Efficient Tag Identification in Mobile RFID Systems

Lei Xie[†‡], Bo Sheng[‡], Chiu C. Tan[‡], Hao Han[‡], Qun Li[‡], Daoxu Chen[†]

[†]State Key Laboratory of Novel Software Technology,
Department of Computer Science and Technology, Nanjing University, China
[‡]College of William and Mary, Williamsburg, VA, USA
Email: [†]xielei@dislab.nju.edu.cn, [‡]{shengbo, cct, hhan, liqun}@cs.wm.edu, [†]cdx@nju.edu.cn

Abstract—In this paper we consider how to efficiently identify tags on the moving conveyor. Considering conditions like the path loss and multi-path effect in realistic settings, we first propose a probabilistic model for RFID tag identification. Based on this model, we propose efficient solutions to identify moving RFID tags, according to the fixed-path mobility on the conveyor. A dynamic program based solution and an adaptive solution are proposed to select optimized frame sizes during the query cycles. Simulation results indicate that by leveraging the probabilistic model our solutions can achieve much better performance than using parameters for the ideal propagation situations.

I. INTRODUCTION

Radio frequency identification (RFID) technology is widely used in supply chain management as a means of monitoring physical goods and assets. For large warehouses with tens of thousands of physical goods entering and leaving each day, the process of collecting RFID tag IDs is highly automated through the use of conveyor belts. Once goods are placed onto a conveyor belt, they move along on a fixed path at a constant speed until they are read by RFID readers carefully placed along the conveyor belt's path. Given the crucial role of supply chains to the economy, improving the performance of reading RFID tags on a conveyor belt is an important component in improving the overall efficiency.

RFID tags are simple devices that are unable to self-regulate their radio transmissions to avoid collisions. An anti-collision protocol is used to regulate tag replies in order to obtain all the available tag IDs in the shortest amount of time. One of the most common standards for supply chain RFID tags is the EPC Class 1 Gen 2 RFID standards. This standard is based on a slotted ALOHA process to regulate tag responses. Prior research [1][2] on improving performance have focused on optimizing the parameters used in this slotted ALOHA model. However, these research only considered optimizing the parameters when reading *static* RFID tags under *ideal* environmental conditions. Real world environmental conditions such as path loss and multi-path effects, as well as the fixed-path mobility of RFID tags are not considered. As such, these results are less applicable to our problem. Other work by [3][4][5] performed practical experiments to determine the best parameters. However, it is unclear how well these findings can be generalized to other environmental settings.

In this paper, we consider the important problem of improving reading performance for RFID tags under a moving conveyor belt. We first bridge the gap between theoretical anti-collision protocols and real world conditions by introducing a probabilistic model for RFID tag identification. Our model offers a fundamental guidance for setting MAC layer parameters, such as the frame size for ALOHA based protocols. Based on our model, we propose protocols that improve reading performance for moving RFID tags. Our proposed solution reduces the reading time by 23% than using parameters derived under ideal conditions with no propagation loss[6]. This improvement is achieved while adhering to the existing EPC Class 1 Gen 2 RFID standards.

We make the following contributions in this paper.

1) We give a probabilistic model for the slotted ALOHA based anti-collision schemes, in order to understand the principles for RFID tag identification under the realistic settings. We analyze how the physical layer property affects the behavior of the MAC layer protocol.

2) We propose efficient solutions to identify moving tags on the conveyor according to the probabilistic model. We present a dynamic program based solution and an adaptive solution of selecting frame sizes for efficient tag identification.

3) To the best of our knowledge, this is the first theoretical work which investigates the realistic model for RFID tag identification by taking the probabilistic propagation into consideration. We give a fundamental illustration of how the current anti-collision protocol works in the realistic settings, and give indications for a closer coupling of the physical and MAC layers.

The rest of the paper is as follows. Sections II and III present the related work and RFID background respectively. We formulate our problem in Section IV, and present our probabilistic model in Section V. Section VI contains the protocols for efficient reading of mobile RFID tags. Our evaluation is in Section VII, and we conclude in Section VIII.

II. RELATED WORKS

RFID anti-collision protocols can be categorized into tree-based and ALOHA-based ones. Tree-based protocols resolve collisions by muting subsets of tags that are involved in a collision [7]. Pan et al. propose a Smart Trend-Traversal protocol [8], a Query Tree-based scheme, to conduct RFID tag

arbitration. The ALOHA protocol was first developed for random access in packet radio networks. To improve efficiency for RFID systems, the slotted ALOHA is developed [1][2]. In [9], Schoute analyzes the performance of the dynamic frame-sized ALOHA. In [10] Floerkemeier presents a Bayesian strategy to dynamically determine frame size. In [11], Vogt uses the Markov process to model the read process and suggests a set of dynamic frame sizes in the read process. An estimate function is also proposed to estimate the tag quantity. In [6], Lee et al. adopt Vogt's tag estimation method and derive the dynamic frame size for achieving maximum channel efficiency. They claim that if the frame size approximately equals the number of tags plus one, the maximum channel efficiency can be obtained. In [12], Murali et al. provide very fast and reliable estimation mechanisms for tag quantity in a more practical approach. To address the multiple-reading problem, Qian et al. propose the Lottery Frame scheme [13], a replicate-insensitive estimation protocol. In [14], Bo et al. consider the problem of identifying popular categories of RFID tags out of a large collection of tags. In [15], a fast *missing* tag detection protocol for RFID based inventory control applications was provided.

The unreliability of the physical layer in RFID systems has a crucial impact on tag reading. To show the effect of errors on the C1G2 protocol, Mitsugi et al. [16] present simulation results that bit errors significantly degrade the C1G2 performance. In [3], Buettner et al. examine the performance of the C1G2 RFID system in a realistic setting. They identify factors that degrade overall performance and reliability with a focus on the physical layer. They find that physical layer considerations have a significant impact on reader performance, and that this is exacerbated by a lack of integration between physical and MAC layers. In [5], Aroor et al. identify the state of the technical capability of passive UHF RFID systems using a simple, empirical, experimental approach. They examine various read distances for free space, near metal and near water situations. In [4], Ramakrishnan et al. describe the first comprehensive benchmark suite for passive UHF RFID tags. In [17] Ren et al. performed extensive experiments on an industrial conveyor belt to determine the effects of mobility on RFID reader performance. In [18] Jeffery et al. conduct experiments in realistic settings and find that within each reader's detection range, there are two distinct regions: major detection region and minor detection region.

