Probing into the Physical Layer: Moving Tag Detection for Large-Scale RFID Systems

Chuyu Wang, Student Member, IEEE, Lei Xie, Member, IEEE, Wei Wang, Member, IEEE, Yingying Chen, Senior Member, IEEE, Tao Xue, and Sanglu Lu, Member, IEEE

Abstract—Logistics monitoring is a fundamental application that utilizes RFID systems to manage numerous tagged-objects. Due to the frequent rearrangement of tagged-objects, a fast RFID-based tracking approach is highly desired for accurate logistics distribution. However, traditional RFID systems usually take tens of seconds to interrogate hundreds of RFID tags, not to mention the time delay involved to locate all the tags, which severely prevents from in-time tracking. To address this issue, we reduce the problem domain by first distinguishing the motion status of the tagged-objects, i.e., “stationary” or “moving”, and then tracking the moving objects with the state-of-the-art localization schemes, which significantly reduces the efforts of tracking all the objects. Toward this end, we propose a moving tag detection mechanism, which achieves the time efficiency by exploiting the useless collision signal in RFID systems. In particular, we extract two kinds of physical-layer features (namely phase profile and backscatter link frequency) from the collision signal received by the USRP to distinguish tags at different positions. We further develop the Graph Matching (GM) method and Coherent Phase Variance (CPV) method to detect the moving tagged-objects. Experiment results show that our approach can accurately detect the moving objects while reducing 80% inventory time compared with the state-of-art solutions.

Index Terms—RFID, Collision Decoding, Tag Inventory

1 INTRODUCTION

With the rapid proliferation of IoT (Internet of Things) industry, RFID, as a key technology, has been deployed in increasingly large numbers to facilitate the smart management. For example, in the logistic monitoring, there are usually more than hundreds of objects attached with RFID tags in the monitoring area. Due to the frequent rearrangement of the tagged-objects, the RFID systems are required to track the movement of all tags timely to prevent the target objects from mistakenly rearranging. However, a Commercial-Of-The-Shelf (COTS) RFID system usually takes tens of seconds to interrogate hundreds of RFID tags [1], [2], not to mention the time delay to track all the tags. This severely hinders the system from tracking the movement of tagged-objects in time. Since only some of the objects are moved at a certain moment, to reduce the efforts of tracking all the objects, one possible solution is to first identify the motion status of the objects, i.e., “stationary” or “moving”, and then only track the “moving” objects. For the “stationary” objects, since they are presumed to be statically placed in a specific location, we do not need to track them. For the “moving” objects, we can leverage the existing localization techniques to track them. Since the “moving” objects only occupy a small part of the total number, we can save a lot of time by only focusing on tracking the moving objects, instead of wasting time in tracking the stationary objects, which makes it possible to perform the fast large-scale monitoring.

To track the moving tags in the monitoring area, existing studies [3]–[5], [11]–[13] usually involve two steps, i.e., a fast tag inventory scheme to interrogate tags, and an effective positioning scheme to determine the motion status of the tags, which would be hard to perform large-scale moving RFID monitoring in a timely manner. Specifically, for the tag inventory schemes in RFID systems, traditional polling-based schemes [3]–[5], [14] leverage the Frame-Slotted-Aloha (FSA) protocol to identify the tags, which usually takes tens of seconds to interrogate hundreds of RFID tags in real RFID systems. The main cause of such time inefficiency is the waste of the collision slots, which usually occupy a large proportion of the overall time slots. Recently, some emerging work try to make use of the collision slots to improve time efficiency of tag inventory [6]–[9] and further detect the missing tags from the collision signals [3], [10]. However, different from a missing tag, a moving tag can still be interrogated by the reader, thus these methods are not suitable to detect the moving tags. In addition, for the positioning schemes, the state-of-the-art localization schemes [11]–[13] usually locate the tags one by one, and they usually take up to several hundreds of milliseconds to locate a unique tag. Therefore, it is difficult to concurrently locate all tags timely using existing solutions, when dealing with hundreds of tags.

In this paper, we propose a fast moving tag detection scheme for large-scale RFID systems, which works as a fundamental premise to support the tracking applications of tagged-objects. The main idea is to extract the physical-layer features of each tag from the collision signal to achieve the time efficiency. Fig. 1 uses the collision signal of three
For each query cycle, we construct an k slots, we focus on extracting the physical-layer features of RFID tags, i.e., the phase profile and the backscatter link frequency, to distinguish the tags at different positions. The two physical-layer features then serve as the fingerprints of each tag to derive the motion status of all tags simultaneously, greatly improving the overall time-efficiency. Toward this end, we design a two-phase tag-detection scheme, including the tag inventory and continuous polling, to determine the motion status of tags. In the tag inventory phase, the RFID reader identifies each tag via a traditional inventory cycle and constructs the original distribution of the physical-layer features for all the tags. It may take tens of seconds due to the waste of collision slots. In the continuous polling phase, the RFID reader continuously monitors the motion status of each tag by issuing multiple query cycles. Differing from the existing solutions, which still identify the tags via the singleton slots, we focus on extracting the physical-layer features from the signal of collision slots. Thus we can save lots of inventory time by making use of the useless collision slots. For each query cycle, we construct an updated distribution of the physical-layer features from both the singleton and collision slots. Then we can detect the moving objects from the two distributions based on the fact that a static tag has stable physical-layer features, while a moving tag has different physical-layer features across the two distributions. Particularly, a Graph Matching (GM) method is proposed to detect the moving tags effectively based on the Hungarian algorithm, and a Coherent Phase Variance (CPV) method is proposed to determine the moving objects when we attach multiple tags on one object for robustness. Since the tag inventory phase and continuous polling phase are executed alternately, the time inefficiency of the tag inventory phase can thus be amortized by the subsequent multiple polling cycles, and the overall time-efficiency is achieved.

There are three main technical challenges in detecting the moving tags. The first challenge is to achieve time efficiency in large scale RFID systems. Due to the long duration of a traditional inventory cycle, it is difficult to continuously update the motion status of all tags within limited time intervals in a large-scale RFID system. To address this challenge, a two-phase monitoring scheme is proposed, which includes a normal tag inventory phase and multiple fast tag polling phases. During the fast tag polling phases, we significantly improve the time efficiency by exploiting the tag collisions to extract the physical-layer features for the detection of moving tags. The second challenge is to detect the motion status of all tags via the physical-layer features. Since the EPC ID is not even transmitted in the collision slot, it is difficult to determine the moving tags only based on the physical-layer features. To solve this problem, we find that the backscatter link frequency of the tag’s response has high degree of difference among different tags regardless of the motion status of the tags, whereas the phase profile from the tag’s response changes accordingly even if a specific tag is moved with a small distance. The coexistence of these two physical-layer features allows fine-grained tags and their locations discrimination to further determine their motion status. The third challenge is to extract the above physical-layer features from the collisions of multiple tag responses. To address this challenge, we investigate the geometrical characteristic of different kinds of collision signals in I-Q plane, and extract the phase profile of each tag response based on the specific geometrical characteristic. For each tag response, we further leverage the special patterns (e.g., preambles) to extract the backscatter link frequency from the signal length.

This paper presents the first study of probing into the physical-layer features to detect the moving tags for large scale RFID systems. Specifically, we make four main contributions (a preliminary version of this work appeared in [15]): First, we develop a mechanism to detect the motion status of RFID tags, which is a fundamental premise for tracking the movement of RFID tags in large-scale RFID systems. Second, our approach is able to extract the physical-layer features, including the phase profile and backscatter link frequency, from the collision signal for efficient moving tag detection. Third, we develop a voting-based scheme to determine the moving objects from multiple attached tags, which can tackle the measurement errors in real environments. Fourth, we evaluated our system in realistic settings and experiment results show that our solution can accurately detect the moving objects while reducing 80% inventory time compared with existing approaches.

