PHY Assisted Tree-based RFID Identification

Yuxiao Hou, Yuanqing Zheng
Department of Computing, The Hong Kong Polytechnic University, Hong Kong
{yxhou, yqzheng}@polyu.edu.hk

***Abstract*—Tree-based RFID identification adopts a binary-tree structure to collect IDs of an unknown set. Tag IDs locate at the leaf nodes and the reader queries through intermediate tree nodes and converges to these IDs using feedbacks from tag responses. Existing works cannot function well under random ID distribution as they ignore the distribution information hidden in the physical-layer signal of colliding tags. Different from them, we introduce *PHY-Tree*, a novel tree-based scheme that collects two types of distribution information from every encountered colliding signal. First, we detect if all colliding tags send the same bit content at each bit index by looking into inherent temporal features of the tag modulation schemes. If such resonant states are detected, either left or right branch of a certain subtree can be trimmed horizontally. Second, we estimate the number of colliding tags in a slot by computing a related metric defined over the signal's constellation map, based on which nodes in the same layers of a certain subtree can be skipped vertically. Evaluations from both experiments and simulations demonstrate that *PHY-Tree* outperforms state-of-the-art schemes by at least 1.79×.**

I. Introduction

RFID systems [27] are widely deployed to label and track items in various applications, such as inventory management [29, 31], access control [3, 9], human-machine interaction [26], localization and mobility tracking [23, 25, 28]. One fundamental operation in RFID systems is to read tag IDs (a.k.a., RFID identification). Two major types of RFID identification schemes are ALOHA-based and tree-based. In ALOHA based schemes [17, 30], each tag randomly selects a time slot and responds to reader's query, leading to frequent tag collisions and low communication efficiency.

In contrast, tree based schemes allow readers to issue binary prefixes for tags to match their IDs with. Previous works have shown that tree-based schemes provide more stable identification performance but incur more reader-tag interactions [21]. Existing MAC-layer tree-based works [13, 14, 16, 19, 21] deliver their optimal performances only when the distribution of tag IDs are highly uniform, which does not often happen in practical scenarios.

In order to improve the performance of tree-based identification, we explore how to infer local tag distribution from physical layer signals. When multiple tags collide, we can detect whether all responding tags reply with the same bit at each bit index by combining prior knowledge of tag coding scheme and physical layer patterns. For example, if the query prefix is "0" and the all-0 state is detected at index 3, the reader can infer that no tag has replied with the prefix pattern "0*1" (where * represents any bit value) and skip querying prefixes matching the pattern "0*1". In the binary query tree, this is equivalent to pruning the right branch of the layer-2 nodes rooted at "0". Since this physical layer information tells whether left or right branches of certain tree nodes is empty and thus could be skipped safely, we call such physical layer information "horizontal information".

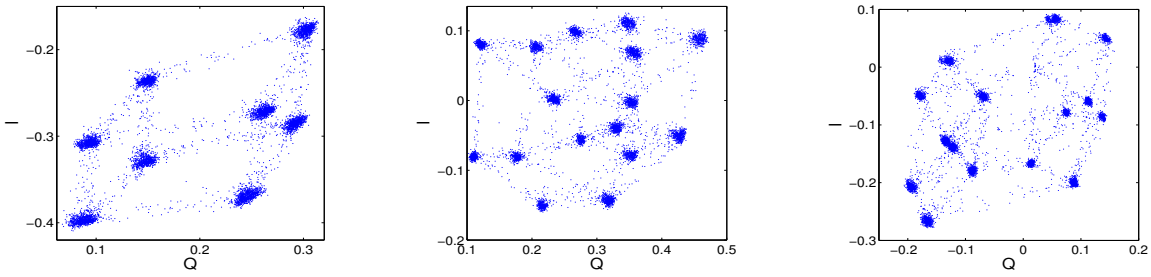
In addition, based on collision patterns in the physical layer, we can roughly infer the number of responding tags in each query. For example, if 4 tags respond when the reader queries prefix 01, instead of appending only one bit to prefix 01, the reader can directly append two bits to the prefix and query 4 new prefixes (0100, 0101, 0110, 0111) to directly resolve the collision. In the binary query tree, the reader skips two children nodes (i.e, 010 and 011) and directly jumps down to the 4 grandchildren nodes. We call such physical layer information "vertical information". We find that by leveraging both types of physical layer information, it is possible to skip many unnecessary queries during the overall identification process.

In this paper, we propose *PHY-Tree*, a novel tree-based scheme that extracts two types of distribution information (i.e., horizontal and vertical information) from every encountered colliding signal and utilizes them to guide a more efficient query over the binary tree. First, by looking into inherent temporal features of the tag modulation scheme, we can detect whether concurrent colliding tags backscatter with the same bit content at each bit index (horizontal info). Second, we can also estimate the number of colliding tags in a slot without extra communication overhead, by computing a related metric that is defined over the signal's constellation map (vertical info). By accumulating these information along the whole identification process, a great amount of information regarding the tag ID distribution can be obtained to avoid many unnecessary queries – this is the reason why *PHY-Tree* outperforms them by nature. We also design a mechanism to compensate for errors in the obtained physical layer information and ensure that all tag IDs in the set are correctly identified. Finally, we conduct experiments on our USRP/WISP testbed and perform extensive trace-driven simulations to evaluate *PHY-Tree*. The evaluation results show that *PHY-Tree* outperforms state-of-the-art tree-based identification scheme by 1.79×.

II. Background & Motivation

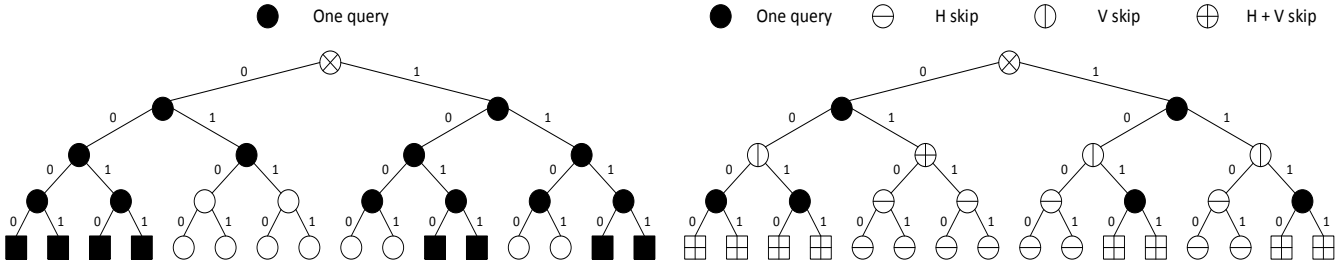
A. RFID backscatter

Passive tags are of small size and battery-free. They are generally used in UHF RFID systems that operate in the range from 860MHz to 960MHz. RFID reader issues continuous wave (CW) onto RFID tags and tags can transmit data by either reflecting or absorbing CW. In other words, each tag



(a) 8 clusters for 3 colliding tags. (b) 16 clusters for 4 colliding tags. (c) 15 clusters for 4 colliding tags.