III. PRELIMINARY

A. Far-Field Propagation and Backscatter Principle

Class-1 Generation-2 (C1G2) RFID systems [19] as defined by EPCglobal are based on UHF frequencies. They use *far-field communication* and the physical property of *backscattering* power [20]. Far-field communication uses the electric radio waves, where the reader sends a continuous base signal that is reflected back by the tag's antenna. A backscatter tag operates by modulating the electronics connected to the antenna so as to control the reflection of electromagnetic energy. Suppose a reader and a tag are separated by a distance of r in the free space propagation scenario. The reader's antenna gain is G_r

and the tag's antenna gain is G_t. The wave length is λ. Assume the transmitter power from the reader is P_1 measured in *watts*. By considering power flux density, the power received at the tag is $P_2 = \frac{P_1 \cdot G_r \cdot G_t \cdot \lambda^2}{(4\pi)^2 r^2}$. According to the radar principle, the amount of energy reflected by an object is dependent on the reflective area of the object. This area is referred to as the *radar cross section* (RCS). We use σ to denote the RCS, then the reflected power from the tag towards the reader is $P_3 = \frac{P_1 \cdot G_r \cdot \sigma}{(4\pi)^2 r^2}$. The power density yielded at the receiver of the reader is given by $P_4 = \frac{P_1 \cdot G_r^2 \cdot G_t \cdot \lambda^2 \cdot \sigma}{(4\pi)^4 r^4}$. According to the above analysis, we have the path loss as follows:

$$\frac{P_1}{P_2} = \frac{P_3}{P_4} = \frac{1}{G_r G_t}\left(\frac{4\pi r}{\lambda}\right)^2. \tag{1}$$

According to Eq.(1), we define the free space path loss over distance r as $PL_{fs}(r) = (\frac{4\pi r}{\lambda})^2$. Apparently the above free space propagation model is *deterministic*. However, this model rarely describes the actual propagation situation accurately. In realistic settings the propagation is *probabilistic*, because orientations of tags affect the backscatter efficiency, and phenomenons such as absorption and multi-path fading further make the attenuation not accurately predicted.

We denote the down-link communication from the reader to a tag as the *forward channel*, and denote the up-link communication from a tag to the reader as the *reverse channel*. For successful reading of a passive tag with the backscatter scheme, there are two thresholds to meet the physical requirements. The first is the *tag power (sensitivity) threshold*, P_{ts}. It is the minimum received power necessary to turn on an RFID chip. The second is the *reader sensitivity threshold*, P_{rs}. It is the minimum level of the tag signal that the reader can detect and resolve. Thus it must satisfy $P_2 > P_{ts}$ for the tag to be powered up and resolve the received signal, and also $P_4 > P_{rs}$ for the reader to detect and resolve the received signal.

B. Tag inventory and access

The MAC protocol for the C1G2 system is based on *Slotted ALOHA*, where each frame has a number of slots and each active tag will reply in a randomly selected slot per frame. When a reader (interrogator) wishes to read a set of tags, it first powers up and transmits a continuous wave (CW) to energize the tags. It then initiates a series of frames, varying the number of slots in each frame to best accommodate the number of tags. After all tags are read, the reader powers down. We refer to an individual frame as a *Query Round*, and the series of *Query Rounds* between power down periods as a *Query Cycle*. For each *Query Round*, the reader can optionally transmit a *Select* command which limits the number of active tags by providing a bit mask. Then a *Query* command is transmitted which contains the uplink frequency and data encoding, the *Q* parameter determining the number of slots for the following frame, and a *Target* parameter. When a tag receives a *Query* command, it chooses a random number in the range $(0, 2^Q - 1)$, where Q is in the range (0,15), and the value is stored in the slot counter of the tag. If a tag stores a 0 in its slot counter, it will immediately backscatter a 16 bit random number, denoted

by *RN16*. Upon receiving *RN16*, the reader will echo *RN16* in an *ACK* command. If the tag successfully receives *RN16*, it will backscatter its ID information (*PC+EPC+CRC16*). Then a subsequent *QueryRep* command will be sent to the tag, signaling the end of the slot and toggling an *inventoried* flag in the tag to make it keep silent in the following rounds. If the ID is not successfully received by the reader, an *NAK* command is sent which resets the tag so as to keep the tag active in the next round. Upon receiving the *QueryRep* command, the remaining tags will decrement their slot counters, and respond with *RN16* if their slot counters are set to 0. When the number of *QueryRep*s is equal to 2^Q, the current *QueryRound* ends. If there are still tags left unread, then the reader issues a *QueryAdjust* command to start a new *Query Round*, and all active tags which have received the first *Query* command can contend for slots in this frame. Note that, within each frame, tags may choose the same random number for the slot counter, which causes multiple tags to reply during a slot. Therefore within each frame there exist three kinds of slots: (1) the empty slot where no tags replies; (2) the single slot where only one tag replies; and (3) the collision slot where multiple tags reply.

IV. PROBLEM FORMULATION

Suppose a freight of containers is moving on a conveyor. The reader must read data from those tags inside the containers. Fig. 1 illustrates this scenario of reading moving tags. We suppose the conveyor has a constant moving speed v. The reader is fixed over the conveyor at a height of h, and it has an effective range of D on the horizon. The tag containers are placed along the conveyor with a distance of $S \geq D$ between each other. In this way, only one container is in the reading range of the reader at any time. Assume the number of tags in a container is N. As the time interval for one container to stay within the effective reading range is limited to $\frac{D}{v}$, the RFID reader system needs to finish identifying all tags inside each container within this time interval. Note that due to random issues in the anti-collision scheme, e.g, the system may occasionally encounter much higher collisions than the expected situation, in realistic settings the reader can never guarantee to finish identifying all tags in time, whatever the speed of the conveyor is. Therefore the objective to design the RFID system is to make the expected duration for identifying all tags as short as possible, so that all tags can be identified in time with high possibility. Fig. 2 illustrates the intuition of this objective. In Fig. 2(a) we use *Poisson distribution* to depict the duration distribution to finish identifying all N tags. The horizontal axis k is the duration and the vertical axis is the probability. We show three distributions with various mean durations $\lambda = 1, 4, 10$. In Fig. 2(b) we depict the corresponding cumulative distribution functions. We observe that in order to finish identifying all tags within a fixed duration k, the scheme with the smallest mean duration $\lambda = 1$ always has the highest probability to achieve that, while the scheme with $\lambda = 10$ always has the lowest probability. Therefore, in order to ensure all tags can be identified in time with high possibility, a practical method is to make the expected duration