2 Related Work

Missing tag detection. There have been active studies on detecting the missing tags based on the RF signals [3], [5], [16]. Yang et al. [5] propose to detect the missing tags based on the statistical signal information, which measures the RF signal of tags for a fairly long time for data collection. Zheng et al. [10] employ an efficient method to detect the missing tags based on signal superposition principle on physical layer. Zhu et al. [16] try to identify the unknown tags in large-scale RFID system. Different from the above work, which only consider the missing tags, we focus on detecting the moving tags, which still can be identified by the reader. Moreover, in regard to a large-scale RFID system, we need to achieve the time efficiency by updating the motion status in time, which is seldom discussed in the above works.

Collision recovery. Many studies focus on recovering the tag signal [7], [8] or estimating tag cardinality [17] from the collision signals based on the dedicated instruments like USRP. With the Software Defined Radio based UHF-RFID reader designed by Buettner et al. [18], several methods are proposed to deal with the collision problems [6]–[8], [19]. Zheng et al. [19] use the Computational RFID tags and SDR reader to improve the data throughput. Wang et al. [6] view collisions as a code across the bits transmitted by the tags to improve the bandwidth in RFID system. Hou et al. [17] investigate the collision signals in physical layer to estimate cardinality of large scale RFID system. Other work [7], [8] try to recover the data from the collision signals by leveraging the time-domain separations. Unlike these work, we try to extract the tag position related physical features from the collision signals to detect the moving tags.
Physical layer identification. Due to the hardware imperfection, leveraging the physical layer features to perform tag identification or authentication has drawn widespread attention recent years [20]–[23]. Danev et al. [21] study the feasibility of physical-layer fingerprint in tag identification in practical settings. Zanetti et al. [20] exploit the difference of backscattered link frequency caused by manufacturing imperfection of tags to distinguish different tags. Furthermore, Ma et al. [22] distinguish the tags by leveraging the internal similarity among pulses of tags’ RN16 preamble signals. Yang et al. [23] leverage the phase deviation of each tag and the specific geometric relationship among these tags to authenticate the legal products. However, apart from the identification, we also need to detect the motion status of each tag, which is a fundamental premise of tracking the movement of all the tags.

3 System Design

3.1 System Goals

In this paper, we propose a fast moving tag detection scheme for large scale RFID systems, so as to further support tracking the movement of all tagged-objects. Since the tagged-objects in real RFID may be frequently moved in and out for logistic distribution, we need to continuously update the motion status within a limited time interval. Therefore, our moving tag detection scheme should be able to improve both the time efficiency in tag inventory and the accuracy in detecting the motion status of all tags: 1) The average duration for each cycle of tag inventory should be sufficiently reduced to achieve the time requirement for large scale RFID systems. 2) There are two kinds of errors in the problem: a) False positive errors: the stationary tags are detected as moving tags. b) False negative errors: the moving tags are detected as stationary tags. Both of the two errors should be effectively reduced in detecting the motion status of all tags.

3.2 System Architecture

In this paper, two kinds of physical-layer features are investigated to effectively detect the motion status of all tags. 1) Phase profile: it is the phase values of the RF signal, which describes the degree that the received signal offsets from sent signal, ranging from 0 to 360 degrees. The phase value from the tag’s response changes even if the tag is moved with a small distance. 2) Backscatter link frequency (BLF): it is the frequency of the tag-to-reader link, which indicates the data rate in the tag’s response signal. Due to manufacturing imperfection, BLF varies among different tags. Therefore, it is suitable to combine the two features to detect the motion status of tags. Moreover, in order to satisfy the time efficiency, we recover each tag response according to the geometrical characteristic of the collision signals in I-Q plane, and extract the aforementioned physical-layer features from collision signals.

Fig. 2 presents the whole architecture of our system. We propose a two-phase monitoring scheme, including the tag inventory and continuous polling phase, to efficiently detect the motion status of all tags. In the tag inventory phase, the reader identifies each tag via a traditional inventory cycle to extract the physical-layer features of all tags. Since the tags are all static in this phase, the physical-layer features of all tags construct an original distribution of the physical-layer features. In the continuous polling phases, the reader continuously monitors the motion status of all tag by issuing multiple query cycles. For each query cycle, the reader constructs an updated distribution of the physical-layer features by effectively extracting the two features from both the singleton and collision signals. By comparing the updated distribution with the original distribution, we utilize a Graph Matching (GM) method to detect the moving tags in every query cycle. Moreover, based on the detected moving tags from the GM method, we further propose a Coherent Phase Variance (CPV) method to detect the moving objects, missing objects and inserting tags based on the multiple tags attached on one object. The multiple query cycles in the continuous polling phase save lots of time from the collision signals, and thus can amortize the time spent in the inventory phase, which takes more time due to the traditional inventory cycle. Therefore, by efficiently extracting the position related features from the collision signal, we can largely reduce the overall time in detecting the motion status compared with the existing C1G2 standard-based methods.

4 Physical-Layer Features Calculation

In this section, we demonstrate how to calculate our physical-layer features from the raw signal of tag response via realistic experiments. We implement a software defined reader (SDR reader) according to the Gen2 project [18]. Specifically, we operate the Gen2 project on our USRP platform [24] with two FLEX-900 daughter boards and two Ladir S9028 antennas on each board for transmitting and receiving, respectively. For the receiving module, we set the sampling rate to 2MHz, which represents 0.5μs per sample.

4.1 Tag Response in a Singleton Slot

According to the EPC C1G2 standard [4], the RFID reader interrogates the tags based on the Frame-Slotted-Aloha (FSA) protocol. In the FSA protocol, each inventory cycle is divided into multiple slots to identify the tags. For each frame, the unidentified tags need to randomly select one slot for its data transmission. The reader starts a slot by sending
a QUERY/QRep command. Any tag that selects this slot will respond the reader with a 16-bit random number RN16 for channel probing. If the reader successfully decodes the RN16 bits, it responds the tag with an ACK, that tells the tag to transmit its EPC-ID. Otherwise, the reader will start the next slot, because there are multiple tags or none tag transmitting in this slot. Therefore, there are two kinds of tag responses generally: 1) RN16, where the tag responds the QUERY or QRep command, 2) EPC-ID, where the tag answers the ACK command. During the tag response, the reader keeps transmitting Continuous Wave (CW) to supply power for the tags. Fig. 3 presents a typical singleton slot in RFID systems, which is collected from USRP. We note that the time of the EPC-ID is about 4 times longer than that of the RN16, because the data length of EPC-ID is much longer than RN16. Meanwhile, since the time interval between the two responses, i.e., the length of ACK, is so small, the position and wireless environment of both RN16 and EPC-ID can be regarded unchanged. Hence, instead of extracting the physical-layer features from the EPC-ID, we can directly extract the feature from the RN16 signal, and then skip the EPC-ID signal to save the inventory time.

4.2 Phase Profile

Next, we demonstrate how to calculate the phase profile from the RN16 signals via the signal transmitting model. In RFID systems, the tag transmits data using backscattering modulation. Hence, the baseband signal received by the reader can be represented as:

\[
s(t) = A e^{j\beta} + x(t) B e^{j(2\pi f t + \Phi)} + \hat{h}(t). \tag{1}
\]

Here, \(A e^{j\beta}\) is the carrier signal (i.e., Continuous Wave, CW), where \(A\) indicates the amplitude, \(j\) is \(\sqrt{-1}\) and \(\beta\) is the corresponding phase value. \(B e^{j(2\pi f t + \Phi)}\) is the backscattered signal of the tag, where \(B\) indicates the amplitude of the backscattered signal and \(f\) indicates the corresponding frequency of the backscattered signal. \(x(t)\) is the binary bits sent by the tag, which is equal to either ‘0’ or ‘1’. \(\hat{h}(t)\) is the ambient noise. Therefore, the actual baseband signal received by the reader is a superposition of the carrier signal and the backscattered signal.