Fig. 1. Ideally, 2^k number of clusters appear in the constellation map when k tags collide as shown in (a) for $k = 3$ and (b) for $k = 4$. However, in practice clusters may overlap and less than 2^k clusters could be observed as shown in (c) for $k = 4$.



(a) Query Tree takes 20 queryies to identify all 8 tag IDs. (b) *PHY-Tree* takes only 6 queries to identify all 8 tag IDs.

Fig. 2. Illustration on how the proposed *PHY-Tree* scheme accelerate identification performance compared to classic Query Tree scheme.

has two transmission states, HIGH (bit 1) and LOW (bit 0) [10, 15]. To increase the robustness against channel degradation and maintain a high decoding rate on the reader side, certain encoding scheme is applied to the data transmitted by a tag. EPC Class 1 Gen 2 (EPC-C1G2) standard [1] provides several encoding schemes including FM0 and Miller-based coding.

When k tags collide in the same time slot, two states of each tag combine linearly in the wireless channel. In principle, there are 2^k states in the mixed signal (in complex values). Among 2^k states, two of them are worth noting: all-0 and all-1 state. The two states occur when the k tags transmit bit 0 or bit 1 at the same time. We denote the two states as resonant states, to distinguish them from mixed states.

If we plot the received complex signal in 2D plane, we observe many clusters in the constellation map. For each cluster, its centroid represents a combined state of k tags and its radius indicates the power of noise from the wireless channel. Ideally we can observe 2^k clusters (as shown in Figure 1(a) and Figure 1(b)). When k increases, it becomes more challenging to accurately infer k from the number of clusters, as shown Figure 1(c) where 4 tags respond simultaneously.

B. RFID identification

The task of RFID identification is to collect all IDs of an unknown tag set. ALOHA based schemes [17, 30] require tags to reply randomly in any time slot of a frame and hence suffer from repetitive collisions. On the other hand, in tree based schemes the reader issues a series of binary prefixes and only those tags whose IDs match the issued prefixes would reply. Consequently, collisions are gradually reduced during such reader-tag interactions, and finally all tags can be identified.

We explain some concepts in tree-based schemes and illustrate the key intuition of our design in Figure 2(a). Each node represents a binary prefix in the tree and is classified into one of the following three types: collision, singleton and empty, depending on the number of responses upon queried. We find that 8 tags are in the bottom layer of the tree (in square shape), whose IDs can be read by traversing from the root to the tags. For instance, the ID of the 3^{rd} tag from left is 0010.

Many tree-based schemes [13, 16, 21] are proposed to reduce the number of query prefixes (represented as black nodes in Figure 2). Although their ways of traversing through the tree differ from each other, one common feature in these works is that they append at most one bit to the current queried prefix to resolve collisions.

C. Motivation

Despite the plenty amount of tree-based identification works, they perform poorly in practical scenarios where tag ID distribution is random. Unlike previous works, we do not assume the prior knowledge of tag ID distribution. To optimize the query, we estimate the local distribution of tags in each query subtree by extracting two types of physical layer information, namely horizontal and vertical information. We note that physical layer hints are obtained directly from the physical layer without extra communication overhead. As the hints are accumulated along the identification process, we could gradually refine the queries and ultimately converge to exact position of each tag ID.

We use an illustrative example to explain how horizontal and vertical information could be utilized in tree-based identification. In Figure 2(b) each of 8 tag IDs is labeled with a square in the bottom layer. We start querying with the two nodes in

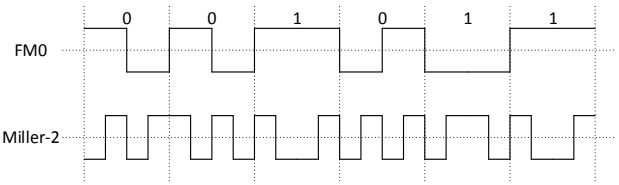


Fig. 3. FM0 and Miller-2 coding of a common binary sequence 001011.

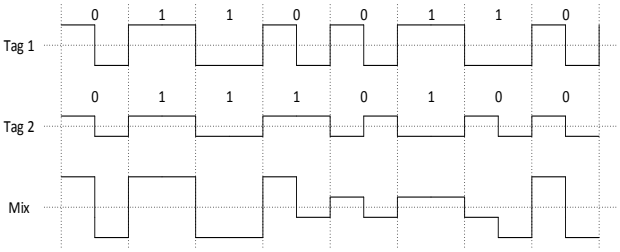


Fig. 4. Colliding signal of two tags with different IDs using FM0 coding.

the first layer: "0" and "1". When "0" is queried and the 4 tags in the subtree reply, the constellation map may exhibit as shown in Figure 2(b). In this case, the reader can leverage such collision hints and skip one query node vertically. As such, two nodes (i.e., 00 and 01) are skipped and marked as "V skip" (vertical skip). Meanwhile, the reader also obtains the horizontal information that no tags reply with prefix 01 since the 4 responding tags have the same bit 0 at index 2. Consequently, the node 01 and its descendants are skipped for query and marked as "H skip" (horizontal skip). It is noted that the skipped node 01 is the result of both skip types and marked as "H+V skip". Similarly, when "1" is queried, the reader obtains both horizontal and vertical information, which respectively indicate that 4 tags collide and no tags reply with the prefix pattern "1*0" (where * represent a bit value). Thus, the reader further skips the nodes 10 and 11 and two subtrees rooted from node 100 and 110.

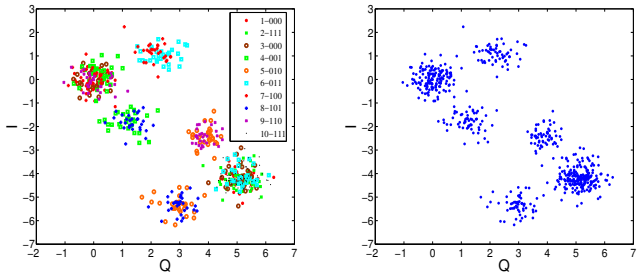
Next the reader proceeds with the unskipped nodes in the third layer (i.e., 000, 001, 101, 111). Upon node 000 is queried, the reader learns that only 2 tags collide and a mixed state appears at bit index 4. For this combination of horizontal and vertical information, the only possibility is that 0000 and 0001 coexist. Thus the reader can directly identify two tag IDs without explicit queries of 0000 and 0001.