to finish reading all tags as small as possible. Assume that the average durations for empty/single/collision slots are all nearly equal to Δt. Since the reader is responsible to set the frame size f for each *Query Round*, it needs to efficiently set a series of frame sizes $f_1, f_2, ..., f_m$ $(0 < m < +\infty)$ so that the expectation of $\sum_{i=1}^{m} f_i \cdot \Delta t$ is minimized. Since Δt is constant, this optimization problem is further reduced to minimizing the expectation of $\sum_{i=1}^{m} f_i$, which is the total expected number of slots used to read all the tags.
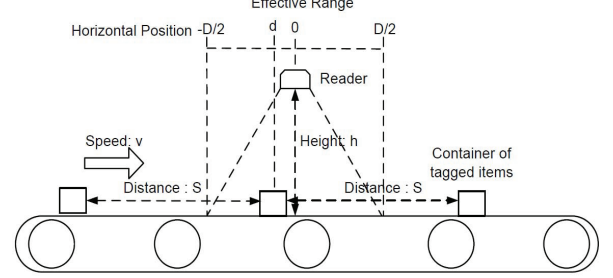

Fig. 1.   Reading the moving tags on the conveyor.

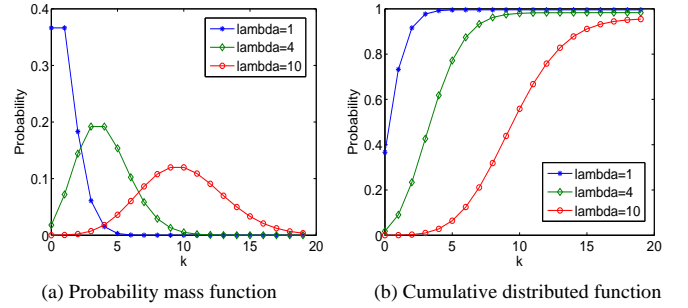(a) Probability mass function   (b) Cumulative distributed function

Fig. 2.   An example to explain the intuition of the objective.

While a container is moving within the reader's range, its horizontal position is d (see Fig.1). For the ease of presentation, we respectively set the horizontal position at the entrance, the middle and the exit of the reader's effective range as $-\frac{D}{2}, 0$ and $\frac{D}{2}$. Thus the distance between the tags in the container and the reader is $r = \sqrt{d^2 + h^2}$. So when the container is moving on the conveyor, the propagation loss for both the *forwarding channel* and the *reverse channel* keeps changing as the distance r varies. Hence the probability varies for the tags to successfully receive and backscatter the signal to the reader, affecting the whole procedure for the tag inventory and access. In order to solve the aforementioned optimization problem, we need to investigate the probabilistic model for RFID tag identification. Thus in the next two sections, we first propose a probabilistic model for RFID tag identification, then we accordingly propose an efficient tag identification scheme for identifying moving tags.

V. PROBABILISTIC MODEL FOR RFID TAG IDENTIFICATION

A. Probabilistic parameters for propagation

According to Section III-A, we know that the realistic propagation model is *probabilistic*, since the path loss and the

backscatter efficiency are *probabilistic*. In order to depict the RFID tag identification process under the probabilistic propagation situation, assume that the distance between the reader and the tag to be identified is r. We use $p_i(r)$ to denote the probability for the tag to successfully receive the message from the reader, and then use $p_t(r)$ to denote the probability for the reader to successfully receive the backscattered message from the tag. Then by definition we have

$$p_i(r) = Prob\{P_2 \geq P_{ts}\}, \tag{2}$$

$$p_t(r) = Prob\{P_4 \geq P_{rs} | P_2 \geq P_{ts}\}. \tag{3}$$

B. Probabilistic model for tag identification

According to Section III-B, considering one specified RFID tag, we can divide the tag inventory and access procedure into four phases:(1) Obtain frame size from the *Query/QueryAdjust* message, (2) Monitor the *QueryRep* messages for selected slot, (3) Backscatter the *RN16* message, and (4) Backscatter the ID message. Here we do not consider the optional *Select* message in the probabilistic model. In the following we respectively analyze the actual impact on the four phases due to the probabilistic propagation.

1) Obtain frame size from Query/QueryAdjust message: Assume before each *Query Round*, the number of tags left unread is n. When the reader sends the *Query/QueryAdjust* message, the tag will detect and resolve this message with probability $p_i(r)$. Thus $n \cdot p_i(r)$ tags are able to select the slots inside the frame.

2) Monitor QueryRep messages for selected slot: For those tags which have successfully selected the slot number according to the frame size f, each tag counts down the slot counter *slot* according to the received *QueryRep* messages. Each *QueryRep* message is detected and resolved by the tag with probability $p_i(r)$. When $slot = 0$ the tag backscatters the *RN16* message to the reader. Due to the loss of *QueryRep* messages, the tag actually may have to wait for more slots after the exact slot it selects. In this situation suppose a tag has selected the xth slot ($1 \leq x \leq f$) within the frame size f, the probability for it to detect x *QueryRep* messages at the actual tth ($t \geq x$) slot is

$$p_\alpha(t,x,r) = (p_i(r)) \cdot C_{t-1}^{x-1}(p_i(r))^{x-1}(1-p_i(r))^{t-x}.$$

As t varies over the range $[x, +\infty]$, if $t \leq f$, then the tag can reply inside the frame; otherwise, the tag cannot reply inside the frame. Therefore if a tag originally selects the xth slot, the probability for it to successfully reply in the frame is

$$p_\beta(f,x,r) = \sum_{t=x}^{f} p_\alpha(t,x,r).$$

As each tag uniformly selects a slot from 1 to f, then the probability for it to successfully reply in the frame is

$$p_\theta(f,r) = \sum_{x=1}^{f} \frac{1}{f} p_\beta(f,x,r) = \frac{1}{f} \sum_{x=1}^{f} \sum_{t=x}^{f} p_\alpha(t,x,r). \tag{4}$$

Theorem 1: According to the probabilistic propagation model, the probability for a tag to successfully reply in the frame with size f by monitoring the *QueryRep* messages is equal to $p_i(r)$, that is $p_\theta(f,r) = p_i(r)$.