Intuitively, the baseband signal received from a single tag response can be expressed in I-Q plane as shown in Fig. 4. The received signal consists of two parts: 1) leakage signal: the constant carrier signal (i.e., CW), 2) backscattered signal: the modulated tag signal. Formally, we define the channel coefficient of the tag’s backscattered signal as \(h = B e^{j(2\pi f t + \Phi)}\), which expresses the channel information of the backscattered signal, i.e., the phase value and signal strength. As for the phase value of the received signal in Fig. 4, it can be represented as:

\[
\theta = \Phi - \beta, \tag{2}
\]

which is the angle difference between the carrier signal phase \(\beta\) and the backscattered signal phase \(\Phi\). We call \(\theta\) the phase profile of the tag in this work.

We further carry out trace-driven evaluations to study the property of the phase profile. Firstly, we evaluate the stability by conducting an empirical experiment on 50 tags with random deployments. For each setting, we deploy each RFID tag 1m in front of the SDR reader [18] based on USRP platform, and measure 100 phase profiles by querying each tag 100 times with the USRP reader. The results are normalized by subtracting the average phase value of each result set. Therefore, 5000 normalized phase profiles are measured to evaluate the stability. As shown in Fig. 5(a), the phase profile varies from \(-5^\circ\) to \(5^\circ\), following a typical Gaussian distribution. So we can treat the phase profile as a stable feature for motion detection.

Secondly, we compare the phase profile of SDR reader with the phase value of commercial reader (Impinj R420) by issuing the same tag. We vary the distance between the antenna and the tag, which ranges from 20cm to 70cm stepping by 1cm. For each step we measure 100 phase values individually. As shown in Fig. 5(b), the phase profile of SDR reader is almost the same as the phase value of commercial reader, where the correlation coefficients calculated on MATLAB is 0.9979. Therefore, the phase profile is sensitive to any tiny movements, e.g., 1cm movement, which guarantees the distinctiveness of the phase profile in the motion detection.

4.3 Backscattered Link Frequency

In regard to the backscatter link frequency (BLF) of the response signal, it can be represented as \(f_t\) in Eq. (1). Due to the manufacturing imperfection, the BLF varies among different tags, which is used to distinguish tags [20], [22]. In fact, \(f_t\) determines the data rate of the tag’s response. In regard to a typical encoding scheme Miller-4 of RFID system in Fig. 6(a), \(f_t\) determines the duration of a binary symbol, i.e., bit-0 or bit-1. Therefore, both bit-0 and bit-1 have the same signal duration according to the Miller-4 encoding scheme. Since the binary length of RN16 signal is fixed, including the same preambles, a 16 bits random number and 1 check bit, it is reasonable to use the corresponding signal length of RN16 to represent the BLF. Meanwhile, we can also convert the signal length of RN16 to the BLF based on the actual symbols in RN16. In regard to other encoding schemes in RFID system, e.g., FM0, we can similarly use the signal length of RN16 to represent the BLF.
We present a cross-correlation based technology to extract the signal length by locating the starting and ending point of RN16 signal. Specifically, we adopt a slide window to calculate the cross-correlation value between the measured samples in the window and the special data sequence, i.e., the preamble or the “dummy 1”. Then we find the window whose cross-correlation value is the maximum, and record the position of the window. In Fig. 6(b), we move the slide window forward to locate the starting point of RN16 based on the preamble sequence. Similarly, we can locate the ending point using the “dummy 1”, while moving the slide window from the end of RN16 signal backward. We use the number of samples between the starting and the ending point to represent the value of the BLF.

To validate the distinctiveness of BLF, we conduct experiments on 50 different tags at 9 random positions in front of the antennas. We repeat querying each tag 100 times with our USRP reader, and then extract the signal lengths in different positions. As shown in Fig. 7(a), the average signal length of 50 tags is randomly distributed from 7620 to 7690 samples. Since the sampling rate of the USRP reader is 2MHz, the signal length is around 3.81ms. It indicates that even though the same model tags are queried with the same settings and the actual data rates are different due to the manufacturing imperfection. Therefore, the distinctive value of BLF can be regarded as a unique physical-layer feature to distinguish among tags. Furthermore, we draw the histogram of the normalized variance of the signal length in Fig. 7(b). The normalized signal length is relatively stable with an average deviation of 2 samples, which is equal to 1μs according to the 2MHz sampling rate. Therefore, the BLF is relatively stable and distinctive even though the tags are at different positions. But some tags have similar BLF values, meaning BLF cannot be used to uniquely differentiate the tags.

5 Physical-Layer Features Extraction from Collision Signal

In the previous section, we have shown how to calculate the physical-layer features in the singleton slots. Such calculations can be used to collect the physical-layer features in the tag inventory phase, since we leverage the traditional C1G2 protocol [4] to identify the tags from the singleton slots. However, it is still time inefficient because we cannot avoid the useless collisions in C1G2 protocol. If c tags select the same slot to transmit the data, a c-collision happens and such slots are wasted in C1G2 protocol. In fact, when we interrogate N tags with an f-slot frame based on the C1G2 protocol, the N tags select slots randomly according to the Binomial distribution and the probability of a c-collision slot can be expressed as:

$$Pr(c) = \binom{N}{c} \left( \frac{1}{f} \right)^{c} \left( 1 - \frac{1}{f} \right)^{N-c} \quad (3)$$

Fig. 8 shows the theoretical probability distribution of different slot types. In this figure, in regard to the slot types, ‘0’ indicates the empty slot, ‘1’ indicates the singleton slot, and ‘2’, ‘3’, ‘4+’ indicates the collision slot with different number of collided tags in one slot. We enumerate the values of f/N, which are the ratio between the frame size f and the total tag number N (N is set to 1000 as default), and calculate the distribution of each slot type according to Eq. (3). In regard to the C1G2 protocol, since it only receives signal from the singleton slot, it need to maximize Pr(1) to improve the time efficiency. As a result, at most 36.8% slots are singleton when f equals to N. Then, about 63.2% slots are wasted, which severely affects the inventory time efficiency. In this case, 2-collision and 3-collision slots occupy 18.4% and 6.13% slots respectively, while only 1.89% slots are remained in 4+-collisions. So if we can efficiently resolve all the tags in singleton, 2-collision and 3-collision slots, we can identify 36.8% + 2 × 18.4% + 3 × 6.13% = 92% tags in a frame, which is 2.5 times compared with current protocol, i.e., 36.8%. Hence, we focus on how to extract physical-layer features from 2-collision and 3-collision slots to improve the time efficiency.

5.1 Model of Collision Signal

Extending Eq. (1) from a single tag to multiple tags, the received baseband signal of a c-collision slot can be expressed as:

$$s(t) = Ae^{jB} + \sum_{i=1}^{c} x_{i}(t)h_{i} + \hat{h}(t) \quad (4)$$

Here, x_{i}(t) is the binary bits sent by tag i over time t. AejB represents the leakage signal from the reader. h_{i} is the channel coefficient of tag i and can be written as:

$$h_{i} = B_{i}e^{j\pi f(l+i+6)s_{i}B)} \quad (5)$$

Fig. 8. Theoretical probability distribution. The probability distribution of different kinds of collision slots in regard to the ratio between the frame size f and the number of tags N. N is set to 1000 as default.
where $B_i$ is the amplitude and $\theta_i$ is the phase profile of tag $i$. Similar to the model in Fig. 4, $h_i$ describes the backscattered signal in I-Q plane as a vector, thus the collided signal of the $c$ tags can be represented as the superposition of the backscattered signal and the leakage signal.

Taking a 2-collision as an example in Fig. 9(a), since both tag $A$ and $B$ send a binary bit, the superposition signal based on the combination of the two binary bits forms four different states $S_0 \sim S_3$ in I-Q plane. State $S_0$ means both tag $A$ and $B$ are transmitting $x_i(t) = 0$, therefore the signal contains only one component: the leakage signal. State $S_3$ means both tag $A$ and $B$ are transmitting $x_i(t) = 1$, therefore the signal is superposition of the leakage signal and the backscattered signals. For state $S_1$ and $S_2$, only one tag is transmitting bit 1, therefore the signal is the combination of the leakage signal and the backscattered signal of the corresponding tag. Here, the key step to decompose the backscattered signals of different tags is to efficiently resolve these states and extract the channel coefficients $h_i$.

