Moving Tag Detection via Physical Layer Analysis for Large-Scale RFID Systems

Chuyu Wang, Lei Xie, Wei Wang, Tao Xue, Sanglu Lu
State Key Laboratory for Novel Software Technology, Nanjing University, China
Email: {wangcyu217, txue}@dislab.nju.edu.cn, {lxie, ww, sanglu}@nju.edu.cn

Abstract—In a number of RFID-based applications such as logistics monitoring, the RFID systems are deployed to monitor a large number of RFID tags. They are usually required to track the movement of all tags in a real-time approach, since the tagged-goods are moved in and out in a rather frequent approach. However, a typical cycle of tag inventory in COTS RFID system usually takes tens of seconds to interrogate hundreds of RFID tags. This hinders the system to track the movement of all tags in time. One critical issue in such type of tag monitoring is to efficiently distinguish the motion status of all tags, i.e., stationary or moving. According to the motion status of different tags, the state-of-art localization schemes can further track those moving tags, instead of tracking all tags. In this paper, we propose a real-time approach to detect the moving tags in the monitoring area, which is a fundamental premise to support tracking the movement of all tags. We achieve the time efficiency by decoding collisions from the physical layer. Instead of using the EPC ID, which cannot be decoded in collision slots, we are able to extract two kinds of physical-layer features of RFID tags, i.e., the phase profile and the backscatter link frequency, to distinguish among different tags in different positions. By resolving the two physical-layer features from the tag collisions, we are able to derive the motion status of all tags concurrently and greatly improve the time-efficiency. Experiment result shows that our solution can accurately detect the moving tags while reducing 80% of inventory time compared with the state-of-art solutions.

Index Terms—RFID, Collision Decoding, Tag Inventory

I. INTRODUCTION

With the rapid proliferation of RFID-based applications, RFID tags have been deployed into various applications in increasingly large numbers. For example, in the application of logistics monitoring, there are usually more than hundreds of goods attached with RFID tags in the monitoring area. Since the tagged-goods are moved in and out in a rather frequent approach, the RFID systems are usually required to track the movement of all tags in a real-time approach. However, a typical round of tag inventory in COTS RFID system usually takes tens of seconds to interrogate hundreds of RFID tags [1, 2]. This greatly hinders the system to track the movement of all tags in time. One critical issue in such type of tag monitoring is to efficiently distinguish the motion status of all tags, i.e., stationary or moving. For the “moving” tags, we can leverage the state-of-art localization schemes to track them; for the “stationary” tags, we do not need to track them any more, since they are supposed to be statically placed in a specified position. In most situations, the stationary tags occupy a rather large proportion while the moving tags occupy only a small proportion in regard to a certain moment. Therefore, it is essential to devise a moving tag detection scheme to detect the motion status of all tags. In this way, we only need to focus on tracking the moving tags, saving a lot of effort which might be wasted in tracking stationary tags.

According to the prior art, in order to track the moving tags in the monitoring area, two schemes are essentially required, i.e., a fast tag inventory scheme to interrogate tags, and an effective positioning scheme to detect the motion status of the tags. For the tag inventory in RFID systems, polling protocols [3, 4] have been proposed to improve the time efficiency. However, they still take tens of seconds to interrogate RFID tags in a real environment, when the cardinality of tags is more than several hundreds. The inefficiency is primarily due to the waste of the collision slots, which usually occupy a fairly large proportion in the overall time slots. Hence, even schemes based on the pooling protocols [5, 6] can efficiently detect the moving tags based on the statistical signal information, but also suffer from the effect of collision slots. Recent research works consider to redesign new protocols or modify the EPC C1G2 protocols to make use of the collision slots for better time efficiency [7–9]. However, they have not yet considered to detect the motion status of the tags. Some protocols [3, 10] are also proposed to detect the missing tags via the physical-layer symbols. But they cannot be used to detect the moving tags since a moving tag can still exist in the monitoring area instead of missing. For the positioning scheme of RFID systems, the state-of-the-art localization schemes [11, 12] usually locate the tags one by one, and the time delay of localizing a unique object is up to several hundreds of milliseconds. When dealing with more than hundreds of RFID tags it is impossible to concurrently locate all tags in a real time approach.