Compared with Figure 2(a) where QT takes 20 queries (marked as black solid circles or squares) to identify all 8 tag IDs, *PHY-Tree* takes only 6 queries thanks to "H skip" and "V skip". To conclude, *PHY-Tree* improves the performance over existing MAC layer schemes as it obtains extra physical layer hints on the local ID distribution in each tag response slot and accumulates them along the whole identification process.

III. Horizontal Information

A. Horizontal Information

In Figure 3, FM0 flips states at the bit boundary between two neighboring bits. An extra state flipping occurs in the mid of bit 0, while the state of bit 1 keeps unchanged. We can thus infer that if bit 0 is sent, the latter half of this bit and of its previous bit should have the same state.



(a) Half-bit cluster. (b) Overall cluster.

Fig. 5. Constellation map of 3 tags. Their 10-bit IDs include all 8 bit states.

The above patterns can be extended to the mix signal of multiple tags, as shown in Figure 4. Specifically, no state flipping occurs in the mid of a bit in all-1 state – the first and second half of bit 1 have the same state. We see that when both tags send bit 1 (i.e., index 2, 3, and 6), no state flips happens in the corresponding indexes in collision signals. For a bit of all-0 state, the latter half of this bit and of its previous bit are the same. We see that when both tags send bit 0 (e.g., index 5 and 8), the latter states remain the same in the corresponding indexes in collision signals. Mixed states do not exhibit these features and can be distinguished from resonant states.

Next we check if resonant states can also be observed in Miller coding. Figure 3 shows an example of Miller-2 coding. We observe that for bit 1, the second and third quarters of the signal have the same state. For bit 0, we observe that the states of the first and third quarter of bit 0 are the same. These two patterns can be easily generalized for Miller-M coding with $M = 4, 8$. In the rest of this paper, we focus on FM0 coding.

By detecting resonant states from the collision signal, the reader can avoid querying unnecessary prefixes. We note that even if a resonant state is detected at an index not right after the prefix, we still can omit some unnecessary queries. For example, in Figure 2(b), if prefix "1" is queried and an all-1 state is detected at index 3, the reader can skip the nodes matching the prefix pattern "1*0"; in this case, the nodes 100 and 110 and their descendant nodes can be skipped.

B. Robust Detection Algorithm

To detect horizon information, we need to compare whether two half-bit states are the same so as to detect both resonant states. To perform the comparison in the complex tag response signal, we can find the corresponding complex values for the two states and compare their real and imaginary parts. However, the backscatter signals may suffer from noises which makes obtaining accurate horizon information more challenging.

Next we design a robust resonant state detection algorithm. In physical layer constellation map, since each cluster represents a combination state from all responding tags, we can infer if states in two half bits are the same by judging whether their corresponding clusters are overlapping with each other.

Figure 5(a) shows an example of overlapping clusters when 3 tags reply simultaneously. Each labeled cluster corresponds to samples in a half bit. The 3 tags transmit the following 10-bit sequences: 0100001111, 0100110011 and 0101010101,

which include all 8 possible states, i.e., from 000 to 111. At the bottom-right corner of Figure 5(a), the latter half of state 000 at index 3 (in brown circle) shares a large overlapping region as the latter half of state 111 at index 2 (in green solid square). In fact, we can distinguish between the first and second half of a bit from the tag response signal in the time domain. Moreover, both the frontier and latter half-bit clusters of state 111 at index 1 (in green solid square) and index 10 (in black dot) are almost overlapping as well. In contrast, states other than 000 or 111 do not have similar overlapping patterns. Figure 5(b) shows the overall constellation map.

To quantify the notion of cluster overlapping, we first compute the centroid of two clusters as the average of corresponding samples. Two clusters are judged as overlapping if the distance between their centroids is below a threshold:

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \leq c \times r_0, \quad (1)$$

where (x_1, y_1), (x_2, y_2) are the coordinates of two centroids and r_0 is the cluster radius. c is tunable and empirically set to 2.5 in Section VII. We can approximate r_0 as the L2-norm of all samples in a cluster:

$$r_0 = \sqrt{\sum_{i=1}^{q} (x_i - \bar{x})^2 + (y_i - \bar{y})^2}. \quad (2)$$

IV. Vertical Information

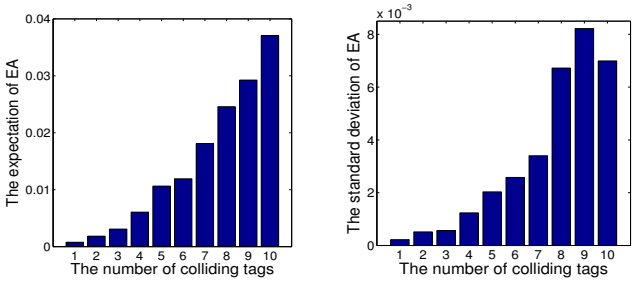
The number of replying tags k in a slot can help the reader to adjust the prefixes in the following queries. For example, if 4 tags respond to the prefix "0", 2 bits should be appended for the next query. In ALOHA schemes [30]: the optimum slot efficiency is achieved when the number of slots equals to the number of tags. Similarly, to accommodate k tags in the subtree, we need to append $\log_2 k$ bits to the previous query.

Many cardinality estimation techniques [6, 18, 20, 32] can estimate k with high accuracy but they involve extra query overhead. Other works [2, 10, 12] estimate k by counting the number of visible clusters in the constellation map of the tag response signal. Although these works are lightweight, they are not scalable to large k values. This is because when k becomes large, individual clusters have the tendency to overlap with each other and consequently, cluster counting based method no longer performs well. As a result, it is hard to accurately count the exact number of responding tags when the number of tags increases. Instead of aiming at the exact k, we find that it suffices for *PHY-Tree* to estimate the logarithmic scale of k to make an intelligent bit appending decision in subsequent queries. To this end, we explore a new metric defined over the signals constellation map to estimate the scale of k.

A. Intuition & Definition

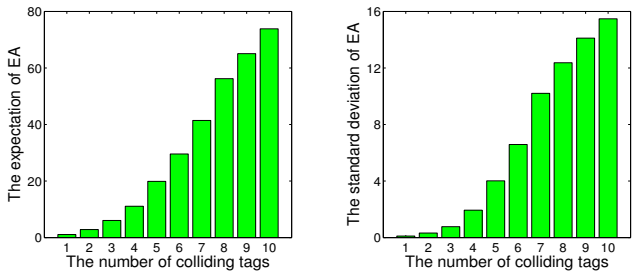
The area occupied by the physical layer symbols is a good candidate of k-indicator because: 1) it might increase when k increases; 2) it is robust to overlapping clusters for large k.