5.2 Channel Coefficients Estimation from Collision Signal

Based on the above model, the superposition of the backscattered signals constitute different states in I-Q plane, which represents different binary bits sent by the collided tags. Hence, we next demonstrate how to resolve the signal states so as to extract features from collision signal. Firstly, we need to find out which combination of binary bits each state represents, e.g., in Fig. 9(a) $S_0$ represents two bit 0. Secondly, we calculate the channel coefficient $h_i$ of each tag using the positions of states. Lastly, for each signal sample $s(t)$ we estimate the binary bits $x_i(t)$ of tag $i$ according to Eq. (4) as:

$$\arg \min_{x_i(t)} |s(t) - \sum_{i=1}^{c} (x_i(t) \cdot h_i) - Ae^{jB}|, \text{ where } x_i(t) \in \{0, 1\}. \quad (6)$$

Eq. (6) chooses the combination of $x_i(t)$, such that the error between the sample $s(t)$ and the generated collided signal, i.e., $\sum_{i=1}^{c} (x_i(t) \cdot h_i)$ is minimized. Then, we can calculate the phase profiles from the channel coefficient $h_i$ of each tag and calculate the BLFs from the binary sequence $x_i$ of each tag based on the cross-correlation method.

5.2.1 Channel Coefficients Estimation for 2-collision Signal

According to the above analysis, we next resolve the signal states in 2-collision problem. There are four states in a 2-collision slot, which is the size of two-bit binary. When we get the collision signals, we first acquire the positions of each state by clustering all the samples into clusters based on the sample distribution in I-Q plane. Then we pick the peaks of density function as the centers of clusters based on [17].

After clustering, we will get four cluster centers, which represent four states similar to Fig. 9(a). As $S_0$ only contains the leakage signal, which can be estimated from the continuous wave, we can first determine the state $S_0$. Since both tags are transmitting the same data during the preamble, signals are switching between state $S_0$ and $S_3$ during the preamble as shown in Fig. 9(b). Therefore, we can determine the state $S_3$ using the preamble part, which contains only $S_0$ and $S_3$. Then, the remained two states are $S_1$ and $S_2$ respectively. At last, we calculate the channel coefficients of each tag as $h_1 = C(S_1) - C(S_0)$ and $h_2 = C(S_2) - C(S_3)$, where $C(S_i)$ is $S_i$ in the I-Q plane.

5.2.2 Channel Coefficients Estimation for 3-collision Signal

When the number of collided tags increases to three, the signal states can be represented with a three-bit binary. This means there are 8 states, which is much more complex, because we need to define 4 more states compared to 2-collision problem. Based on the combination of vectors from 3 backscattered signals in I-Q plane, the 8 states constitutes a parallelepiped. Our basic idea is that since the 3-collision signal can be always represented as the superposition of 2-collision signal and one tag signal, we can resolve a 2-collision problem inside the 3-collision problem and then handle the 3rd tag signal. Taking Fig. 9(c) as an example, the parallelogram $S_0S_2S_3S_4$ and $S_1S_2S_7S_8$ are the superposition of two tags and the vector pointing from one parallelogram to another represents the signal of the 3rd tag. By resolving one parallelogram, we can get the channel coefficients of two tags and further calculate the coefficients of the 3rd tag.

To find the parallelogram in the parallelepiped, we divide the states into two parts according to their ranks of amplitude. Due to the geometric feature, the four states with smaller amplitudes can either constitute a parallelogram as shown in Fig. 9(c), or a plane tetrahedron as shown in Fig. 9(d). For the parallelogram case, we have found the parallelogram such as Fig. 9(c). For the plane tetrahedron case, the edges among the four states inside the parallelepiped intersect at one state, e.g., in Fig. 9(d) three vectors intersect at $S_2$. For this situation, we can exchange one of the other three states with a symmetrical state, e.g., replacing $S_1$ with $S_5$ in Fig. 9(d), to constitute a parallelogram. Then, we will find the parallelogram so as to further extract the channel coefficients.

The detailed steps for estimating the channel coefficients in 3-collision slot are as follows: Firstly, we cluster signals into 8 clusters, similar as in the 2-collision problem. Sec-
ondly, we determine state $S_0$ and $S_7$ from the preamble, because all the three tags transmit the same data and signal switches between state $S_0$ and $S_7$. Here, $S_0$ means all the three tags transmit $x_i(t) = 0$ and $S_7$ means $x_i(t) = 1$. Thirdly, we search for the parallelogram in the parallelepiped. We sort the eight states based on the amplitudes of cluster centers. Based on the analysis above, then we search for the parallelogram among first five states with smaller amplitudes. There are only two possible choices: 1) For the parallelogram case we can use 1st-4th state to constitute a parallelogram. 2) For the plane tetrahedron case, we can replace 4th state with 5th and use 1st-3rd and 5th state to constitute a parallelogram.

To decide the correct choice of parallelogram, we leverage the property that the opposite sides of parallelogram are parallel and equal in length. For each choice there are three possible edge pairs. If we use the rank of state to represent the vertex, the three pairs are $(12, 34)$, $(13, 24)$ and $(14, 23)$ in choice 1. We use the similarity among the edge pairs to express the likelihood of constituting a parallelogram as:

$$
Sim = \frac{\tilde{a} \cdot \tilde{b}}{|\tilde{a}| \times |\tilde{b}|} \times \frac{1}{(|\tilde{a}| - |\tilde{b}|)},
$$

(7)

where $(\tilde{a}, \tilde{b})$ is the possible edge pair. The left part in the multiplier is the cosine value of the edge pair and the right part is the reciprocal of the length difference. We choose the edge pair with the maximum similarity as the corresponding opposite side of the parallelogram. Then we can resolve the vertex sequence of the parallelogram based on the similarity.

Lastly, we measure the channel coefficients of each tag based on the geometric construction in the I-Q plane. Based on our division, the selected parallelogram will always include either state $S_0$ or $S_7$. Hence, we can measure the two channel coefficients in the parallelogram according to the states $S_0$ and $S_7$ similar as in the 2-collision problem. Since $S_7$ contains three channel coefficients and $S_0$ contains none, the channel coefficient of the 3rd tag can be easily computed based on $S_0$, $S_7$ and the first two channel coefficients.

### 5.3 Features Estimation from Collision Signal

After extracting the channel coefficients from the collision slot, we can directly compute the phase profile based on Eq. (2). Suppose the channel coefficient for tag $i$ is estimated as $h_i$. Then it represents that $h_i$ includes two states according to the estimation method for channel coefficient: one for bit ‘0’ and one for bit ‘1’. According to Eq. (2), we can then calculate the phase profile of the tag based on the channel coefficient, which is the phase difference between the two states bit ‘0’ and bit ‘1’ in the I-Q phase.

In regard to the BLF, we utilize Eq. (6) to roughly recover the RN16 signal of each tag and calculate the signal length of RN16 for each tag. Particularly, we have already estimated the channel coefficient of each tag as $h_i$ and the summation of the signal of all the tags has been measured as $s(t)$. Then we can calculate the binary samples of each tag as $x_i(t)$ from $h_i$ and $s(t)$ based on Eq. (6). Here, the value of binary bit trace $x_i(t)$ is set to either ‘0’ or ‘1’. Due to the large noise of the collision signal, it is usually difficult to accurately decode the binary bit trace. It means that even though the whole trace of $x_i(t)$ is almost correct, some samples may be incorrect, which is later shown in Fig 10(c). However, according to the special encoded pattern of preamble and “dummy 1” as shown in Fig. 6(a), we can still determine the starting and ending point.

Specifically, we first use the moving average method to smooth the recovered binary trace $x_i(t)$ and reduce the influence of the incorrect samples. Then we adopt the cross-correlation process similar to Section 4.3 to decide the signal length. The cross-correlation method calculates the similarity between the special encoded pattern and the recovered trace $x_i(t)$. The similarity will reach the peak when the trace $x_i(t)$ is the same as the special encoded pattern, which is regarded as the starting and ending point of RN16 for each tag $i$. Finally, we use the length between the starting and ending point as the indicator of BLF.