In this paper, we propose a real-time approach to detect the moving tags in large scale RFID systems, which is a fundamental premise to support tracking the movement of all tags. Since a missing tag must be moved first, it can also be simplified as a moving tag detection problem. In our problem, we achieve the time efficiency by effectively decoding collisions from the physical layer. Instead of using the EPC ID, which cannot be decoded in collision slots, we are able to extract two kinds of physical-layer features of RFID tags, i.e., the phase profile and the backscatter link frequency, to distinguish the tags in different positions. These physical-layer features serve as fingerprints of tag identities and positions. By resolving the two physical-layer features from the tag collisions, we are able to derive the motion
status of multiple tags simultaneously, and greatly improve the time-efficiency. Specifically, we propose a two-phase tag monitoring scheme including the tag inventory and continuous polling. In the tag inventory phase, the RFID reader constructs a physical fingerprint for each tag individually via traditional tag inventory. In the continuous polling phase, the RFID reader continuously issues multiple query cycles to interrogate the tags. For each polling cycle, the RFID reader measures a new distribution of physical-layer features via both the singleton and collision slots. By matching the updated distributions to the original distributions, our solution is able to efficiently detect the moving tags. The above two phases are executed alternately, and the time overhead of the tag inventory phase can be amortized by the following multiple polling cycles, such that the overall time-efficiency is achieved.

There are three key technical challenges. The first challenge is to achieve real-time time efficiency in large scale RFID systems. In a large scale RFID system, it is rather difficult to continuously update the monitoring results within limited time intervals. To address this challenge, we propose a two-phase monitoring scheme including a normal tag inventory phase and multiple fast tag polling phases, we significantly improve the time efficiency in extracting the physical-layer features via decoding the tag collisions. The second challenge is to detect the motion status of all tags via the physical-layer feature decoding the tag collisions. The third challenge is to extract the above physical-layer features from the collisions of multiple tag responses. To address this challenge, we exploit the relationship between the physical-layer features and the motion status of tags. We find that the phase value from the tag’s response changes even if the specified tag is moved with a small distance, while the backscatter link frequency of the tag’s response has high degree of distinction among different tags. We thus leverage these physical-layer features to detect the motion status of specified tags. The third challenge is to extract the above physical-layer features from the collisions of multiple tag responses. To address this challenge, we recover each tag response according to the geometrical characteristic of the collision signals in I-Q plane, and extract the phase profile of each tag response. Further, we refer to special patterns to identify the starting and ending parts of recovered RF signals based on cross-correlation, and extract the backscatter link frequency from the signal length of each tag.

We make three contributions in this paper. First, to the best of our knowledge, we are the first to propose a moving tag detection scheme for tag monitoring by leveraging the physical-layer features, which is a fundamental premise for tracking the movement of RFID tags in large-scale RFID systems. Second, our solution is able to accurately detect the motion status of all tags, by referring to the physical-layer features, including the phase profile and backscatter link frequency. Moreover, we extract these physical-layer features of multiple tags from collision slots, which significantly improves the time efficiency. Third, we implemented a prototype system and evaluated its performance in realistic settings. Experiment result shows that our solution can accurately detect the moving tags while reducing 80% of inventory time compared with state of arts solutions.

II. RELATED WORKS

Collision Recovery Many works focus on how to extract tag cardinality [13] or recover the tag signal [8, 9] from the collision signals based on the specialized instruments like USRP. Since Buettner et al. propose a Software Defined Radio based UHF-RFID reader [14], several researchers further leverage this platform to deal with the collision problems [7–9, 15]. Wang et al. [7] implement a new scheme which enables rateless code transmitting. [8, 9] use the time-domain separation to recovery the data from the collision signals. Hou et al. [13] present a physical-layer cardinality estimator from the collision signals for large scale RFID system.