Proof: Since $p_\alpha(t,x,r)$ denotes the probability for the tag to receive x *QueryRep* messages at the tth slot, and in order to succeed to do it within the specified frame size f, it should satisfy $1 \leq x \leq t \leq f$. Therefore according to the definition for $p_\theta(f,r)$ in Eq. (4), $p_\theta(f,r)$ can be expressed in the following equivalent form:

$$p_\theta(f,r) = \frac{1}{f} \sum_{t=1}^{f} \sum_{x=1}^{t} p_\alpha(t,x,r).$$

Then, as we have

$$\sum_{x=1}^{t} p_\alpha(t,x,r) = p_i(r) \cdot \sum_{x=1}^{t} C_{t-1}^{x-1}(p_i(r))^{x-1}(1-p_i(r))^{t-x}$$
$$= p_i(r) \cdot [p_i(r) + (1-p_i(r))]^{t-1} = p_i(r),$$

thus

$$p_\theta(f,r) = \frac{1}{f} \sum_{t=1}^{f} p_i(r) = p_i(r).$$

The theorem gets proved. ∎

The *Query* message actually acts as the role of the first *QueryRep* message, especially when $f = 1$ the unique *Query* message takes the role of the *QueryRep* message. Thus the first phase and the second phase are not independent to each other. However, as long as the frame size f is large enough, we can assume that the first phase and the second phase are independent to each other. In this situation, given the number of unread tags, n, the expected number of tags successfully obtaining a slot inside the frame is

$$n' = n \cdot p_i(r) \cdot p_\theta(f,r) = n \cdot (p_i(r))^2.$$

Note that after successfully receiving the *Query* message, tags randomly select the slots inside the frame with size f, using a uniform approach. Then due to the probabilistic loss of *QueryRep* messages, the actual slot t in which the tag replies is deferred to some extent over the original selected slot x. Since the original slot x is selected with uniform probability, then as indicated in Theorem 1's proof, the probability for the tag actually replies in slot t is $P(t) = \sum_{x=1}^{t} \frac{1}{f} \cdot P_\alpha(t,x,r) = \frac{p_i(r)}{f}$. Since $P(t)$ is the same for all slot t, it infers that the actual slot t in which the tag replies is uniformly distributed inside the frame. Therefore, if we do not consider the propagation loss for backscattering, then, according to the binomial distribution, the expected number of empty slots, single slots, and collision slots, n_0, n_1, n_c, can be calculated as follows:

$$\begin{cases} n_0 = f \cdot (1 - \frac{1}{f})^{n'}, \\ n_1 = n' \cdot (1 - \frac{1}{f})^{n'-1}, \\ n_c = f - n_0 - n_1. \end{cases} \tag{5}$$

We denote the above mentioned empty/single/collision slot as *original empty/single/collision slot*, which means

no/one/multiple tag(s) select(s) this slot and is (are) ready to backscatter the *RN16* message to the reader. Specifically, we define *s-collision slot* as the slot which is selected by s tags.

3) Backscatter the RN16 *message:* If a tag has received enough *QueryRep* messages for its selected slot, then it backscatters the *RN16* message to the reader, and the reader is able to detect and resolve the backscattered message with probability $p_t(r)$. According to the slotted Aloha protocol, due to the probabilistic loss of the *RN16* message, for some *original single slots* it is possible for the reader unable to detect the *RN16* messages, thus transferring those original single slots into *actual empty slots*. For the *original s-collision slots*, s tags try to send the *RN16* message simultaneously. Due to the probabilistic loss of the message, when all s messages are lost with a probability of $p_1(s, r)$, it is possible that the reader is unable to detect any of the messages, thus transferring this slot into *actual empty slot*; similarly when $s - 1$ messages are lost with a probability of $p_2(s, r)$, it is possible for the reader to detect only one of the messages, thus transferring this slot into *actual single slot*; otherwise, this slot remains as a collision slot with a probability of $p_3(s, r)$. The above probabilities, $p_1(s, r), p_2(s, r)$, and $p_3(s, r)$, are calculated as follows:

$$\begin{cases} p_1(s, r) = (1 - p_t(r))^s \\ p_2(s, r) = s \cdot (1 - p_t(r))^{s-1} \cdot p_t(r) \\ p_3(s, r) = 1 - p_1 - p_2 \\ \quad = 1 - (1 - p_t(r))^s - s \cdot (1 - p_t(r))^{s-1} \cdot p_t(r) \end{cases} \quad (6)$$

Suppose the actual numbers of empty slots, single slots and collision slots are n'_0, n'_1, n'_c. Fig. 3 shows the translation relationship from original empty/singel/collision slots to actual empty/singel/collision slots.
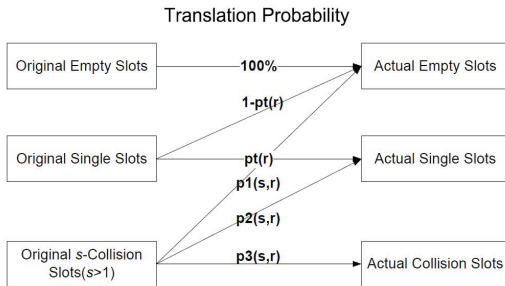

Fig. 3. Relationship for slot transformation

Suppose n' tags are able to successfully obtain a slot in the frame of size f, then the proportion for the *original s-collision slots* $P(s)$ inside the frame is

$$P(s) = C_{n'}^s \cdot (\frac{1}{f})^s (1 - \frac{1}{f})^{n'-s}.$$

Thus given f, n' and $p_t(r)$, we can calculate n'_0, n'_1, n'_c as follows:

$$\begin{cases} n'_0(n, r, f) = n_0 + n_1 \cdot (1 - p_t(r)) \\ \qquad\qquad + f \cdot \sum_{s=2}^{n'} P(s) \cdot p_1(s, r), \\ n'_1(n, r, f) = n_1 \cdot p_t(r) + f \cdot \sum_{s=2}^{n'} P(s) \cdot p_2(s, r), \\ n'_c(n, r, f) = f \cdot \sum_{s=2}^{n'} P(s) \cdot p_3(s, r). \end{cases} \quad (7)$$

4) Backscatter the ID message: When a tag successfully transmits a *RN16* message to the reader without collisions, the reader will transmit a *ACK* message to the tag, and then the tag will backscatter its ID message to the reader. The above process is successfully completed with a probability of $p_i(r) \cdot p_t(r)$, otherwise the tag will keep active for the next round. Therefore the number of tags that successfully read within the frame is $n'_1(n, r, f) \cdot p_i(r) \cdot p_t(r)$.