### 5.4 Case Study

Fig. 10 presents an example of extracting the physical-layer features from a 3-collision slot. Firstly, we cluster the samples into eight clusters based on the density function in the I-Q plane as shown in Fig. 10(a). In the following we denote each state according to the amplitude rank of states in the I-Q plane. Secondly, we determine state $S_0$ from continuous wave and state $S_7$ from the preamble as shown in Fig. 10(b). Thirdly, we search for the parallelogram based on the first 5 states. The left quadrilateral in Fig. 10(b) is the estimated parallelogram and the right one is the symmetrical one. Fourthly, we calculate the channel coefficients including the phase profiles, which is shown as the arrow in the figure. Lastly, we recover the RN16 bit trace of each tag according to Eq.(6). We compute the signal length of RN16 as the indicator of BLF based on cross-correlation results. Fig. 10(c) shows the ending part of three separated RN16 signals and points out the slide window which has the peak cross-correlation value. In this figure, we can clearly see the
recovered RN16 signal of each tag. The covered trace is not the perfect square wave, which has some noise. It is caused by the incorrect calculated binary bits. We use the moving average method to smooth the whole trace and reduce the effect of these incorrect bits.

6 MOVING OBJECT DETECTION VIA A SINGLE TAG

6.1 Motivation and Intuition

As shown in Section 4.3, BLF is only able to differentiate dozens of tags, but cannot distinguish tags with the same BLF. However, in our scenario, the most tags are stationary, meaning we can also use the unique positions, i.e., phase profiles, to represent each static tag. Therefore, we can combine BLF and phase profiles to detect the moving tags, which integrates both the inherent tag features and the position-related tag features. In this section, we focus on how to detect the moving tags by using the phase profiles, after extracting the physical-layer features. BLF can be combined with the phase profile similarly.

The phase profile always changes if the tag moves, even if the moving distance is small, thus the basic idea is to compare the updated phase profiles with the stationary phase profiles. Suppose the stationary phase profiles of the $N$ tags obtained in the tag inventory phase is $P = (\theta_{1}, \theta_{2}, \ldots, \theta_{N})$, where $\theta_{i}$ is the $i$th phase profile. In the each continuous polling cycle, we also obtain the updated phase distribution $P' = (\theta'_{1}, \theta'_{2}, \ldots, \theta'_{N})$. We compare $P'$ with $P$ to detect the moving tags in each polling cycle. For the convenience of demonstration, we use the terms “phase profile”, “tag” interchangeably to represent the corresponding phase profile.

Traditional CIG2 protocol usually costs tens of seconds from singleton slots to obtain the phase distribution, which is quite inefficient due to the collision problem and the long EPC-ID signal. Instead, we can extract the phase distribution $P'$ only from the RN16 signal of collision signal. Hence, we propose a Fast Tag Polling (FTP) scheme by suppressing the transmitting of EPC-ID signal with a new command QrepSup. The QrepSup command is used to respond the RN16 signal of the tags. It not only starts the next slot like the Qrep command in CIG2, but also makes the tags, which is resolved in the collision slot, silent for the following frames in the query cycle. Based on the FTP scheme, we can acquire the phase distribution $P'$ in a short time in every polling cycle.

6.2 Multi-dimensional Phase Distribution Construction

Even though the movements of tags lead to the difference between the two phase distributions $P$ and $P'$, the phase profile extracted from one antenna cannot sufficiently distinguish the moving tags due to the following two reasons: 1) The phase value is periodical, meaning different tags may have the same phase profile. 2) Measurement errors of phase profile will affect the accuracy of moving detection.

Fig. 11 presents a simple example. $P_{1}$ and $P_{2}$ are two phase distributions of five tags $T_{1} \sim T_{5}$ measured by two antennas. We find that $T_{2}$ and $T_{3}$ have similar phase profile in $P_{1}$, and $T_{3}$ and $T_{4}$ also have similar phase profile in $P_{2}$. Therefore, we cannot uniquely distinguish each tag based on the phase profile from one antenna. To solve the problem, we can combine the two phase profiles together as a 2-dimensional phase profile, which is shown as the matrix in the figure. We can clearly see that $T_{3}$ can be uniquely detected, because no tag has similar 2-dimensional phase profile with $T_{3}$.

In actual applications, we can exploit with a transmitting antenna and multiple receiving antennas to acquire the multiple distributions [25]. But combining multiple distributions to construct a multi-dimensional phase profile is still challenging, because we do not obtain the tag IDs. Taking Fig. 11 as an example, if tag $T_{1}$ collides with $T_{2}$ in a 2-collision, we can extract the phase profiles $7^\circ$ and $194^\circ$ from one antenna and $192^\circ$ and $18^\circ$ from another antenna. But we cannot determine whether the phase pair $(7^\circ, 192^\circ)$ belongs to the same tag or not, because we do not have the tag IDs. Fortunately, the two antennas receive the same collision signal in the physical layer, meaning the BLFs of the same tag from the two antennas should match perfectly. Based on the understanding, we can combine $7^\circ$ and $192^\circ$ together, if the corresponding BLF values are similar. We can easily extend the matching method to multi-dimensional phase profile, where we use multiple receiving antennas.

Therefore, the process of constructing the multi-dimensional phase profile consists of two steps: We first try to extract the physical-layer features from the receiving signal of each antenna individually. For singleton slots, we just directly combine the phase values together to construct a multi-dimensional phase profile. For a c-collision slot, we combine the phase profiles received from different antennas by matching the BLFs to construct c multi-dimensional phase profiles.

6.3 Moving Tag Detection via Graph Matching

Based on the two phase distributions $P$ and $P'$, we demonstrate how to detect the moving tags by comparing the two phase distributions. The intuition is that for a stationary tag, its phase profile should be stable between the two distributions, while for a moving tag, its phase profile should change across the two distributions. Therefore, if we can match the stationary tags together, the rest tags are moving.

So the moving tag detection problem is transformed into a matching problem of stationary tags. In particular, we calculate the Euclidean distance $d_{i,j}$ between the $i$th phase profile $\theta_{i}$ in distribution $P$ and the $j$th phase profile $\theta'_{j}$ in $P'$ as:

$$d_{i,j} = ||\theta_{i} - \theta'_{j}||.$$  (8)

Thus, if $\theta_{i}$ and $\theta'_{j}$ belong to the same stationary tag, the distance $d_{i,j}$ should be small. Otherwise, the distance $d_{i,j}$ should be large, because $d_{i,j}$ represents either the phase change of the moving tag or the phase difference of two different tags, i.e., a random distance. Supposing $\kappa$ is an empirical threshold about the phase variance of a stationary tag, we can use it to filter the obvious abnormal distance by setting $d_{i,j}$, whose value exceeds $\kappa$, to $\infty$. So if we can match the phase profiles in the two distributions, so that the
overall sum of distances for these matchings is the minimum among all the possible matchings, then only the stationary tags should be matched, because the distances of the moving tags are large in statistic.

Matching between the two phase distributions can be reduced to an arrangement problems. For the first tag in $P$, we have $N$ phase profiles in the undated distribution to arrange, and for the second tag, we have $N - 1$ phase profiles to arrange. It means for all the $N$ tags in $P$, there are $N \times (N - 1) \times \cdots \times 1$ possible matchings. It is not suitable to enumerate all the possible matchings since the complexity is $O(N^3)$. So we propose a Graph Matching (GM) method by employing the Hungarian algorithm [26] to solve the problem in polynomial time. Firstly, we calculate the pairwise distance between distributions $P$ and $P'$ according to Eq. (8). It is a $N \times N$ matrix $D_{N \times N}$. Secondly, we filter all the unlikely pairs by setting the distance in $D_{N \times N}$, whose value exceeds $k$, to $\infty$. Thirdly, we utilize $D_{N \times N}$ as the input of the Hungarian algorithm and get a matching result. Lastly, we pick up all the tags in $P$, whose phase profiles are unmatched, as the moving tags.