Physical Layer Identification Previous studies [16, 17] focus on physical-layer identification by leveraging the hardware imperfection in tag manufacturing. Davide et al.[16] distinguish from different tags based on the frequency difference $\Delta f_{11}$ caused by manufacturing imperfection of tags. Han et al. [18] leverage the internal similarity among pulses of tags’ RN16 preamble signals as the fingerprint for distinguishing. Zheng et al. [10] employ a method to detect the missing tags based on physical-layer signals. Different from previous work, in this paper, we focus on how to design a real-time tag monitoring scheme to efficiently detect the motion status of all tags, so as to further support tracking the movement of all tags. We aim to improve both the time efficiency in tag inventory and the accuracy in detecting the motion status of all tags.

III. SYSTEM OVERVIEW

A. Design Goals

In this paper, we propose a real-time approach to detect the motion status of all tags in the monitoring area, so as to further support tracking the movement of all tags. Because the tags may change their motion status any time, we need to continuously update the motion status within a limited time interval. Therefore, our objective in designing a moving tag detection scheme is to improve both the time efficiency in tag inventory and the accuracy in detecting the motion status of all tags. 1) The average time for each cycle of tag inventory should be sufficiently reduced to achieve the real-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 identified as moving tags. b) False negative errors: the moving tags are identified as stationary tags. Both of the two errors should be effectively reduced in detecting the motion status.

B. System Framework

In order to effectively detect the motion status of all tags, we exploit the relationship between the physical-layer features and the motion status of tags. The following two physical-layer features are investigated: 1) Phase profile: it is the phase value of an RF signal. 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 determines the tag’s data rate in
the 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, 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.

![System framework](image1)

**Fig. 1:** System framework

As for detecting the motion status of tags, we propose a two-phase monitoring scheme, including the tag inventory and continuous polling phase, to efficiently extract the physical-layer features for detection. In the tag inventory phase, the reader issues multiple query cycles to extract the physical-layer features of all the tags in stationary status. In the continuous polling phase, the reader continuously issues multiple query cycles to extract physical-layer features of tags in real-time. By comparing the real-time features with the stationary features, we utilize a Graph Matching Method (GMM) to detect the motion status of tags in every query cycle of continuous polling phase. The continuous polling phase contains multiple real-time query cycles to amortize the time spent in the inventory phase. We show the whole framework in Fig. 1.

### IV. Physical-Layer Features

In this section, we demonstrate the concept of our physical-layer features via realistic experiments. We implement a software defined reader (SDR reader) according to the Gen2 project [14]. Specifically, we operate the Gen2 project on our USRP platform with two FLEX-900 daughter boards and two Larid 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.

**A. The Response of a Normal Singleton Slot**

Fig. 2 illustrates a typical slot in RFID systems, which is collected from USRP. The reader sends a QUERY/QRep command to start a slot. All the tags that select this slot, will transmit its RN16 to the reader. If the reader succeeds in decoding the RN16 bits, it then sends an ACK to the tag, that tells the tag to transmit its EPC-ID. During the tag response, the reader keeps transmitting continuous wave (CW) to supply power. Hence, there are two kinds of tag response generally: 1) RN16 period, responding the QUERY or QRep command from the reader. 2) EPC-ID period, answering the ACK command. In fact, both the RN16 signal and the EPC-ID signal contain preamble, data bits and check bits. As a result, the time of the EPC-ID period is about 4 times longer than that of the RN16 period as shown in Fig. 2. Meanwhile, since the time interval between the two responses is so small, the position and wireless environment can be regarded unchanged. So we can extract the physical-layer feature directly from the RN16 and omit the EPC-ID signal.

![A typical singleton slot in RFID systems](image2)

**Fig. 2:** A typical singleton slot in RFID systems

**B. Phase Profile**

In RFID systems, the tag transmits data using backscattering modulation. Hence, the received signal of one tag is:

\[
y(t) = (A_1 + x(t)A_2 \cdot \cos(2\pi f_c t + \beta)) \cdot \cos(2\pi f_c t + \beta) + n(t),
\]

where \( A_1 \cdot \cos(2\pi f_c t + \beta) \) is the signal of carrier, \( x(t)A_2 \cdot \cos(2\pi f_c t + \beta) \cdot \cos(2\pi f_c t + \beta) \) is the signal of tag and \( n(t) \) is the ambient noise. Here, \( x(t) \) is the binary bits sent by the tag. After converting the signal to baseband by the removing carrier \( \cos(2\pi f_c t) \), the baseband signal can be represented as:

\[
s(t) = A_1e^{j\beta} + x(t)A_2e^{j(2\pi f_c t + \theta + \beta)} + \tilde{n}(t).
\]

Therefore, the actual received signal is a superposition of carrier wave and backscattered signal.

