and stringent tag synchronization. In [30] and [31], orthogonal codes for RN16 are designed for collision recovery.

*Link Layer Study.* The second group of studies aims to design considerably efficient anti-collision protocols in the link layer [12]. RFID tags rely on the centralized schedule of readers to avoid tag collisions. For example, [32] and [33] propose schemes based on frequency division multiple access and space division multiple access, respectively. Time division multiple access (TDMA) protocols constitute the largest group of anti-collision protocols [13], [14], [15], [34], most of which are variants of DFSA, such as Q-adaptive. Many works have also proposed tree-based protocols [35], which are also a category of TDMA protocols. These protocols operate by splitting responding tags into multiple subsets using a random number generator. In [36] basic tree splitting is presented. In [37] and [35] adaptive binary tree splitting, which dynamically adjusts the tree based on collision history, is proposed.

## 12 LIMITATIONS & CONCLUSION

In this work, we present a rate-adaptive reading for the case when the percent of mobile tags is relatively lower.

*Limitations.* Is it possible to improve the reading rate even if the percent exceeds 20 percent? The answer is negative to the application-layer solutions. The channel resource is limited. The only way of improving reading rates for all tags is to extend the channel utilization in the physical layer. Parallel decoding might be a potential solution, which has been studied for years. Unfortunately, today's commercial readers expose very limited APIs for us to further optimize the performance. Especially no APIs are available for us to control the physical layer.

*Conclusion.* In this work, we present Tagwatch for the rate-adaptive reading of mobile tags through selective reading. A key innovation is the two-phase reading design, in which mobile tags are read for a relatively long time in the second phase, without competition from stationary tags, thereby improving their IRRs at a significant level. The Tagwatch system has not only been tested and used in practical applications, but will also open a wide range of exciting opportunities.

## ACKNOWLEDGMENTS

The research is supported by GRF/ECS (NO. 25222917), NSFC General Program (NO. 61572282), Shenzhen Basic Research Schema (NO. JCYJ20170818104855702), and Alibaba

Innovative Research Program. We thank all the reviewers for their valuable comments and helpful suggestions, and particularly thank Prof. Samir R. Das for the shepherding.

## REFERENCES

- [1] G. M. Gaukler, "Item-level RFID in a retail supply chain with stock-out-based substitution," *IEEE Trans. Ind. Inform.*, vol. 7, no. 2, pp. 362–370, May 2011.
- [2] L. Shanguan, Z. Zhou, X. Zheng, L. Yang, Y. Liu, and J. Han, "ShopMiner: Mining customer shopping behavior in physical clothing stores with COTS RFID devices," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, 2015, pp. 113–125.
- [3] J. Wang, D. Vasitis, and D. Katabi, "RF-IDraw: Virtual touch screen in the air using RF signals," in *Proc. ACM Conf. SIGCOMM*, 2014, pp. 235–246.
- [4] M. Gowda, A. Dhekne, S. Shen, R. R. Choudhury, L. Yang, S. Golwalkar, and A. Essanian, "Bringing IoT to sports analytics," in *Proc. USENIX Symp. Netw. Syst. Des. Implementation*, 2017, pp. 499–513.
- [5] J. Wang, F. Adib, R. Knepper, D. Katabi, and D. Rus, "RF-compass: Robot object manipulation using RFIDs," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 3–14.
- [6] T. Wei and X. Zhang, "Gyro in the air: Tracking 3D orientation of batteryless internet-of-things," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 55–68.
- [7] L. Yang, Y. Li, Q. Lin, X.-Y. Li, and Y. Liu, "Making sense of mechanical vibration period with sub-millisecond accuracy using backscatter signals," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 16–28.
- [8] C. Angerer, R. Langwieser, and M. Rupp, "RFID reader receivers for physical layer collision recovery," *IEEE Trans. Commun.*, vol. 58, no. 12, pp. 3526–3537, Dec. 2010.
- [9] J. Wang, H. Hassanieh, D. Katabi, and P. Indyk, "Efficient and reliable low-power backscatter networks," in *Proc. ACM SIGCOMM Conf. Appl. Technol. Archit. Protocols Comput. Commun.*, 2012, pp. 61–72.
- [10] P. Hu, P. Zhang, and D. Ganesan, "Leveraging interleaved signal edges for concurrent backscatter," in *Proc. 1st ACM Workshop Hot Topics Wireless*, 2014, pp. 13–18.
- [11] J. Ou, M. Li, and Y. Zheng, "Come and be served: Parallel decoding for COTS RFID tags," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 500–511.
- [12] D. K. Klair, K.-W. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," *IEEE Commun. Surv. Tut.*, vol. 12, no. 3, pp. 400–421, Jul.–Sep. 2010.
- [13] Z. Bin, M. Kobayashi, and M. Shimizu, "Framed ALOHA for multiple RFID objects identification," *IEICE Trans. Commun.*, vol. 88, no. 3, pp. 991–999, 2005.
- [14] J.-R. Cha and J.-H. Kim, "Novel anti-collision algorithms for fast object identification in RFID system," in *Proc. IEEE 11th Int. Conf. Parallel Distrib. Syst.*, 2005, pp. 63–67.
- [15] G. Khandelwal, K. Lee, A. Yener, and S. Serbetli, "ASAP: A MAC protocol for dense and time-constrained RFID systems," *EURASIP J. Wireless Commun. Netw.*, vol. 2007, no. 2, pp. 3–3, 2007.
- [16] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices," in *Proc. 20th Annu. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 237–248.
- [17] L. Yang, Q. Lin, X. Li, T. Liu, and Y. Liu, "See through walls with COTS RFID system!" in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 487–499.
- [18] H. D. Hristov, *Fresnel Zones in Wireless Links, Zone Plate Lenses and Antennas*. Norwood, MA, USA: Artech House, Inc., 2000.
- [19] H. Wang, D. Zhang, J. Ma, Y. Wang, Y. Wang, D. Wu, T. Gu, and B. Xie, "Human respiration detection with commodity wifi devices: Do user location and body orientation matter?" in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 25–36.
- [20] L. Yang, J. Han, Y. Qi, C. Wang, T. Gu, and Y. Liu, "Season: Shelving interference and joint identification in large-scale RFID systems," in *Proc. IEEE INFOCOM*, 2011, pp. 3092–3100.
- [21] D. W. Engels and S. E. Sarma, "The reader collision problem," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 2002, pp. 6–pp.
- [22] RAIN RFID Reader Antenna Hub. 2012. [Online]. Available: <https://support.impinj.com/hc/en-us/articles/202755698-Speedway-Antenna-Hub-Product-Brief-Datasheet>