Based on the above observation, we define an indicator of k, named "effective area" (EA), as follows. First we find the smallest rectangular region that contains all data samples in the constellation map and divide it into small square grids.



(a) Expectation. (b) Standard deviation.

Fig. 6. Experiment results.



(a) Expectation. (b) Standard deviation.

Fig. 7. Simulation results.

Next we count the number of samples in each grid and set a threshold to differentiate noise grids from signal grids – the grids with the number of samples below the threshold are regarded as noise. EA is computed as the multiplication of the number of signal grids and the area of the grid unit and can be explained as the summation of areas of all signal grids.

B. Basic Observations from Experiment Results

We conduct experiments on USRP/WISP testbed (shown in Figure 8) to validate whether EA is a good indicator of k. We configure multiple WISP tags to reply concurrently with random number (RN) sequences to single reader queries. As it is hard for us to obtain traces of larger k from our testbed, we generate them by adding up available experimental traces instead. For example, we can synthesize a trace of $k = 6$ by adding up two traces of $k = 2$ and of $k = 4$.

Figure 6(a) plots EA versus k based on the collected and synthesized traces. To compute EA, we empirically choose the grid size to be 0.01×0.01 and the grid density threshold to be 13% of total number of samples in a response slot. For each k, we compute EA values for corresponding traces and display an average EA in Figure 6(a). We see that EA increases exponentially over k. We also plot the standard deviation of EA versus k in Figure 6(b). We find though the standard deviation of EA also increases when k increases, it is small compared to the corresponding mean value of EA, meaning that EA is a stable indicator of k across diverse traces.

To confirm whether the tendency between EA and k from Figure 6 is general or specific to certain configurations, we simulate k-tag baseband response signals ($k = 1, 2, ..., 10$) with the SNR profile of collected experimental traces, which often falls in the range [10dB, 30dB]. We set the AWGN noise power N_0 to 0.1. Each tag randomly picks up an initial signal

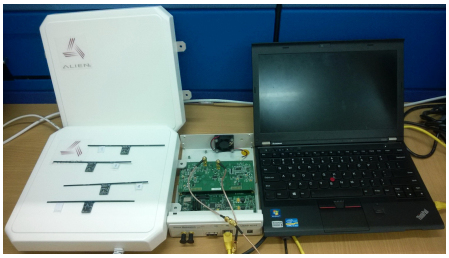


Fig. 8. Our USRP-N210 and WISP 4.1 testbed.

amplitude in [1, 10] and phase in [0, 2π]. Since the power scale of simulated traces is much larger than that of experimental ones, we adjust the grid size and the density threshold to better distinguish between signal grids and noise grids and hence compute a more accurate EA. Empirically, we adjust the grid size to be 0.50 $\times$ 0.50 and leave the grid density threshold the same as in Figure 6. We plot the average EA versus k for simulated traces in Figure 7, from which we observe a similar tendency as in Figure 6.

C. Appending Rule

The number of observed clusters in the constellation map is always upper bounded by the number of data samples N_s in a response slot. Each of our experimental traces contains 6000 to 8000 samples, which allow us to estimate k up to 12.

In fact, it makes no big difference to infer 8 or more than 8 (but less than 16) colliding replies if k is used for tree-based identification. Specifically, we need to append $\lceil \log_2 k \rceil$ bits to the current prefix for further queries, if the length of the prefix after appending does not exceed the tag ID length. Following this rule, 4 bits will be appended if k is detected to be in the range $[9, 16]$. Hence, if we can infer 8 tags at most and the corresponding EA is denoted as a_0, we would append 4 bits for any measured a satisfying $a > a_0$.

One may argue that more dedicated rules can be applied to determine the number of appended bits. For $k = 9$, appending 3 bits may resolve most collisions but appending 4 bits may incur some inefficient queries to which no tag replies. Though it seems that appending 3 bits is more optimal than appending 4 bits, it does not fundamentally improve the identification performance, due to two reasons. First, our mapping model between EA and k is a coarse one. Second, in general tree-based schemes, a rough estimation of k for each reader query without incurring extra overhead is sufficient to bring promising performance gain for the whole query process.

D. Modelling EA over k

We propose a non-parametric model for the mapping between EA and k. Since inferring k up to 8 suffices (according to Section IV-C), we adopt a range based model for each $k \in [1, 8]$. If the measured EA is in a certain range, the associated k can be obtained. To train the range-based model 8 tags are chosen from the tag set and multiple traces are collected for each k. Then the expectation and standard deviation of EA are computed; the minimum/maximum of the range of EA for k are obtained by subtracting/adding the standard deviation from/to the expectation. Such a model is robust because the range estimation errors for k_1 are not easily propagated to that for k_2, for $k_1 \neq k_2$.

V. *PHY-Tree* Protocol

We propose an efficient and robust RFID identification scheme by exploiting both horizontal and vertical information.

A. A Basic Protocol

Suppose a target set contains M tags, each has an ID with the length of L bits. We denote the ID of the tag j as $ID_j = b_{L-1}b_{L-2}...b_1b_0$, where b_{L-1} is MSB and b_0 is LSB. When the reader queries with an l-bit prefix $Q = q_{l-1}q_{l-2}...q_1q_0$ ($1 \leq l \leq L$), tag j checks if the first l bits of its ID, $b_{L-1}b_{L-2}...b_{L-l}$, match Q. If they match, the tag j replies to the reader with its ID.

We first design a basic identification algorithm. The reader starts the query from the first layer of the binary tree (i.e., $Q = 0$ or $Q = 1$). Upon receiving the colliding physical layer signal from k tags for the prefix Q, the reader can compute an estimation of the number of replying tags $\hat{k}$ (vertical information) and obtain an inference vector $G = g_{L-1}g_{L-2}...g_1g_0$ (horizontal information), where g_i ($i = 0, 1, ..., L-1$) indicates the bit state at index i. Specifically, g_i is set to 0 or 1 if an all-0 or all-1 state is detected respectively and -1 otherwise.

Since only tags matching Q respond, we know that g_{L-1} $g_{L-2}...g_{L-l} = q_{l-1}q_{l-2}...q_1q_0$. If collision happens, the reader resolves the collision as follows. First it uses $\hat{k}$ and the appending rules to derive the optimal number of appended bits h. The reader appends h bits to the current queried Q to obtain the appended prefix query set $\mathcal{Q}'$ for subsequent queries. For example, if h is 3, $\mathcal{Q}'$ includes prefixes from $q_{l-1}q_{l-2}...q_1q_0000$ to $q_{l-1}q_{l-2}...q_1q_0111$. Second the reader uses G to filter out unnecessary prefixes from $\mathcal{Q}'$. If $g_{L-l-3} = 1$ in the above example, $\mathcal{Q}'$ is reduced to only 4 prefixes: $q_{l-1}q_{l-2}...q_1q_0001$, $q_{l-1}q_{l-2}...q_1q_0011$, $q_{l-1}q_{l-2}...q_1q_0101$ and $q_{l-1}q_{l-2}...q_1q_0111$.