Intuitively, we can model the received signal from a single tag response in I-Q plane as shown in Fig. 3. The received signal consists of two parts: 1) leakage signal: the constant carrier signal (i.e., CW), 2) backscattered signal: the modulated tag signal. As for the phase value of the backscattered signal, it can be represented as:

\[
\theta = \Phi - \beta,
\]

which is the difference between the carrier signal phase \( \beta \) and the backscattered signal phase \( \Phi \) in Fig. 3. We call \( \theta \) the phase profile of the tag in this work.

![Model of the received signal of a single tag](image3)

**Fig. 3:** Model of the received signal of a single tag

We carry out trace-driven evaluations to study the property of the phase profile. Firstly, we evaluate the stability by conducting an empirical experiments on 50 tags with random deployments. For each setting, we measure 100 phase values by querying each tag 100 times. The results are normalized by subtracting the average phase value of each result set. As shown in Fig. 4(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
C. Backscattered Link Frequency

Due to manufacturing imperfection, the backscatter link frequency (BLF) of the response signal, i.e., $f_i$ in Eq. (1), varies among different tags. Thus, it can be used to distinguish tags as in [16, 18]. In fact, $f_i$ determines the length of a square wave, i.e., the duration of high/low voltage in Fig. 5(a). In regard to the typical modulation schemes in RFID systems, i.e., FM0 and Miller Modulation, data-0 and data-1 share the same amount of square waves. Taking Miller-4 Modulation as an example, both data-0 and data-1 contain eight square waves as shown in Fig. 5(a). Since the length of RN16 signal is fixed, it is reasonable to use the corresponding signal length of RN16 to represent BLF. The signal length of RN16 can be also transmitted to the BLF based on the actual sampling rate.

We use the cross-correlation 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 as shown in Fig. 5(b). Then we find the window whose cross-correlation value is the maximum, and record the position of the window. In this example, we locate the starting point of RN16 based on the special preamble sequence. Similarly we can locate the ending point using “dummy 1”. We use the number of samples between the starting and the ending point to represent 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 to extract the signal lengths in different positions. As shown in Fig. 6(a), the signal length of 50 tags are randomly distributed from 7620 to 7690. This indicates that the signal length can be used to distinguish among tags. Furthermore, we draw the histogram of the normalized variance of signal length in Fig. 6(b). The signal length is relatively stable with an average deviation of 2 samples, which is equal to 1µs. Therefore, BLF is stable and distinctive even though they are at different positions. But there are also some tags with the same BLF value which means we need to combine it with other attributes.

V. EXTRACT PHYSICAL-LAYER FEATURES

In the previous section, we have demonstrated the physical-layer features in the singleton slots. We can collect the physical-layer features in these singleton slots in the tag inventory phase. However, it is still time inefficiency because we cannot avoid useless collisions in RFID system. If $c$ tags select the same slot to transmit the data, a $c$ collision happens. When $N$ tags select slots randomly from a frame with $f$ slots according to the Binomial distribution, the probability of a $c$-collision slot can be expressed as:

$$Pr(c) = \binom{N}{c} \left(\frac{1}{f}\right)^c \left(1 - \frac{1}{f}\right)^{N-c}.$$  (4)

Fig. 7 shows the theoretical probability distribution of collision slot. The C1G2 standard improves the time efficiency by maximizing $Pr(1)$, thus at most 36.8% slots are singleton. 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 92% tags in a frame, which is 2.5 times compared with current protocol. Hence, we focus on how to extract physical-layer features from 2-collision and 3-collision slots.