Fig. 12 shows an example of the matching process. We use two gray matrices to represent the two distributions $P$ and $P'$ of five tags from three antennas, i.e., $A_1$, $A_2$ and $A_3$. We use the color of gray grid to represent the value of phase profiles, and three gray grids in one line represent three phase profiles of one tag received by antenna $A_1$, $A_2$ and $A_3$. The objective is to match the static tags in the two distributions together, where the distance between the three matching phase profiles in $P$ and $P'$ is the smallest. We use dashed lines to represent all the possible matching schemes and calculate the distances of them. The Hungarian algorithm takes the distances as input and returns the matchings, represented as the solid lines in the figure. So we can detect the moving tag as unmatched one, which is marked with a circle.

Currently, the phase profile is the main factor to detect the motion status, where the BLF is used to construct the multi-dimensional phase profile. In fact, since BLF is an inherent feature, it can be further used to distinguish the tags with similar phase profiles. We can add BLF into the distance calculation of the GM method, which may act as one more dimension of the phase profiles. Then, we can distinguish the tags, whose phase distance is small but BLF distance is large.

## 7 Moving Objects Detection via Multiple Tags

### 7.1 Motivation and Intuition

In practical environment, RFID systems usually suffer from various degrees of misreading phenomenon. As a result, even though some tags are stationary, they are likely to be detected as moving/missing tags based on the GM method due to the misreading. When the wireless environment becomes better, these tags can usually be successfully identified later. To solve the false detection, one intuitive solution is to expand the detection cycle, i.e., any tag is regarded as a moving tag if it is detected as moving for several continuous cycles. However, this solution delays the detection time and affects the system performance. In this paper, we propose to attach multiple tags on the object for the detection of moving/missing events. The basic idea is to determine the moving status from the majority tags attached on the object, so that a portion of reading errors will not affect the detection result. Additionally, based on the redundant tag deployment, it is possible to distinguish the moving objects from the missing objects, and even identify the newly joined tags, which are moved in from outside.

### 7.2 Phase Variance of Multiple Tags

We first study the phase variance of the multiple tags attached on a moving object via a real experiment. Even though these tags may also rotate along with the objects, the moving effect can be usually much larger than the rotation effect, when the tags are deployed vertically in the warehouse [27]. Therefore, we neglect the rotation effect and focus on the moving effect. When an object is moving, all the attached tags are under the same movement with the object, leading to the similar phase variance across these tags. To validate the hypothesis, we conduct an empirical study in the realistic environment by moving a carton box attached with four tags. Particularly, we continuously move the carton box $40cm$ in front of the antenna, and measure the phase values of these tags simultaneously. Fig. 13(a) presents the phase variances of the four tags during the movement of the box. We find that the measured phase values of the four tags are similar as expected. We call such phase variance the coherent phase variance of the multiple tags on the same object. We further present the correlation matrix among the four phase sequences in Fig. 13(b). We note that all of the correlation values are larger than $99\%$, which indicates they have the same variance pattern. Therefore, it is reasonable to use the coherent variance pattern to facilitate the detection of moving objects.

### 7.3 Moving Objects Detection from Unmatched Tags

Next we demonstrate how to use the coherent phase variance pattern to determine the moving objects. Consider we have detected the moving tags by matching between the stationary phase distribution $P$ and updated phase distribution $P'$ based on the GM method in Section 6. Denote the phase...
Fig. 14. Results of GM method. We can only detect the moving tags from the GM method, but when we attach more tags on one object, it is possible to determine the moving/missing objects from multiple tags.

profiles of unmatched tags in $P$ as set $D$, which are collected in the tag inventory phase. Meanwhile, we denote the phase profiles of unmatched tags in $P'$ as another set $D'$, which can be either the moving tags unmatched due to the phase change or the newly joined tags. Therefore, our mission is to efficiently find the matching between $D$ to $D'$ based on the coherent phase variance, such that the moving tags are matched together while the missing tags and inserting tags are unmatched.

The basic idea is that the multiple tags on the same moving object have the same moving trace, so that the phase variance of these tags should also be similar. We exploit Fig. 14 as an example to demonstrate the matching method. We attach $w$ ($w = 3$) tags on each object for the detection of moving objects. After the GM method, 6 tags of two objects in $P$ are unmatched, whose phase profiles are $D = (t_1, \ldots, t_6)$. And the unmatched phase profiles of 5 tags in $P'$ are denoted as $D' = (t'_1, \ldots, t'_5)$. Suppose object $O_1$ is moving and the object $O_2$ is missing. Then, only 3 tag pairs should be matched between $D$ and $D'$ as shown in the figure, such that the phase variance of the 3 tag pairs are similar. If we use a brute force search to find the matching, the possible matchings will increase exponentially along with the scale of $D$ and $D'$. In this example, even though the size of $D$ and $D'$ are only 6 and 5, respectively, for each tag in $D'$, there are 7 possible matching candidates, i.e., $t_1$ to $t_6$ and null. Therefore, there are $S^2 = 78,125$ possible matching combinations in total.

To tackle the problem, we propose Coherent Phase Variance (CPV) algorithm, a greedy method to search for the matching of the moving tags for each object. Instead of searching for a global matching between $D$ and $D'$, our idea is to separate the tags in $D$ into objects, and then search for the matching phase profiles in $D'$ for each object. Since phase profiles $D$ are achieved from the stationary phase distribution $P$ in the tag inventory phase, we know which tags in $D$ belong to one object. Therefore, we know $(t_1, t_2, t_3)$ are from the same object $O_1$. Hence, we first focus on searching for the matching tags of object $O_1$ and then handle other objects one by one.

We propose a five-step method to search for the matching tags of one object. Firstly, we calculate the phase difference node $m_{i,j}$ between each tag $t_i$ of object $O_k$ and each tag $t'_j$ in $D'$ as:

$$m_{i,j} = ||t'_j - t_i||,$$  
where $t_i \in O_k, t'_j \in D'$.  
(9)

If $t_i$ and $t'_j$ are the phase profiles of the same tag, then $m_{i,j}$ calculates the phase variance due to the movement of the object. Otherwise, it is only a random phase variance. Secondly, we build a graph from all the nodes $m_{i,j}$ and connect the similar nodes based on the coherent phase variance. Specifically, we link nodes $m_{i,j}$ and $m_{i+1,k}$ of two adjacent rows, if the phase variances are similar, e.g., less than the standard deviation of phase measurements. Note that there is no link between nodes in the same column, i.e., $m_{i,j}$ and $m_{i+1,j}$. Since we measure the phase profiles from multiple antennas, it is quite effective to search for the similar nodes. Thirdly, we search for the path from the graph, that traverses all the $w$ rows as the matching. Here, the path means all the nodes in the path have similar phase variances. If multiple paths exist, we just choose the one with minimum phase variance, which means this path is most likely to be the matching result. Then we can determine the matching tags as follows: for node $m_{i,j}$ in the path, tag $t'_j$ in $D'$ is the matched tag for $t_i$ in $D$ after the movement. Fourthly, we remove the matched tags in $D'$, since they have been matched, and handle the other objects one by one. Finally, we could distinguish the moving objects from the missing objects and the remained tags of $D'$ are the newly joined tags.

Fig. 15 shows an example of the searching flow. We firstly calculate the phase difference nodes $m_{i,j}$ for the object $O_1$, i.e., tag $t_1, t_2, t_3$. We use the gray scale to represent the phase difference. Secondly, we build the graph by connecting the similar phase variance nodes in adjacent rows. Thirdly, we search for the path that traverses all the 3 rows as $m_{1,2} \rightarrow m_{2,3} \rightarrow m_{3,4}$. Therefore, the matching tags for $(t_1, t_2, t_3)$ are $(t'_2, t'_3, t'_4)$, respectively. Fourthly, we remove $(t'_2, t'_3, t'_4)$ from $D'$, and finally search for the matching tags for the object $O_2$. Since we find there is no valid path connecting all the rows of $O_2$, the object $O_2$ is a missing object. Moreover, we can regard tags $t'_2$ and $t'_3$ in $D'$ as the newly joined tags.