C. Obtain the propagation and backscatter parameters

In the above subsection, we have built a probabilistic model for the RFID tag identification. Recall that we utilize two parameters $p_i(r)$ and $p_t(r)$ to formalize the probabilistic model, thus obtaining these two important parameters is a prerequisite to the probabilistic model. However, it is rather difficult to directly compute these two parameters by using realistic physical layer parameters obtained from sophisticated instruments [3]. Here we claim that by using one reader and one tag, it is enough to estimate $p_i(r)$ and $p_t(r)$ from the following experiment results. We separate the reader and the tag with a distance of r. First we set the frame size f to 1 for the reader, and let the reader issue m_1 rounds to read the same tag and check the number of successful responses m'_1. Thus we get the read ratio as $rt_1(r) = \frac{m'_1}{m_1}$. Secondly we set the frame size f to a large enough value, say $f = 100$, and also let the reader issue m_2 rounds to read the same tag and check the number of successful responses m'_2. Thus we get the read ratio as $rt_f(r) = \frac{m'_2}{m_2}$.

According to the probabilistic model, for the case when frame size $f = 1$, the *Query* message is successfully detected and resolved by the tag with probability of $p_i(r)$, and the tag successfully backscatters the *RN16* message with probability of $p_t(r)$. After that the reader successfully responds the *ACK* message with probability of $p_i(r)$, and the tag successfully backscatters the ID message with probability of $p_t(r)$. So $rt_1(r) = (p_i(r))^2 \cdot (p_t(r))^2$. For the case when frame size f is set to a large enough value, first the *Query* message is successfully detected and resolved by the tag with probability of $p_i(r)$, then the tag monitors enough *QueryRep* messages for its selected slot with probability $p_\theta(f, r)$. It has been proved that $p_\theta(f, r) = p_i(r)$. Then the tag successfully backscatters the *RN16* message with probability of $p_t(r)$. After that the reader successfully responds the *ACK* message with probability of $p_i(r)$, and the tag successfully backscatters the ID message with probability of $p_t(r)$. Hence $rt_f(r) = (p_i(r))^3 \cdot (p_t(r))^2$. Therefore we can derive that $p_i(r) = \frac{rt(f)}{rt(1)}$ and $p_t(r) = \sqrt{\frac{(rt(1))^3}{(rt(f))^2}}$. We can store the values of $p_i(r)$ and $p_t(r)$ at various distance r. In our probabilistic model, for simplification of the problem we do not consider the impact

on the propagation probability due to the various message lengths, because these command messages all have very small message lengths. If it is necessary to take the message length into consideration, we can similarly estimate the bit-wise propagation probability parameters according to the above methodology.

VI. EFFICIENT TAG IDENTIFICATION ACCORDING TO THE PROBABILISTIC MODEL

A. Dynamic programming based solution

In this section we solve the optimization problem proposed in Section IV by using dynamic programming. According to the problem statement in Section IV, we further summarize the notations in Table I.

TABLE I
NOTATION OF PARAMETERS

Parameters	Definitions
$F(n, d)$	The minimum expected number of slots required to read n tags starting from horizontal position d
$h(n, d, f)$	Given n tags, the expected number of tags read in frame size f starting from position d
$g(n, d, f)$	Given n tags, the expected number of tags left unread in frame size f starting from position d
Δt	The average time interval for one slot
$t(d)$	The upper bound for candidate frame sizes, which is the minimum value of the supported maximum frame size f_{max} and the number of available time slots f_{ava}, i.e., $t(d) = \min\{f_{max}, f_{ava}\}$, where $f_{ava} = \frac{D/2 - d}{v \cdot \Delta t}$

Without loss of generality, we assume by adjusting the speed of the conveyor, v, all tags can be identified within the reader's effective range for average cases, i.e., the expected duration for identifying all N tags is smaller than D/v. For $F(n, d)$, which is the minimum expected number of slots required to read n tags starting from position $d(-D/2 \leq d \leq D/2)$, it has the formulation as follows:

$$F(n, d) =$$
$$\begin{cases} 0 & \text{if } n = 0 \\ \frac{1}{(p_i(r))^2 \cdot (p_t(r))^2} & \text{if } n = 1 \\ \min_{\{0 < f \leq t(d)\}}\{f + F(g(n, d, f), d + fv\Delta t)\} & \text{otherwise.} \end{cases}$$

In the above formulation, if $n = 0$, there is no tag left to be read, then the minimum number of slots required is equal to 0. If $n = 1$, then the number of slots required is expected to be $\frac{1}{(p_i(r))^2 \cdot (p_t(r))^2}$, where $r = \sqrt{d^2 + h^2}$, and h is the height of the reader over the conveyor. Otherwise when $n > 1$, by enumerating all possible frame sizes $f \in [1, t(d)]$ for the first frame, $F(n, d)$ is obtained from the minimum value of $f + F(g(n, d, f), d + fv \cdot \Delta t)$, where the second term denotes the minimum expected number of slots needed to read all tags left after we choose the first frame size f. Note the position is updated to $d + f \cdot v \cdot \Delta t$. Here we assume we can choose any integer value for the frame size f. If we need to conform to the Q protocol [1], we can select the candidate frame size f from the set $2^0, 2^1, ..., 2^Q$, for $2^Q \leq t(d)$ and $Q \leq 15$.