A. Model of Collision Signal

Based on Eq. (2), the received baseband signal of a $c$-collision slot can be expressed as:

$$s(t) = Ae^{j\beta} + \sum_{i=1}^{c} x_i(t) h_i + n(t).$$  (5)
Here, \( x_i(t) \) is the binary bits sent by tag \( i \) over time \( t \). \( h_i \) is the channel coefficient of tag \( i \) and can be written as:

\[
h_i = B_i e^{j(\theta_i t + \beta_i)},
\]

where \( B_i \) is the amplitude and \( \theta_i \) is the phase profile of tag \( i \). Actually, \( h_i \) represents a vector in I-Q plane, thus the collided signal can be represented as the superposition of the vectors.

Taking a 2-collision as an example in Fig. 8(a), since both tag A and B send a binary bit, we can use four different states. \( S_0 = x_A x_B = 0 \cdot 0 \) means both tag A and B are transmitting the same data during the preamble, signals are switching between state \( S_0 \) and \( S_2 \) during the preamble as shown in Fig. 8(b). Therefore, we can determine the state \( S_3 \) using the preamble part. Hence, 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 \( h_2 = C(S_2) - C(S_0) \), where \( C(S_i) \) is \( S_i \) in the I-Q plane.

2) Channel Coefficients Estimation for 3-collision Slot: When the number of collided tags increases to three, the sign 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 signal vectors in I-Q plane, the 8 states constitute a parallelepiped. Our basic idea is that since 3-collision signal can be always represented as the superposition of 2-collision signal and one tag signal, we can find and resolve a 2-collision problem inside the 3-collision problem and then handle the 3rd tag signal. For example, parallelogram \( S_0 S_2 S_3 S_4 \) and \( S_1 S_2 S_3 S_5 \) in Fig. 8(c) 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. 8(c), or a plane tetrahedron as shown in Fig. 8(d). For the parallelogram case, we have found the parallelogram such as Fig. 8(c). For the plane tetrahedron case, the edges among the four states inside the parallelepiped intersect at one state, e.g., in Fig. 8(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 \( S_3 \) with \( S_4 \) in Fig. 8(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 slots are as follows: Firstly, we cluster signals into 8 clusters, similar as in the 2-collision problem. Secondly, 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, 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:

\[
\text{Sim} = \frac{\|\vec{a} \cdot \vec{b}\|}{\|\vec{a}\| \times \|\vec{b}\|} \times \frac{1}{(\|\vec{a}\| - \|\vec{b}\|)},
\]

where \((\vec{a}, \vec{b})\) are the possible edge pair. The first part of Eq(8) is the cosine value of the edge pair and the second 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 based on the geometric construction. The parallelogram will always include either state \(S_0\) or \(S_7\) because they are symmetric. 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. Then the channel coefficient of the 3rd tag can be easily computed based on \(S_0\), \(S_7\) and the first two channel coefficients, since \(S_7\) contains the channel coefficients of three tags and \(S_0\) contains none.

C. Measure the Physical-Layer Features from Collision Signal

After extracting the channel coefficients from the collision slot, we can directly compute the phase profile based on Eq. (3). For backscatter link frequency (BLF), we utilize Eq. (7) to recover the RN16 signal of each tag. It is difficult to decode these RN16 signals into complete binary sequences due to the ambient noise. But according to the special encoded pattern of preamble and “dummy 1” as shown in Fig. 5(a), we can decide the starting and ending point of each RN16 signal using cross-correlation. Here we adopt the same cross-correlation process as in Section IV-C to decide the signal length. Then we use the signal length as the indicator of BLF.

D. Case Study

Fig. 9 is 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 I-Q plane as shown in Fig. 9(a). In the following we denote each state as the amplitude rank of states in I-Q plane. Secondly, we determine state \(S_0\) from continuous wave and state \(S_7\) from the preamble as shown in Fig. 9(b). Thirdly, we search for the parallelogram based on the first 5 states. The left quadrilateral in Fig. 9(b) is the estimated parallelogram from the first 5 states and the right one is the symmetrical one. Fourthly, we calculate the channel coefficients including the phase profiles as the arrow in the figure. Lastly, we recover the RN16 signal of each tag according to Eq.(7). Here, we use “1” to represent high voltage and “-1” to represent low voltage. We compute the signal length of RN16 as the indicator of BLF based on cross-correlation values. Fig. 9(c) shows the ending part of three separated RN16 signals and points out the slide window which has the maximum cross-correlation value.