7.4 Combating the Identification Errors

The above method solves the problem of distinguishing the moving objects in the ideal environment, where every existing tag is successfully identified and the phase profile is correct. However, due to the thermal noise, a moving tag may be misread in the realistic environments. Since it is unlikely all the tags on the same object encounter such reading errors at the same time, we could leverage the deployment of multiple tags and use a voting-based method to determine the moving object. In particular, for an object attached with $w$ tags, if more than $\lceil w/2 \rceil$ tags occur in the tag set $D$, this object is regarded as a possible moving/missing object. Otherwise, the object is determined as static and the corresponding tags are unmatched due to the multi-path
effect or RF noise. Therefore, the object status can be usually correctly estimated if the majority of the attached tags are identified successfully.

**Algorithm 1** Match the moving tags for object \( O_k \)

**Input:** Phase variance node \( m_{i,j} \) for unmatched tags on object \( O_k \) and attached tag number \( w \) attached on each object

**Output:** Matched tag set \( T \)

1: for \( i = 1 \) to \( w \) and \( t_i \in O_k \) do
2: for \( j = i + 1 \) to \( w \) and \( t_j \in O_k \) do
3: calculate the similarity between two nodes \( \epsilon_{i,j} = m_{i,j} - m_{i',j} \);
4: for Any \( \epsilon_{i,j} \) is less than an empirical threshold \( \rho \) do
5: Build a link from \( m_{i,j} \) to \( m_{i',j} \);
6: end for
7: end for
8: end for
9: Search for the path that contains the most nodes;
10: if More than one path exist then
11: Calculate the phase variance across each path;
12: Select the path with minimum phase variance;
13: end if
14: return \( T = \{ t'[j \text{ node } m_{i,j} \text{ in the selected path}] \} \).

Based on the above analysis, we need to match the majority of the \( w \) tags attached on one object, instead of all the \( w \) tags. So we cannot find a path that links all the \( w \) rows together in Fig. 15. Instead, we need to search for the path that traverses most of the rows in Fig. 15. If more than one path have the same number of nodes, we choose the one with minimum phase variance across the path, similarly. Algorithm 1 presents the pseudo code of our algorithm.

We next analyze the complexity of the algorithm. Consider \( m \) tags from \( D \) and \( n \) moving tags from \( D' \) are matched based on the algorithm. The complexity of a brute matching solution is \( O(n^m) \), which means for each tag in \( D' \), there are \( m + 1 \) possible matching candidates. However, if there are \( w \) tags attached on one object, \( w \times n^2 \) nodes are calculated in the graph. Therefore, the complexity of graph building is \( O(n^2) \), where \( w \) is constant. Besides, the complexity of path searching is less than \( O(n^2) \). Since \( m \) tags at most represent \( m \) objects, the complexity upper bound is \( O(mn^2) \).

8 Performance Evaluation

In the following, we evaluate the performance from three aspects. Firstly, we evaluate the feature extraction scheme based on a small-scale experiment in the realistic environment. Then we evaluate the GM method and the CPV method in terms of the detection accuracy of motion status. Lastly, we evaluate the time efficiency of our methods.

8.1 Evaluating the Accuracy in Feature Extraction

8.1.1 Experiment Settings

We perform an experiment with practical deployment to evaluate the extracted physical-layer features by issuing 50 tags. All signal traces are collected with the GNURadio/USRP platform as shown in Fig. 17. Due to the power limitation of USRP for scattering 50 tags, we use a four-step scheme to collect the real collision signal. Firstly, we emulate the process of slot selection on MATLAB according to C1G2 standard. Secondly, we collect the responding signal trace for each slot on the USRP platform according to the result of slot selection. Thirdly, we extract physical-layer features from singleton and collision slots. Lastly, we modify the frame size based on the number of identified tags for a new frame. In this experiment, the initial frame size is set to 64 according the tag cardinality. Since we focus on the accuracy of physical-layer feature extraction, we do not use a larger tag cardinality, which should have the similar characteristic. For each collision slot, we first set the frame size to 1, which makes the tags collide in the only one slot, and collect the corresponding collision signal. Then we keep the tags unmoved, and set the frame size based on C1G2 protocol, which allows us to collect the ground truth signal trace of each tag in the singleton slot.
To evaluate the robustness of the physical-layer features, we also extract the features from the two-collision signals in two typical complicated environments. First, we vary the distance between the two collided tags to examine the effect of the nearby tags. Secondly, we mimic the storage environment by attaching the tags on the boxes as shown in Fig. 17 and examine the robustness of the features by deploying the boxes in the free-space (FS), reflection environment (RE), side by side (SS), stack environment (ST) and block environment (BL).

8.1.2 Results

We can save 60% frames to extract physical-layer features by decoding the collision signals, while the standard variance of errors of extracted phase profiles and BLF are 9.7° and 4.3 μs respectively. Fig. 16(a) shows the collision distribution and corresponding slot utilization of the first frame. When we extract features from singleton, 2-collision and 3-collision slots, there are only 4.6 (9.2%) tags unidentified on average after the first frame. The identification ratio is about 2 times of C1G2 standard, which only identifies tags in the singleton slot. Further, Fig. 16(b) reports unidentified tags number of each frame. To identify all the tags, C1G2 standard usually need 5 frames typically. However, if we could exploit the collision slot, we could finish the inventory in 2 frames.

Meanwhile, as shown in Fig. 16(c) and 16(d), both errors of phase profiles and BLF can be restricted. We exploit the cumulative distribution function (CDF) to show the error of the extracted physical-layer features from collision signal. The error of 80% signal length are 5 μs and 8 μs for 2-collision and 3-collision respectively. For the phase profile, the error of 80% sample are 8° and 16°, respectively. The errors are caused by the noise imported by the collided tags.

We further present the robustness evaluation in Fig. 16(e)-Fig. 16(h). We find that BLF is independent with the distance between tags, meaning it is a stable inherent feature of each tag. For the phase profile, we find that the phase of static tag (Tag 1) is stable when the distance is larger than 2 cm, meaning the phase profile is affected by nearby tags when they are quite close to each other. For the moving tag (Tag 2), the phase is linearly decreasing as expected. Moreover, for the storage deployment, we find that the SS and ST environments achieve similar performance with the free-space environment, because the tags are scattered in the space. For the RE and BL environments, the tags are either affected by the reflection signal or the block of other tags. Nevertheless, both the BLF and phase variance are still less than 4 samples and 0.4 radian, which are sufficient for the motion detection.

8.2 Evaluating the Accuracy in Moving Tag Detection

8.2.1 Experiment Settings

We further perform extensive simulations to evaluate GM and CPV method over different parameters on MATLAB. In this experiment, we monitor hundreds of tags, which are scattered randomly inside a 10 m × 10 m room and the receiving antennas are attached on each wall randomly.
We deploy the transmitting antenna in the center of room for simplicity. The phase values are measured according to the transmitting distance and wavelength. To simulate the signal noise in the real settings, we also add some signal noise to all the physical-layer features based on the statistical result in the realistic experiments on USRP. We first extract the stationary phase distribution from singleton slot in the tag inventory phase. Then, we randomly select some tags and move them 20cm away from their initial positions. In continuous polling phase, we extract the current phase distribution from singleton and collision slots. For every polling cycle, we utilize the GM method to detect the moving tags. Furthermore, when we attach several tags on one object, we leverage the CPV method to detect the moving/missing/inserting objects based on the GM results.

In this experiment, we study the effect of the following parameters on the detection accuracy in the GM method: phase variance $\sigma$, number of moving tags $\eta$, number of antennas $\tau$ and number of monitoring tags $\omega$. $\kappa$ is set to $2\omega$ for simplicity. The default setup is $\sigma = 8$, $\eta = 3$, $\tau = 8$ and $\omega = 500$. Furthermore, we study the effect of number of antennas, number of tags, number of moving tags and number of tags attached on one object, when we leverage the CPV method to detect the moving objects, the missing object and the inserting tags. Finally, we study the effect of the antenna deployment when we deploy 8 receiving antennas in the monitoring room. We utilize false positive error (FP) and false negative error (FN) to evaluate the accuracy of GM method. False positive error means the stationary tags are identified as moving tags, and false negative error means the moving tags are identified as stationary tags. Meanwhile, we use similar metrics to evaluate the accuracy of detecting the moving objects, missing object and inserting tags of the CPV method.