Generally h is solely determined as $\lceil log_2\hat{k} \rceil$. However, if subsequent d bits are detected as resonant states right after the prefix Q, i.e., $g_{L-l-1}...g_{L-l-d}$ takes valid values (0 or 1), h is adjusted to be $\max(\lceil log_2\hat{k} \rceil, d+1)$.

We describe the whole query process as follows. In the binary tree representation, the reader queries in the breadth-first mode. Denote $\mathcal{N}$ as the queue of subsequent prefixes for query. $\mathcal{N}$ is initialized with 2 prefixes "0" and "1" and is automatically sorted in increasing prefix length. In the case that two prefixes in $\mathcal{N}$ have the same length, the one with smaller value is queried first. For example, prefix 010 is queried before prefix 011. The reader first obtains the minimum prefix length l_{min} in $\mathcal{N}$ and queries all prefixes with length l_{min} in $\mathcal{N}$. Each time the reader pops out the first prefix in $\mathcal{N}$ for query and waits for tag responses. If less than 3 tags reply, the reader continues to pop out a new Q in $\mathcal{N}$ for query. Specifically if one or two tags reply to Q, they can be uniquely identified. Otherwise, the reader extracts G, $\hat{k}$ and d from the colliding signal to compute h. The reader appends h bits to Q to get the initial $\mathcal{Q}'$, filters out some prefixes in $\mathcal{Q}'$

using G and adds the updated $\mathcal{Q}'$ to $\mathcal{N}$. The reader repeats the above query procedure until $\mathcal{N}$ becomes empty.

B. Error Compensation

The bit state detection algorithm in Section III may mistake mixed states for resonant states and guide the reader to ignore some subtree nodes of the current prefix. Therefore, some tags in the set are missed by the reader. We need to eliminate this problem to meet the basic requirement of RFID identification.

A straightforward solution is to record the list of unqueried prefixes due to the guidance of G for each query in the basic algorithm because the missed tags may reply to these prefixes. The reader may adopts the original QT scheme [13] to query prefixes in this list and identifies the remaining tags, without using either horizontal or vertical information. Specifically, it uses depth-first query method to traverse the whole binary tree and always appends one bit for the next queries if collision is encountered. Although it ensures no tag is missed, this solution counters against the benefit brought by horizontal information and degrades the performance gain.

Here we propose a more efficient compensation scheme that could reserve the benefit of horizontal information. The simulation results in Section VII-A show that the accuracy of detecting resonant states is high under tight replying tag synchronization. This indicates that a majority of tags in the target set are identified using the basic algorithm. This motivates us to design the following error compensation scheme. After the basic algorithm stops, we mute those tags which have been already identified so that they will not interfere with the second round, where the reader performs a fresh identification using QT, starting from the first layer. Since only a few unidentified tags are left in the second round, it does not take many reader queries to finish the second round, indicating its efficiency.

C. Further Improvement in Efficiency

By combining horizontal and vertical information in another perspective, we can identify 2 colliding tags without further queries and hence gain extra identification ability. Basically if we can estimate $\hat{k} = 2$ for one query with high accuracy, we can uniquely identify each of two tags with its partial ID, rather than decode the whole ID. We explain the reason as follows. Since each tag owns a unique ID, IDs of the two tags must take different bit values in at least one bit index (e.g., i). In other words, a mixed bit state exists at index i. Our bit state detection algorithm can find out bits with mixed states and its accuracy is quite high when only two tags reply (reported in Section VII-A). The partial IDs for two tags are thus set as the queried prefix concatenated with bit 0 and 1 at index i.

We incorporate this feature in the tree-based algorithm by slightly modifying the appending rule used in Section V-A:

$$h = \max(\lceil log_2\hat{k} \rceil - 1, d + 1). \quad (3)$$

The term $\lceil log_2\hat{k} \rceil - 1$ in Eq.3 reflects the change brought by extra identification ability of 2 tags. It is noted that we cannot extend the partial ID identification to the case in which more than 2 tags reply, due to two reasons. First, the estimation error in the EA-k model increases when k increases, leading to the inaccurate estimation results for $k > 2$. Second, even if k is correctly estimated, we cannot identify these tags definitely as more than k possible state combinations exist in several bits.

VI. Discussion on Synchronization

When we illustrate two types of physical layer information in Section III and IV, we assume that the responding tags in a slot are tightly synchronized. In real world scenarios, tags have large diversities in their response delays due to multiple factors such as manufacturers, types, antenna orientations, etc. In this section, we briefly discuss how imperfect synchronization affects the two types of physical layer information and suggest how to mitigate the impact.

On one hand, the imperfect synchronization makes negligible impact on the vertical information. We estimate the number of concurrent replies by processing samples in the entire time slot, rather than samples in individual bit durations. Thus the vertical information does not rely on the internal timings between contiguous bit durations in the response signal.

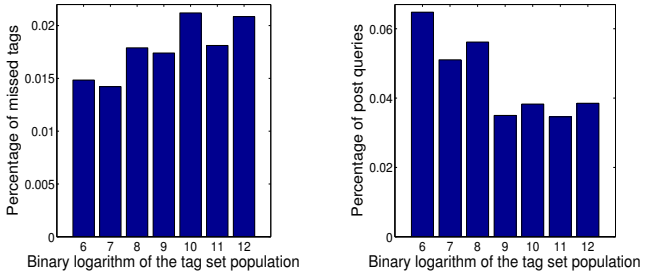
On the other hand, imperfect synchronization degrades the quality of the horizontal information. Specifically, discovering all-0 and all-1 states in a bit requires bit-level synchronization. Without perfect synchronization, we may accidentally detect a resonant state as a mixed state (false negative), which leads to missed opportunities for saving some reader queries, or detect a mixed state as a resonant state (false positive), which results in some tag IDs being not identified. Unlike Buzz [24], which can calibrate for each tag's clock to adjust its starting time offset so that responding tags achieve tight synchronization, we aim to identify all IDs of a completely unknown tag set. Thus, we should not devote any effort to calibration; otherwise such an effort can afford us to collect all tag IDs one by one.