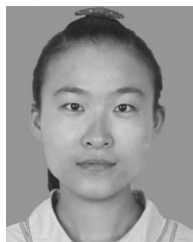
- [23] Impinj, Inc, 2017. [Online]. Available: <http://www.impinj.com/>
- [24] Alien, 2017. [Online]. Available: <http://www.aliantechnology.com>
- [25] EPCglobal. (2010). [Online]. Available: [https://www.gs1.org/sites/default/files/docs/epc/llrp\\_1\\_1-standard-20101013.pdf](https://www.gs1.org/sites/default/files/docs/epc/llrp_1_1-standard-20101013.pdf)
- [26] J. Suomela, "Identifying and controlling stray reads at RFID gates," Ph.D. dissertation, School Electrical Eng., Aalto Univ., Espoo, HKI, 2012.
- [27] D. Shen, G. Woo, D. P. Reed, A. B. Lippman, and J. Wang, "Separation of multiple passive RFID signals using software defined radio," in *Proc. IEEE Int. Conf. RFID*, 2009, pp. 139–146.
- [28] A. Bletsas, J. Kimionis, A. G. Dimitriou, and G. N. Karystinos, "Single-antenna coherent detection of collided FM0 RFID signals," *IEEE Trans. Commun.*, vol. 60, no. 3, pp. 756–766, Mar. 2012.
- [29] R. S. Khasgiwale, R. U. Adyanthaya, and D. W. Engels, "Extracting information from tag collisions," in *Proc. IEEE Int. Conf. RFID*, 2009, pp. 131–138.
- [30] L. Kang, K. Wu, J. Zhang, H. Tan, and L. Ni, "DDC: A novel scheme to directly decode the collisions in UHF RFID systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 2, pp. 263–270, Feb. 2012.
- [31] L. Kong, L. He, Y. Gu, M.-Y. Wu, and T. He, "A parallel identification protocol for RFID systems," in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 154–162.
- [32] H.-C. Liu and J.-P. Ciou, "Performance analysis of multi-carrier RFID systems," in *Proc. IEEE Int. Symp. Perform. Eval. Comput. Telecommun. Syst.*, 2009, pp. 112–116.
- [33] J. Yu, K. Liu, and G. Yan, "A novel RFID anti-collision algorithm based on SDMA," in *Proc. IEEE 4th Int. Conf. Wireless Commun. Netw. Mobile Comput.*, 2008, pp. 1–4.
- [34] J. Wang, Y. Zhao, and D. Wang, "A novel fast anti-collision algorithm for RFID systems," in *Proc. IEEE Int. Conf. Wireless Commun. Netw. Mobile Comput.*, 2007, pp. 2044–2047.
- [35] J. Capetanakis, "Tree algorithms for packet broadcast channels," *IEEE Trans. Inf. Theory*, vol. IT-25, no. 5, pp. 505–515, Sep. 1979.
- [36] D. R. Hush and C. Wood, "Analysis of tree algorithms for RFID arbitration," in *Proc. IEEE Int. Symp. Inf. Theory*, 1998, Art. no. 107.
- [37] J. Myung and W. Lee, "Adaptive binary splitting: A RFID tag collision arbitration protocol for tag identification," *Mobile Netw. Appl.*, vol. 11, no. 5, pp. 711–722, 2006.



**Qiongzhen Lin** received the BS degree from the School of Software, Tsinghua University, China, in 2012. He is now working toward the PhD degree in School of Software, Tsinghua University, China. His research interests include radio frequency identification (RFID) and sensor network, mobile sensing and pervasive computing. He is a student member of IEEE and ACM.



**Lei Yang** received the BS and PhD degrees from the School of Software, Department of Computer Science and Engineering, Xi'an Jiaotong University. He is currently working as a research assistant professor with the Department of Computing, The Hong Kong Polytechnic University. Previously, he was a postdoc fellow with the School of Software of Tsinghua University. His research interests include RFID and backscatters, wireless and mobile computing. He is a member of the IEEE.



**Chunhui Duan** received the BS degree from the School of Software, Tsinghua University, and is now working toward the PhD degree in the School of Software, Tsinghua University. Her research interests include RFID, wireless network, mobile sensing and pervasive computing. She is a student member of the IEEE.



**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 Chang Jiang chair professor and dean of School of Software, Tsinghua University, China. He is an ACM distinguished speaker and now serves as the chair of ACM China Council and also an associate editor for the *IEEE/ACM Transactions on Networking* and the *ACM Transactions on Sensor Network*. His research interests include RFID and sensor network, the Internet and cloud computing, and distributed computing. He is a fellow of the IEEE and ACM.

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