Now we need to calculate $g(n, d, f)$, the expected number of tags left unread in frame size f. As the tags' container is continuously moving on the conveyor, the container's position actually changes slightly for each slot inside the frame. Since the candidate frame size f is restricted in $[1, t(d)]$, according to the definition of $t(d)$ in table I, as long as we set a not so large value for f_{max} (empirically f_{max} should not be larger than n for efficiency reason), we can assume that the container's position does not change too much within each selected frame. Then the average position for those tags within the frame is $d' = d + 0.5 \cdot fv\Delta t$. Thus the average distance between the tags and the reader within the frame is $r = \sqrt{d'^2 + h^2}$. According to the probabilistic model, the expected number of tags successfully read within the frame size f is

$$h(n, d, f) = n'_1(n, r, f) \cdot p_i(r) \cdot p_t(r).$$

Hence,

$$g(n, d, f) = n - h(n, d, f).$$

Based on the above analysis, as the total number of tags in one container is N, and the starting position for the container to enter the reader's range is $d = -D/2$, the objective for the optimization problem is to calculate $F(N, -D/2)$. According to the formulation of $F(n, d)$, this optimization problem has overlapping subproblems and optimal substructures. We can solve it with dynamic programming. Algorithm 1 utilizes the dynamic programming methodology to calculate various $F(n, d)$ for various (n, d) pairs. We set $M = \frac{D}{v \cdot \Delta t}$ as the number of discrete positions at different slots, and we use $i(0 < i \leq M)$ to denote the index of the discrete positions, then the corresponding discrete position $d = (i - M/2) \cdot v \cdot \Delta t$. Then we maintain a table $F[n][i]$ of size $(N + 1) \times M$ to store all discrete values for $F(n, d)$, and a table $f[n][i]$ of size $(N + 1) \times M$ to store the optimal values for the first frame size f corresponding to $F(n, d)$. Inside this algorithm, we first initialize the $F[n][i]$ values for $n = 0, 1$. Then we iteratively calculate the optimal values for $F[n][i]$ and $f[n][i]$ in a bottom-up approach. When the iteration finishes, all the $(N + 1) \cdot M$ values are calculated.

By utilizing the optimal parameters stored in $f[n][i]$, we propose Algorithm 2 to perform the RFID tag identification. The reader first performs a read round with initial frame size C and obtains n_0, n_1, n_c and n_s. Here n_0, n_1, n_c are respectively the number of empty/single/collision slots, while n_s is the number of tags successfully identified in the single slots. Then the reader estimates the number of active tags for this round as n, thus the number of tags left unread is $n - n_s$. In the following steps for each round the reader first selects the optimal frame size f^*. Given the number of tags left unread $n - n_s$ and the current position of the container d, the function selectOptimalFrameSize($n - n_s, d$) in Alg.2, line 7 just outputs $f^* = f[n - n_s][i]$ as the optimal frame size, where $i = \frac{d + D/2}{\Delta d}$. Then the reader performs a read round with the frame size f^*. After that the reader estimates the number of active tags for this round. In order to guarantee those tags left unread to always have opportunities to seize a slot inside the frame,

the reader issues a *Query* command instead of the *QueryAdjust* command to start each frame. The above process continues till both the number of single slots n_1 and the number of collision slots n_c are equal to 0. Due to the probabilistic propagation loss, this gives a tighter constraint to finish the reading cycle than simply checking if $n_c = 0$, thus making the number of tags left unread as small as possible.

Algorithm 1 Decide optimal frame sizes for slotted-ALOHA

1: $\Delta d = v \cdot \Delta t$
2: $M = \frac{D}{v \cdot \Delta t}$
3: **for** $i \in [0, M-1]$ **do**
4:     $F[0][i] = 0$, $f[0][i] = 0$
5:     $d = (i - M/2) \cdot \Delta d$
6:     $r = \sqrt{d^2 + h^2}$
7:     $F[1][i] = \lceil \frac{1}{(p_i(r))^2 \cdot (p_t(r))^2} \rceil$, $f[1][i] = 1$
8: **end for**
9: **for** $n \in [2, N]$ **do**
10:    **for** $i \in [0, M-1]$ **do**
11:        $d = (i - M/2) \cdot \Delta d$
12:        $r = \sqrt{(d + 0.5 \cdot f \cdot \Delta d)^2 + h^2}$
13:        $g(n, d, f) = \lfloor n - n_1'(n, r, f) \cdot p_i(r) \cdot p_t(r) \rfloor$
14:        $F[n][i] = \min_{\{0 < f \le M-i\}} \{f + F[g(n, d, f)][i + f]\}$
15:        $f[n][i] = \arg\min_{\{0 < f \le M-i\}} \{f + F[g(n, d, f)][i + f]\}$
16:    **end for**
17: **end for**

Algorithm 2 Tag identification with optimal frame sizes

1: $\Delta d = v \cdot \Delta t$
2: $d = -D/2$
3: $[n_0, n_1, n_c, n_s]$=performReadRound(C)
4: $[n]$=performTagSizeEstimate(f, d, n_0, n_1, n_c)
5: $d = d + C \cdot \Delta d$
6: **while** $n_c \ne 0$ or $n_1 \ne 0$ **do**
7:    $[f^*]$=selectOptimalFrameSize($n - n_s, d$)
8:    $[n_0, n_1, n_c, n_s]$=performReadRound(f^*)
9:    $[n]$=performTagSizeEstimate(f, d, n_0, n_1, n_c)
10:   $d = d + f^* \cdot \Delta d$
11: **end while**

Note that the objective for the optimization problem is to minimizing the total time for identifying all the tags. Since in the problem formulation we make the assumption that the average time intervals for empty/single/collision slots are nearly equal to Δt, the optimization problem is equivalent to minimize the total used slots. Actually if we consider distinct time intervals for various kinds of slots, the dynamic programming still works with slightly modification. Suppose the average time intervals for empty/single/collision slots are respectively t_e, t_s and t_c, then $F(n, d)$ can be reformulated as:

$$F(n, d) =$$
$$\begin{cases} 0 & \text{if } n = 0 \\ (\frac{1}{(p_i(r))^2 \cdot (p_t(r))^2}) \cdot t_s & \text{if } n = 1 \\ \min_{\{0 < f \le t(d)\}} \{T(f) + F(g(n, d, f), d + vT(f))\} & \text{Otherwise} \end{cases}$$

Here $T(f)$ is the time interval for the frame of size f, and we have $T(f) = n_0'(n, r, f) \cdot t_e + n_1'(n, r, f) \cdot t_s + n_c'(n, r, f) \cdot t_c$, and $r = \sqrt{(d + 0.5 f v \Delta t)^2 + h^2}$.

B. Estimate the number of tags

For each query round the reader needs to estimate the number of active tags, so as to provide an accurate input parameter for selecting the next optimal frame size. Current estimation schemes [12] [21] are not really suitable for our probabilistic model, because their methodology is based on the ideal case when $p_i(r) = p_t(r) = 1$. Therefore we propose an alternative scheme to estimate the number of tags. Our methodology is based on Theorem 2.

Theorem 2: As defined in Eq. (7), $n_0'(n, r, f)$ is monotonically decreasing with n. $n_c'(n, r, f)$ is monotonically increasing with n.