We suggest several approaches to mitigate the impact of imperfect synchronization on our scheme. Each tag can use lower frequency for backscattering its ID to reduce the unsynchronization rate, which is the ratio of the maximum starting time offset over the individual bit duration. For example, if 4 symbols are used to encode one bit, the unsynchronization rate can be reduced by half compared to using FM0 coding. Better circuits are expected to be designed for RFID tags to further reduce their unsynchronization rates [7, 8]. Besides, to completely eliminate false positives for the horizontal information, we design an error compensation mechanism in Section V-B. Additionally, powerful collision recovery schemes like BiGroup [15] (which can decode 4 to 5 collision tags without assuming tight synchronization) can augment our scheme to further improve the overall identification performance.

VII. Evaluation

A. Microbenchmark

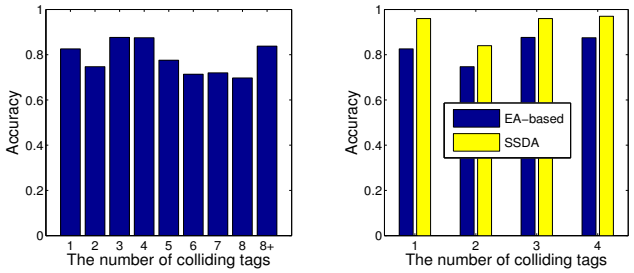
In this section, we evaluate the effectiveness of horizontal and vertical information. We collect practical traces from our USRP/WISP testbed in the office environment, as shown in Figure 8. The USRP motherboard is equipped with an RFX900 daughterboard operating at 900Hz UHF band to transmit reader commands and receive backscatter signals from tags.



(a) Percentage of missing tags in basic query round.

(b) Percentage of NoQ in the error compensation round.

Fig. 9. Impact of false positive of horizontal information on *PHY-Tree*.



(a) Inference accuracy of the EA model for wide-range k.

(b) Comparison with SSDA for small-range k .

Fig. 11. Accuracy comparison between EA-k model and SSDA[10].

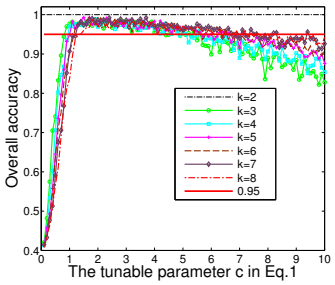


Fig. 10. Optimal range for c in resonant state detection with different ks.

The laptop connects to the USRP device and performs offline processing on the backscatter signal in a single slot. The sampling rate at the receiving port is 4 million symbols per second. Each physical layer symbol is represented by a complex value containing both quadrature and in-phase components. We configure WISP tags to reply their random number (RN) sequences concurrently to single reader queries. We collect 258 traces in various operation environments.

1) Horizontal Information: We first study the performance of the resonant state detection in obtaining horizontal information. Three important performance metrics are overall accuracy, false positive rate (FPR) and false negative rate (FNR). The overall accuracy refers to the ratio of the number of correct detection of both resonant and mixed states over the total number of bit states. False positive rate refers to the ratio of the number of detecting mixed state as resonant state over the number of actual mixed states. False negative rate refers to the ratio of the number of detecting resonant state as mixed state over the number of actual resonant states. Unless otherwise noted, we repeat an algorithm for 100 runs to obtain its average performance for each setting.

We find the optimal c, an important tunable parameter in Eq.1. We vary k from 2 to 8 and design a set of 10-bit tag IDs (which is enough to accommodate 8 different IDs) for each k. With random SNR and fixed noise power of each generated trace, we compute an average overall accuracy for each k.

We plot the overall accuracy over c in the range $[0.1, 10]$ step by 0.1 in Figure 10. From this figure, we discover that for each k, the overall accuracy first increases sharply with c, maintains high in a certain range and decreases slowly with c in the end. We highlight the 0.95-accuracy line in this figure and find an optimal range $[1.39,\ 4.07]$ of c for all ks. For $k = 2$, the overall accuracy is always 100% for any c, demonstrating the credibility of the partial ID identification of 2 colliding tags.

After obtaining the optimal c, we study the accuracy of the resonant state detection with different k. We set tag ID length to 100 to better accommodate more unique tag IDs and each of the all-0, all-1 and mixed state occupies 33.3% of the total 100 bits. We choose 6 values for c from the optimal range (1.5, 2, 2.5, 3, 3.5 and 4). A slight difference from the setting in Figure 10 is that we generate random set of tag IDs in each run given a value of k. We plot the overall accuracy, FPR and FNR over k from 3 to 100 in Figure 12.

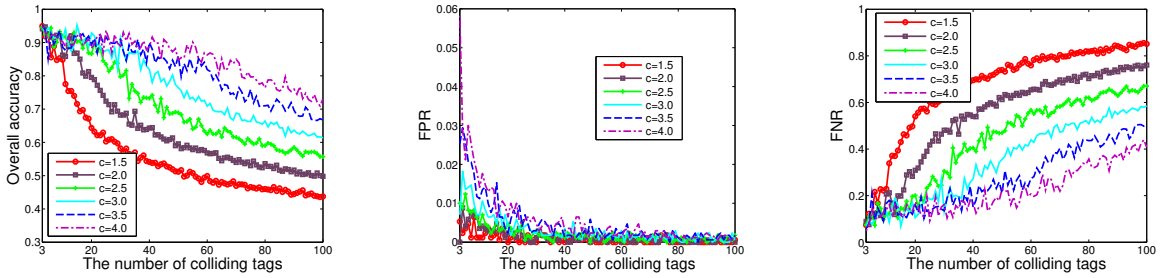
In Figure 12(a), the trend is that the overall detection accuracy of both resonant and mixed states decreases when k increases. This is natural as collisions from more tags impose more challenges on the detection. Nevertheless, the algorithm achieves around 70% accuracy when 20 tags collide for $c = 1.5$. From Figure 12(b), we find FPR increases when c increases. We emphasize in Section V-B that high FPR of horizontal information is not desired and hence should be as low as possible. As such, we can trade off between the overall accuracy and FPR; if we set c to 2.0, we can obtain $\leq 1\%$ FPR and achieves $\approx 80\%$ overall accuracy when k is 20. In Section VII-B when we compare *PHY-Tree* with existing tree-based identification schemes, we fix c to 2 in *PHY-Tree*. In Figure 12(c), we can see that FNR increases over k and decreases over c, which matches the trend in Figure 12(a).

Any non-zero false positive rate would trigger our error compensation scheme. If the number of missing tags after the basic query round is large, the follow-up error compensation round incurs large query overhead and renders our proposed *PHY-Tree* inefficient. We thus need to study about such overhead to better evaluate our scheme.

Two metrics are considered: the percentage of missed tags over the tag set population M in the basic query round and the percentage of queries in the error compensation round over the total number of queries. We evaluate the *PHY-Tree* algorithm on M tags with randomly generated unique 15-bit IDs, where M varies from 2^6 to 2^{12}. Figure 9 plots the result.