VI. DETECT THE MOVING TAGS

A. Motivation and Approaches

After we extract the phase profiles from the collision signals, we demonstrate how to detect the moving tags by using the phase profiles. Since the phase profile always changes even if the tag is moved a small distance, the basic idea is to compare the updated phase profiles with the stationary phase profiles. Suppose we need to monitor \(N\) tags, we obtain the stationary phase profiles of the \(N\) tags in the tag inventory phase. We use a vector \(P = (\theta_1, \theta_2, \cdots, \theta_N)\) to represent the phase distribution, where \(\theta_i\) is the \(i\)th phase profile in distribution \(P\). Then in the each cycle of the continuous polling phase, we obtain an updated phase distribution \(P' = (\theta_1', \theta_2', \cdots, \theta_N')\). We compare \(P'\) with \(P\) to detect the moving tags in each polling cycle.

Traditional C1G2 protocol usually costs tens of seconds to obtain the tag information, which can build the phase distribution \(P'\). It is 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 Scheme (FTPS) by suppressing the transmitting of EPC-ID signal with a new command \(\text{QrepSup}\). The \(\text{QrepSup}\) command is used to respond the RN16 signal of the tags. It not only starts the next slot like the Qrep command in C1G2, but also makes the tags, which
is resolved in the collision slot, silent for the following frames in the query cycle. On the other hand, all the unidentified tags, caused by the unresolved collision or environment factors, will receive the Qrep command and respond in the next frame similar as C1G2 protocol. Based on FTPS, we can acquire the phase distribution $P'$ in a short time in every polling cycle.

**B. Construct the Multi-dimensional Phase Profiles**

Even though the movements of tags lead to the difference between the two phase distributions $P$ and $P'$, but the phase profile extracted from one antenna cannot sufficiently detect 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) Errors of phase measurement will affect the accuracy of detecting the movements.

Fig. 10 shows a simple example. $P_1$ and $P_2$ are two stationary phase distributions of five tags $T_1$~$T_5$ measured by two antennas in the tag inventory phase. When tag $T_3$ is moved, its phase profiles will change in both $P_1$ and $P_2$. But since $T_2$ has similar phase profile with $T_3$ in $P_3$, we cannot accurately determine which is the moving tag. Similarly, $T_1$ has similar phase profile with $T_3$ in $P_2$, which makes the detection confusing. However, if we can combine the two phase profiles of the same tag in $P_1$ and $P_2$ together, every tag will have a special phase profile of two phase values. We call it a two-dimensional phase profile. In the figure, we use a matrix to depict the phase distribution of the two-dimensional phase profiles. Thus, $T_3$ can be accurately detected because no tag has similar two-dimensional phase profile as it.

![Fig. 10: Example of the multi-dimensional phase profile](image)

In actual applications, we can exploit an SDR reader [19] with a transmitting antenna and two receiving antennas to acquire the two distributions $P_1$ and $P_2$. But it is difficult to construct a two-dimensional phase profile simply based on the phase values from two antennas. Suppose tag $T_2$ and $T_3$ collide in a 2-collision and we have extracted 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 ($7^\circ, 192^\circ$) belong to the same tag or not, because we do not have the tag IDs. Obviously, it is not practical to enumerate all the possible matchings when we have more antennas.

To solve the problem, we use the backscatter link frequency (BLF) to match them. Because the two receiving antennas receive the same signal in the physical layer, the extracted BLFs of the same tag from the two antennas should match perfectly. Suppose we have also extracted the corresponding BLF values for each phase profile, then we are more likely to combine $7^\circ$ and $192^\circ$ together because the extracted BLFs of tag $T_2$ should be similar from the two receiving antennas. Such matching method can be easily extended 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. If it is a $c$-collision slot, we combine the phase profiles received from different antennas by matching the BLFs to construct $c$ multi-dimensional phase profiles.