Due to lack of space, we omit the proof of Theorem 2. According to Theorem 2, since $n_0'(n, r, f)$ and $n_c'(n, r, f)$ are monotonic with variable n, given the constant values of $p_t(r), p_i(r), f$, the reader will obtain various expected values of n_0', n_c' with various n. Based on this property, we can estimate the number of tags n by leveraging the *binary search* method, according to either $n_0'(n, r, f)$ or $n_c'(n, r, f)$. Here we take the observed number of empty slots $n_0'(n, r, f)$ as the metric for estimation. Assume that the upper bound for n is N, i.e., $n \in [0, N]$, then the corresponding expected number of empty slots $n_0' \in [n_0'(N, r, f), n_0'(0, r, f)]$. We now use binary search to find the value of n according to n_0'. We start from the value $n' = N/2$, and compare $n_0'(n', r, f)$ with n_0'. If $n_0'(n', r, f) > n_0'$, then n falls into the range $(N/2, N]$. If $n_0'(n', r, f) < n_0'$, then n falls into the range $[0, N/2)$. Otherwise $n = N/2$. We continue the iterative search until we find an exact point value for n. This binary search procedure will iterate at most $\log_2(N)$ steps.

C. Adaptive frame size selection

In Section VI-A we have proposed a dynamic programming based solution for selecting optimal frame sizes. By using this methodology we can accurately select the optimal frame sizes as the container moves along the conveyor. However, the dynamic program requires to maintain two $(N+1) \times M$ sized tables. Algorithm 1 needs $(N+1) \times M$ iterations to calculate values for the tables. For each iteration different frame sizes $f \in (0, M - i]$ should be tested. According to Eq.(7) for each f it requires at most N iterations to calculate $n_1'(n, r, f)$, thus the computation complexity is $O(N^2 \cdot M^2)$. Although the two tables can be preprocessed in the reader beforehand, the values of N and M can be rather large numbers. Hence in some situations the reader may not afford to conduct such calculations.

Therefore an adaptive approach is essential for the reader to quickly select appropriate frame sizes while guaranteeing efficiency for the anti-collision scheme. We attempt to obtain a local optimal frame size for each *Query Round*, so as to maximize the proportion of single slots within each frame. Our solution relies on the following Theorem.

Theorem 3: To maximize the proportion of single slots within each frame, $\frac{n_1'}{f}$, the local optimal frame size is $f^* = n \cdot (p_i(r))^2 \cdot p_t(r)$, and the efficiency is $\frac{n_1'}{f^*} = \frac{1}{e}$, here e is the natural number.

Proof: According to the probabilistic model, after the phases (1) and (2), only $n' = n \cdot p_i(r)^2$ tags are able to be involved in the *Query Round*. We denote these tags as a set S. Then due to the probabilistic propagation in the reverse channel with probability $p_t(r)$, the final number of single slots is n_1'. For those tags in set S, since each tag will uniformly select a slot and successfully backscatter *RN16* with probability $p_t(r)$, we can depict the process with an equivalent model as follows. Each tag in set S is able to uniformly select a slot inside the frame with probability $p_t(r)$ and backscatter with probability equal to 1. Then based on this model the final number of tags involved in the *Query Round* is $n_f = n' \cdot p_t(r)$. According to the conclusion in [6], for the ideal propagation case, the local optimal frame size should be $f^* = n_f$ in order to maximize the channel efficiency for each frame, and the fraction of single slots is $\frac{1}{e}$. Thus we have $f^* = n' \cdot p_t(r) = n \cdot (p_i(r))^2 \cdot p_t(r)$, and the efficiency $\frac{n_1'}{f^*} = \frac{1}{e}$. The theorem gets proved. ∎

Based on Theorem 3, for each *Query Round* the reader can quickly calculate f^* according to the current number of active tags n, propagation probability $p_i(r)$, and $p_t(r)$ at the container's position d. Therefore the adaptive approach to select frame sizes is as follows: As the container enters the reader's effective range, the reader continuously tracks its position according to $d = v \cdot t - D/2$. For each *Query Round* the reader first estimates the number of tags left unread n. Since the values of $p_i(r)$ and $p_t(r)$ have already been captured for various distances $r = \sqrt{d^2 + h^2}$, then according to the current position d for the tags, the reader issues a new frame with size $f^* = n \cdot (p_i(r))^2 \cdot p_t(r)$. The process continues until no collision slots or single slots exist inside the frame.

VII. Performance Evaluation

We have conducted simulations in Matlab. We consider a scenario of reading moving tags on the conveyor, where the effective range for the reader is $D = 12$ meters (m), the moving speed of the conveyor is $v = 3$ meters/second (m/s), and the average time interval of each slot is $\Delta t = 10$ millisecond (ms). For the *Query Cycle* we set the default value for the initial frame size as $f = 10$. We respectively consider two situations for the propagation environment. The first situation is a controlled condition which is near to the free space environment, and the second situation is a noisy condition where propagation attenuations are distinct. We use the parameters in Fig. 4(a) to simulate the propagation probability for $p_i(r)$ and $p_t(r)$. Here for simplification we set $p_i(r) = p_t(r)$. As the reader's effective range is $D = 12$m, the distance between the tag and the reader ranges from 0m to 6m.

We evaluate the performance of various strategies to select frame sizes. In Fig. 4(b) we compare the required number of

slots for various strategies. We find that using the dynamic program based strategy (DP) always leads to the smallest number of slots to complete the tag reading, compared with the other strategies. The adaptive strategy (AD) to select local optimal frame sizes also achieves good performance for reading the tags, in that its required number of slots is only 7%-8% larger than the former. Both strategies require fewer slots than the strategy that selects $f = n$, which proves to be near optimal in the ideal propagation situation. The fixed frame size strategy with $f = 32$ always requires the largest number of slots, due to the lack of dynamic adjustment for frame size. In Fig. 4(c) we compare the required number of slots for the above four strategies under the two different propagation situations, when the number of tags is 50. From left to right the required numbers of slots are respectively corresponding to: (1) dynamic program based strategy, (2) adaptive strategy, (3) "$f = n$" strategy, and (4) fixed frame size strategy. We observe that for each strategy the required numbers of slots under the controlled situation are always smaller than the noisy situation. The reason is that in the noisy situation, the propagation loss is more frequent than the controlled situation, thus causing the reader to need more slots to finish reading the tags. In Fig. 4(d) we evaluate the normalized throughput for various initial frame sizes according to the dynamic program based strategy. The normalized throughput is defined as the number of identified tags divided by the required total number of slots. As the number of tags increases from 10 to 40, the throughput increases rapidly. When the number of tags is over 40, the throughput increases slowly and converges to about 0.25. The achieved throughput is smaller than 0.368, which is the theoretical maximum throughput for the ideal propagation situation.