In Figure 9(a), we find that for all M values, the average percentage of missed tags is $\leq 2.2\%$. In Figure 9(b), we find the percentage of queries in the compensation round is $\leq 4\%$ when $M > 8$. Both figures demonstrate the small overhead and high efficiency of our error compensation scheme.

2) Vertical Information: Next we study the accuracy of obtained vertical information. As discussed before, we only need to infer up to 8 tag replies with reasonable accuracy and



(a) Overall accuracy. (b) False positive rate. (c) False negative rate.

Fig. 12. Performance of the resonant state detection algorithm in obtaining horizontal information when k varies in a wide range and c is fixed to be optimal.

for $k > 8$, we only need to judge if $\hat{k} > 8$. We adopt the non parametric model and train the range for each $k \in [1, 8]$. The training set contains 100 randomly generated traces and the testing set contains another 100 newly generated traces, for each k. Each tag replies with 100-bit RN sequence. In both the training and testing sets, we create 9-tag collision traces for $k > 8$, which is the most challenging scenario for our EA-k model since EA increases over k. We compare the estimation $\hat{k}$ with the ground truth k and compute the ratio of the number of correctly inferred traces over the overall number of traces as the accuracy. Figure 11(a) plots the result. It is observed from Figure 11(a) that our EA model achiev-es $\approx 80\%$ accuracy on average. The corresponding accuracy is somehow low for certain k value like 2, 6, 8. This is because of overlapping ranges incurred in the training test: the upper bound of the range for k is larger than the lower bound of the range for $k + 1$. To alleviate this issue, we can set the middle of the overlapping range as the common value for the upper bound of k and lower bound of $k + 1$.

In Figure 11(b), we compare the accuracy between EA-k model and the SSDA algorithm in [10], which can estimate up to 4 colliding tags. We implement SSDA using the original settings in [10] and set the k range to $[1, 4]$ for fair comparison. We observe from Figure 11(b) that SSDA achieves higher accuracy ($\approx 90\%$) than our EA-k model does. Actually we can combine both methods for better accuracy. Specifically, we can use both methods to judge whether both estimation results for k are below 4. If it is so, we apply SSDA for better accuracy; otherwise we apply EA model for better scalability.

B. Comparison with Existing Identification Schemes

In this subsection, we compare *PHY-Tree* with three existing tree-based identification works: QT [13], STT [16] and TH [21]. Similar to [21], we adopt the average number of queries (NoQ) as the performance metric, which is defined as the ratio of total number of reader queries over tag set population M. We also consider three types of tag ID distribution, namely uniform, block and random (described in Section II-C). In block distribution we first set the block size b to 10. Tag ID is of 15 bits and M varies from 2 to 2^{12}. We plot the comparison results under three ID distributions in Figure 13.

Our observations in Figure 13 are as follows. First, *PHY-Tree* consumes the least amount of queries among all schemes under three ID distributions. In the random ID distribution, which represents the most typical case in real world scenarios, the average number of queries for *PHY-Tree* is around 1 and thus reduces the number of queries of TH (which is the most efficient scheme among existing ones) by $1.79\times$. This gain is quite promising: for any tag set with large cardinality, our *PHY-Tree* is $1.79\times$ more time-efficient than TH, the tree hopping algorithm with accurate cardinality as the input. Second, existing schemes take more queries to handle non uniform distributions, especially the random distribution. Our *PHY-Tree* can always utilize partial physical layer information to enhance identification performance and outperform all these works in any kind of tag ID distributions.

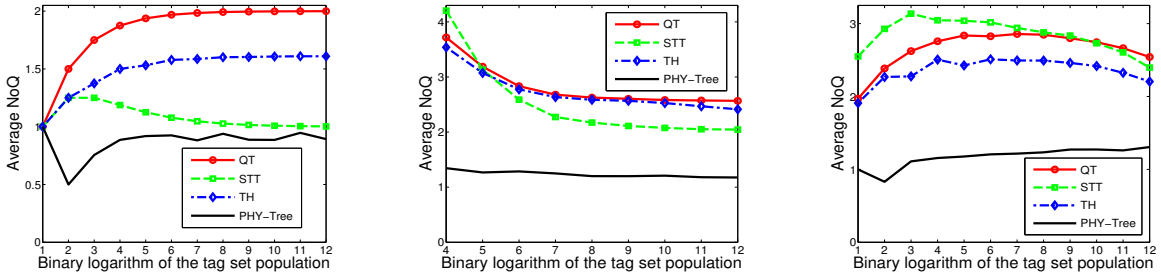
Among three types of tag ID distribution, block distribution lies between in terms of ID continuity, depending on the block size b. Random and uniform distribution can be regarded as a block distribution with $b = 1$ and $b \approx M$ respectively.

VIII. Related Work

Many research works suggest fast RFID identification schemes. In the ALOHA [30] scheme, tags reply in random slots and only singleton slots can be identified. Their performances maximize when the issued frame size is optimized. Tree-based schemes [4, 5, 13, 14, 16, 21] aim to reduce the number of collisions through reader-tag interactions. Hybrid approaches [17] combine both for identification purpose. Some other works [33–35] study the identification problems in dynamic RFID systems. Different from these works, we utilize physical layer information from tag replies to improve the identification efficiency.

Collision recovery methods aim to decode collisions of multiple tags. Buzz [24] assumes bit-level synchronization among responding tags and decodes the aggregated rateless codes of multiple tags. [2, 22] analyze the constellation map and separate 2 tags. BiGroup [15] decodes up to 5-tag reply with reasonable accuracy by extracting temporal-spatial features from the received signal. Laissez-Faire [11] detects signal edges and separates signal edges of multiple tags and thereby decodes tag collisions. Laissez-Faire, however, requires the tags to transmit with assigned initial offsets and bit durations. Unlike those works, we focus on improving the overall query process to identify all tags with minimum number of queries.

Existing works [10, 28] have explored how to utilize physical layer information to support other operations. PLACE [10] counts the number of clusters in contellation map to enhance RFID cardinality estimation. Tagoram [28] utilizes



(a) Uniform distribution. (b) Block distribution with $b = 10$. (c) Random distribution.

Fig. 13. Performance comparison between several schemes under three types of tag ID distribution.

phase information from RFID tags and tracks the labelled items.

IX. Conclusion

Traditional tree-based RFID identification methods are only MAC-layer solutions and perform poorly when the tag ID distribution is random. We propose a novel tree-based RFID identification scheme that utilizes physical layer information to improve the identification performance. Given the current queried tree node, *PHY-Tree* can skip its children nodes in either left or right branch by opportunistically detecting resonant states; meanwhile, it also estimates how many tags collide together and skips its children nodes in the same layers. By accumulating both information along the whole identification process, *PHY-Tree* significantly outperforms previous MAC layer tree-based schemes.