8.2.2 Results

GM method can correctly detect 85% of the moving tags with about 6% false positive errors. Fig. 18(a)~18(d) report the results from different points of views. We note the number of antennas and tags affect the accuracy obviously, while the number of moving tags slightly affect the accuracy. Specifically, FN errors are mainly caused by the relative scales between antenna number and tag number. When the number of antennas is relatively small compared with the number of tags, we cannot exclusively distinguish all tags due to lower dimension of phase profiles. As a result, some moving tags cannot be detected, which leads to high FN errors. As the number of antennas increases to be comparable with the cardinality of tags, we can accurately detect the moving tags with high probability. FP errors are mainly caused by the measuring phase variance. When the phase profiles are relatively accurate, it is less likely to identify a stationary tag as a moving one. Besides, both FP and FN are not sensitive about the number of moving tags, because the probability of correct and incorrect identifying each tag are the same. Hence, in Fig. 18(c) both FP and FN vary a little.

CPV method can correctly detect the moving objects with only 7% false negative errors. Fig. 18(e)~18(j) present the results of the CPV method. Firstly, comparing Fig. 18(e) and Fig. 18(f) with Fig. 18(a) and Fig. 18(b), we find that the CPV method can dramatically decrease the FP error and the FN error, because these false detected tags usually can be filtered based on the coherent phase variance of the multiple tags. Particularly, the FP error is reduced to 0, meaning that no stationary objects are detected as moving objects based on the CPV method. From the other four figures, we also find all the FP errors are less than 1% based on the redundant tags. Besides, we also reduce the FN error to about 7%, which indicates the efficiency of the CPV method. In Fig. 18(h), we find that when we increase the number of tags attached on one object, we can slightly reduce the FP and FN errors, which means 3 or 5 tags are enough for accurately detection. Moreover, we can further detect both the missing objects and inserting tags accurately based on the coherent phase variance of the moving tags. We note that in Fig. 18(i) and 18(j), the accuracy of the missing objects and the inserting tags are both larger than 98%, which indicates that the CPV method is useful in detecting different kinds to object status. Such improvement compared with the GM method is achieved by attaching the redundant tags on the objects.

The scattered deployment of receiving antennas can improve the accuracy of motion detection. We try six different antenna deployments as shown in Fig. 18(k), including random deployment (RN) and uniform deployment on 1 line (1LN), on 2 adjacent lines (2AL), on 2 opposite lines (2OL), on 4 lines (4LN) and on a circle (CIRC). Based on the detection result in Fig. 18(l), we find 1LN achieves the worst performance, which CIRC and 4LN achieve the best performance. This is because when we scatter the receiving antennas around the room (e.g., 4LN or CIRC), the discrimination of the phase profile received by each antenna is maximized, and thus we can achieve better detection accuracy.

8.3 Evaluating the Time Efficiency

The proposed Fast Tag Polling (FTP) scheme can save 80% inventory time when querying more than hundreds of tags compared with C1G2 protocol. We compare with C1G2 standard to evaluate the time efficiency. Specifically, we evaluate the querying time and the cost frames of conducting a typical query cycle. We use the time of each period in [1], [2] to estimate the querying time, i.e., 1ms for collision slot and 4ms for singleton slot. We respectively set the initial frame size based on the number of tags for C1G2 and FTP, which can maximize the slot utilization for either C1G2 or FTP.

As shown in Fig. 19(a), the modified protocol can save 80% time compared with C1G2 standard through decoding the collision slot. As for the cost frames, even though we need to identify hundreds, the modified protocol can extract features within 2 frames for most cases. By setting initial frame size appropriately based on the number of tags, we can identify over 80% tags in the first frame. Then the rest tags can be easily identified with a next frame. But C1G2 standard can only identify about 35% tags in the first frame, which leads 7 ~ 10 frames in total.

Both the GM method and the CPV method can efficiently detect the moving objects within 0.1s. Besides, we also evaluate the computation latency of both the GM and CPV method by varying the number of tags and moving objects. Particularly, since the GM method is based on the Hungarian algorithm, which is $O(n^3)$ complexity, to speed up the method in application, we first select the obvious stationary tags, whose distances $d_{i,j}$ are $\infty$ except for one
small distance. Then we can use a smaller distance matrix to detect the possible moving objects. Fig. 19(c) and Fig. 19(d) present the corresponding results. We find that both the GM method and CPV method can be efficiently executed within 0.05s no matter we increase the tag cardinality or the moving objects. Particularly, the CPV method is slightly slower than the GM method, because we need to decide all the moving/missing/inserting tags in the method. Overall, all the proposed methods can efficiently detect the moving objects with small latency.

9 DISCUSSION

Limitation on USRP. Since the USRP follows the UHF G1C2 protocol to communicate with the tag, the collision RN16 signal received by the USRP is similar to the COTS RFID reader. However, the COTS RFID reader does not provide the collision information. Therefore, the whole system is evaluated on the USRP testbed instead of the COTS RFID reader, and we used the USRP testbed proof the feasibility of the detecting method. Even though the COTS RFID reader can also detect the motion status of the tags, it only exploits the singleton slot, thus the time efficiency is much lower than the USRP. In regard to the power constraint of USRP, the identification distance of the USRP reader in the lab environment is about 3m, which is smaller than the identification distance of COTS RFID reader, i.e., 5 – 6m. But one can attach an amplifier between the USRP and antenna to increase the transmitting power. Since the RFID system is forward-link limitation [28], the increment on the transmitting power can efficiently improve the identification distance of the RFID system, which can support the large-scale interragination.

Non-linear combination of collision signal. Recently, several work try to solve the non-linear combination of collision signals in the real environments [7–9]. Since the nearby tags are transmitting together, the tags may backscatter the signals from nearby tags, leading to the non-linear combination [9]. Apparently, we can hardly extract the phase profiles from these non-linear combinations. Thus, we can recognize such special combination by examining the signal summation, then regard these tags as unsolved collision signal and ask these tags to transmit in the following frames. Even though we may increase some time latency, we can still achieve the whole phase distribution for moving detection by neglecting these non-linear combination signal.

Deployment cost of the multiple antennas. Based on our experiment results, the detection accuracy is increased with growing number of antennas, which suggests that we can deploy more antennas to monitor large-scale tags. Actually, it is a trade-off between the number of antennas and the detection accuracy. If the users want to deploy less antennas to save money, our system can not uniquely determine the moving tags. However, since the two phase distributions are different after the tag’s movement, our system can still detect the moving event and estimate the phase profiles of the moving tags. Here, the phase profiles cannot uniquely represent one tag due to the insufficient number of antennas, but our system can still provide several possible moving objects. Then, the users can further interrogate these tags to determine the actual moving tags. Based on the understanding, our system can still detect the moving event when the number of antennas is not sufficient, but more inventory cycles and time are required to uniquely determine the moving tags.

10 CONCLUSION

In this paper, we propose a fast approach to detect the moving tags in the monitoring area. We achieve the time efficiency by decoding collisions from the physical layer. We are able to extract two physical-layer features, i.e., the phase profile and the backscatter link frequency, to distinguish different tags. By resolving the physical-layer features from collisions, we are able to derive the motion status of multiple tags. Experiment result shows that when monitoring 1000 tags, our solution can accurately detect the moving tags while reducing 80% time compared with the state-of-art solutions.

ACKNOWLEDGMENT

This work is supported in part by National Natural Science Foundation of China under Grant Nos. 61872174, 61832008, 61872173, 61802169; JiangSu Natural Science Foundation under Grant No. BK20180325. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization. Yingying(Jennifer) Chen is partially supported by the US National Science Foundation Grant CNS-1820624 and CNS-1826647. Lei Xie is the corresponding author.

REFERENCES