**C. Detect the Moving Tags via Graph Matching**

Based on the two phase distributions $P$ and $P'$, we demonstrate how to detect the moving tags. The intuition is that for a stationary tag, its phase profile should be stable. Considering the measurement errors, the distance between the two phase profiles from the same tag should be smaller compared with the distances from different tags. Here, we denote 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:

$$d_{i,j} = ||\theta_i - \theta_j'||.$$  \hspace{1cm} (9)

For a moving tag, because it has changed its position, the distances from other phase profiles should be large in a statistical manner. Suppose $\kappa$ is an empirical threshold about the phase variance, we can set $d_{i,j}$, whose value exceeds $\kappa$, to $\infty$ to filter the parings, which are not the phase pairs of a stationary tag. 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.

It is not suitable to enumerate all the possible matchings since the complexity is $O(n!)$. So we propose a Graph Matching Method (GMM) by employing the Hungarian algorithm [20] to solve the problem in polynomial time. Firstly, we calculate the pairwise distance between distributions $P$ and $P'$ according to Eq. (9). 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 $\kappa$, 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. 11 shows an example of the process. We use two gray matrices to represent the two distributions $P$ and $P'$ of five tags from three antennas. And we use dashed lines to represent all the possible pairing 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 in the figure.

In this work, 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, BLF can further help distinguish tags. Since BLF is an inherent feature, it will not be changed by the position. If we add BLF into the distance calculation of GMM, we can filter the phase profiles pair, whose phase distance is small but BLF distance is large.
VII. 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 GMM solution in terms of the detection accuracy of motion status. Lastly, we compare FTPS and C1G2 standard to evaluate the time efficiency.

A. Evaluate the Accuracy in Feature Extraction

1) Experiment Settings: We perform a realistic experiment to evaluate the extracted physical-layer features by issuing 50 tags. All signal traces are collected with the GNURadio/USRP platform as described in Section IV. Due to the power limitation of USRP for scattering 50 tags, we use a four-step scheme to collect the realistic 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.

2) Results: GMM can correctly detect 85% of the moving tags with about 6% false positive errors. Fig. 12(e)–12(h) 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. This is because when fixing other parameters, the probability of correct and incorrect identifying each tag is the same. Hence in Fig. 12(g) both FP and FN varies a little.

B. Evaluate the Accuracy in Moving Tag Detection

1) Experiment Settings: We further perform extensive simulations to evaluate GMM over different parameters on MATLAB. In this experiment, we monitor hundreds of tags, which are scattered randomly inside a 10m x 10m 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. We first extract the stationary phase distribution from singleton slot in the tag inventory phase. Then in continuous polling phase, we extract the real-time phase distribution from singleton and collision slots. For every polling cycle, we utilize the Hungarian algorithm to detect the moving tags.

In this experiment, we study the effect of the following parameters on the detection accuracy: 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. We utilize false positive error (FP) and false negative error (FN) to evaluate the accuracy. False positive error means the stationary tags is identified as moving tags, and false negative error means the moving tags is identified as stationary tags.

Because phase values vary in accordance with Gaussian distribution as evaluated. We import the variances according to the collision type. For singleton slot, a fixed standard deviation of $4^\circ$ is used based on the evaluation in Section IV-B. For collision slot, we import parameters $\sigma$ and $2\sigma$ to denote the standard deviations of 2-collision and 3-collision respectively. We cannot extracted phase profiles from $4^\circ$-collision slots.

2) Results: GMM can correctly detect 85% of the moving tags with about 6% false positive errors. Fig. 12(e)–12(h) 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. This is because when fixing other parameters, the probability of correct and incorrect identifying each tag is the same. Hence in Fig. 12(g) both FP and FN varies a little.

C. Evaluate the Time Efficiency

FTPS 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 set the initial frame size based on the number of tags.

As shown in Fig. 13(a), the modified protocol can save 80% time compared with C1G2 standard through decoding the
In this paper, we propose a real-time 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. 61472185, 61373129, 61321491, 91218302, 61502224; JiangSu Natural Science Foundation, No. BK20151390; Key Project of Jiangsu Research Program under Grant No. BE2013116; EU FP7 IRSES MobileCloud Project under Grant No. 612212; CCF-Tencent Open Fund; China Postdoctor Science Fund under Grant No. 2015M570434. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization. Lei Xie is the corresponding author.

**REFERENCES**