Acknowledgement

We acknowledge the support from the Hong Kong ECS under Grant PolyU 252053/15E and the Hong Kong PolyU under Grant G-YBMT.

References

[1] EPCglobal C1G2 UHF RFID protocol at 860MHz-960MHz, http://www.epcglobalinc.org/standards/uhfc1g2. July 2014.

[2] C. Angerer, R. Langwieser, and M. Rupp. RFID Reader Receivers for Physical Layer Collision Recovery. *IEEE Trans. on Communications*, 58(12):3526-3537, 2010.

[3] G. Avoine and P. Oechslin. A Scalable and Provably Secure Hash-Based RFID Protocol. In *IEEE PerCom Workshops*, 2005.

[4] N. Bhandari, A. Sahoo, and S. Iyer. Intelligent Query Tree (IQT) Protocol to Improve RFID Tag Read Efficiency. In *IEEE ICIT*, 2006.

[5] J. Capetanakis. Tree Algorithms for Packet Broadcast Channels. *IEEE Trans. on Information Theory*, 25(5):505-515, 1979.

[6] B. Chen, Z. Zhou, and H. Yu. Understanding RFID Counting Protocols. In *ACM MobiCom*, 2013.

[7] N. Cho, S. Song, S. Kim, S. Kim, and H. Yoo. A 5.1-μw UHF RFID Tag Chip Integrated with Sensors for Wireless Environmental Monitoring. In *IEEE ESSCIRC*, 2005.

[8] R. Greeff, F. Smith, and D. Ovard. RFID Device Time Synchronization. Patent US7889083, 2006.

[9] H. Hassanieh, J. Wang, D. Katabi, and T. Kohno. Securing RFIDs by Randomizing the Modulation and Channel. In *USENIX NSDI*, 2015.

[10] Y. Hou, J. Ou, Y. Zheng, and M. Li. PLACE: Physical Layer Cardinality Estimation for Large-Scale RFID Systems. In *IEEE INFOCOM*, 2015.

[11] P. Hu, P. Zhang, and D. Ganesan. Laissez-Faire: Fully Asymmetric Backscatter Communication. In *ACM SIGCOMM*, 2015.

[12] R. Khasgiwale, R. Adyanthaya, and D. Engels. Extracting Information from Tag Collisions. In *IEEE RFID*, 2009.

[13] C. Law, K. Lee, and K. Siu. Efficient Memoryless Protocol for Tag Identification. In *ACM DIALM Workshop*, 2000.

[14] V. Namboodiri and L. Gao. Energy-Aware Tag Anticollision Protocols for RFID Systems. In *IEEE PerCom*, 2007.

[15] J. Ou, M. Li, and Y. Zheng. Come and Be Served: Parallel Decoding for COTS RFID Tags. In *ACM MobiCom*, 2015.

[16] L. Pan and H. Wu. Smart Trend-Traversal: A Low Delay and Energy Tag Arbitration Protocol for Large RFID Systems. In *IEEE INFOCOM*, 2009.

[17] C. Qian, Y. Liu, H. Ngan, and L. Ni. ASAP: Scalable Identification and Counting for Contactless RFID Systems. In *IEEE ICDCS*, 2010.

[18] C. Qian, H. Ngan, and Y. Liu. Cardinality Estimation for Large-Scale RFID Systems. In *IEEE PerCom*, 2008.

[19] J. Ryu, H. Lee, Y. Seok, T. Kwon, and Y. Choi. A Hybrid Query Tree Protocol for Tag Collision Arbitration in RFID Systems. In *IEEE ICC*, 2007.

[20] M. Shahzad and A. Liu. Every Bit Counts - Fast and Scalable RFID Estimation. In *ACM MobiCom*, 2012.

[21] M. Shahzad and A. Liu. Probabilistic Optimal Tree Hopping for RFID Identification. In *ACM SIGMETRICS*, 2013.

[22] D. Shen, G. Woo, D. Reed, and A. Lippman. Separation of Multiple Passive RFID Signals Using Software Defined Radio. In *IEEE RFID*, 2009.

[23] J. Wang, F. Adib, R. Knepper, D. Katabi, and D. Rus. RF-Compass: Robot Object Manipulation using RFIDs. In *ACM MobiCom*, 2013.

[24] J. Wang, H. Hassanieh, D. Katabi, P. Indyk. Efficient and Reliable Low-Power BackScatter Networks. In *ACM SIGCOMM*, 2012.

[25] J. Wang and D. Katabi. Dude, Where's My Card? RFID Positioning That Works with Multipath and Non-Line of Sight. In *ACM SIGCOMM*, 2013.

[26] J. Wang, D. Vasisht, and D. Katabi. RF-IDraw: Virtual Touch Screen in the Air Using RF Signals. In *ACM SIGCOMM*, 2014.

[27] R. Want. An Introduction to RFID Technology. *IEEE Pervasive Computing*, 5(1):25-33, 2006.

[28] L. Yang, Y. Chen, X. Li, C. Xiao, M. Li, and Y. Liu. Tagoram: Real-Time Tracking of Mobile RFID Tags to High Precision Using COTS Devices. In *ACM MobiCom*, 2014.

[29] R. Zhang, Y. Liu, Y. Zhang, and J. Sun. Fast Identification of the Missing Tags in A Large RFID System. In *IEEE SECON*, 2011.

[30] B. Zhen, M. Kobayashi, and M. Shimizu. Framed ALOHA for Multiple RFID Objects Identification. *IEICE Transactions on Communications*, 88(3):991-999, 2005.

[31] Y. Zheng and M. Li. P-MTI: Physical-Layer Missing Tag Identification via Compressive Sensing. In *IEEE INFOCOM*, 2013.

[32] Y. Zheng and M. Li. ZOE: Fast Cardinality Estimation for Large-Scale RFID Systems. In *IEEE INFOCOM*, 2013.

[33] X. Liu, S. Zhang, K. Bu, and B. Xiao. Complete and Fast Unknown Tag Identification in Large RFID Systems". In *IEEE MASS*, 2012.

[34] X. Liu, B. Xiao, S. Zhang, and K. Bu. Unknown Tag Identification in Large RFID Systems: An Efficient and Complete Solution. *IEEE Trans. on Parallel and Distributed Systems*, 26(6):1775-1788, 2015.

[35] Q. Xiao, B. Xiao, and S. Chen. Differential Estimation in Dynamic RFID Systems. In *IEEE INFOCOM*, 2013.