We evaluate the performance of estimating the number of tags in Fig.4(e). We track a query cycle and compare the estimated number of active tags and the actual number of active tags. We observe that for each round the estimated number of tags is very near the actual number of tags. The maximum estimate error is at most 14% when the number of actual tags is larger than 20. As the number of tags reduces, the maximum estimate error is at most 50%, this is because when the tag size is small, the variance for the number of empty slots or collision slots becomes relatively large, thus causing the estimate error larger than the large tag size case.

In Fig.4(f) we measure the expected maximum number of tags which can be identified for various conveyor speeds, under the noisy condition. We use the dynamic program based strategy to read tags, varying the conveyor speed v. For each specified conveyor speed v, we gradually increase the number of tags n to see if they can be identified in time. We run 500 simulations for each settings, if we find that those tags can be identified for most cases ($\geq 50\%$), we keep increasing n, otherwise we stop and get the corresponding expected maximum number of tags, $n_{max}(v)$. As v increases from $1m/s$ to 4m/s, the expected number of available slots reduces from 1200 to 300, hence $n_{max}(v)$ reduces from 250 to 65. It means that given a fixed conveyor speed, if the tag size
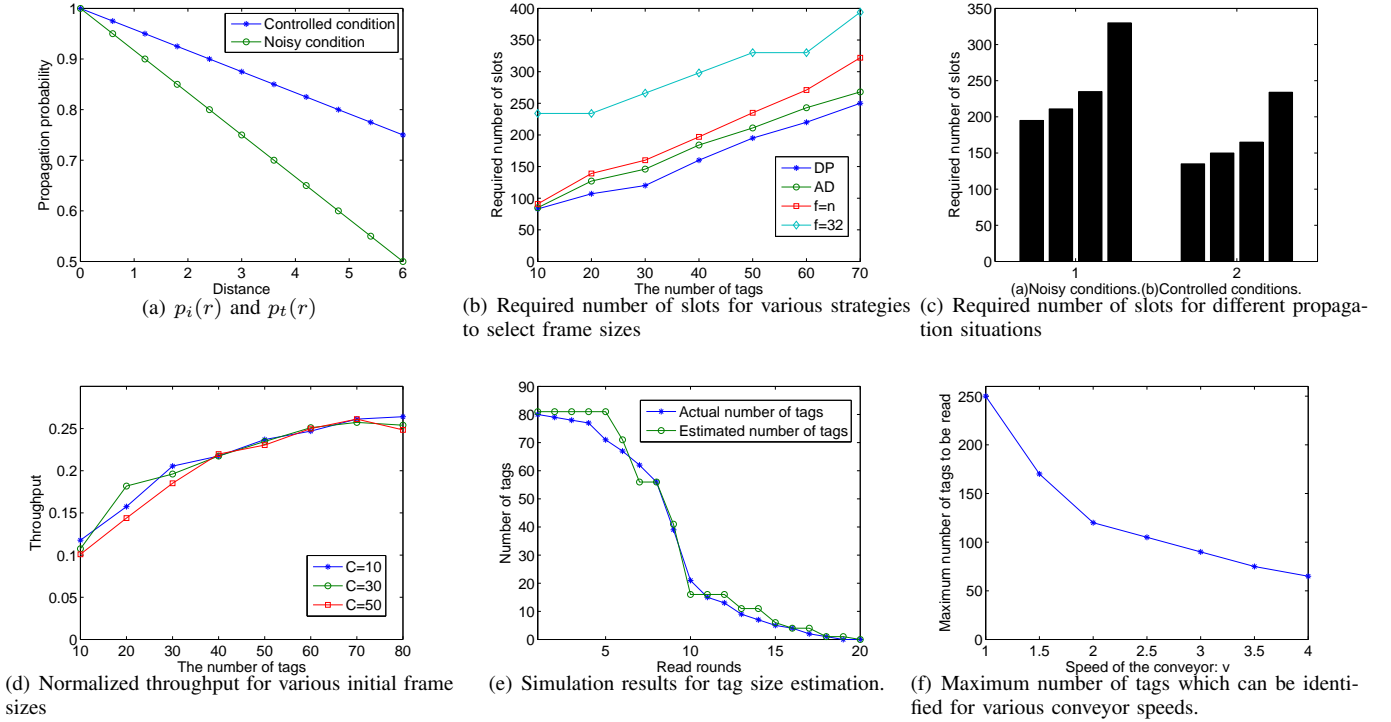
(a) $p_i(r)$ and $p_t(r)$

(b) Required number of slots for various strategies to select frame sizes

(c) Required number of slots for different propagation situations

(d) Normalized throughput for various initial frame sizes

(e) Simulation results for tag size estimation.

(f) Maximum number of tags which can be identified for various conveyor speeds.

Fig. 4.   Simulation results

exceeds $n_{max}(v)$, then conventionally the reader cannot finish identifying all the tags in time.

VIII. CONCLUSION

In this paper we consider how to efficiently identify tags on the moving conveyor. Considering conditions like the path loss and multi-path effect in realistic settings, we first propose a probabilistic model for RFID tag identification. Based on this model, we propose efficient solutions to identify moving RFID tags, according to the fixed-path mobility on the conveyor.

ACKNOWLEDGMENTS

REFERENCES

[1] Y. Maguire and R. Pappu, "An optimal q-algorithm for the iso 18000-6c rfid protocol," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 16–24, 2009.

[2] B. Zhen, M. Kobayashi, and M. Shimuzu, "Framed aloha for multiple rfid objects identification," *IEICE Transactions on Communications*, vol. E80-B, no. 3, pp. 991–999, 2005.

[3] M. Buettner and D. Wetherall, "An empirical study of uhf rfid performance," in *Proc. of MobiCom*, 2008.

[4] K. M. Ramakrishnan and D. D. Deavours, "Performance benchmarks for passive uhf rfid tags," in *Proc. of the 13th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems*, 2006.

[5] S. R. Aroor and D. D. Deavours, "Evaluation of the state of passive uhf rfid: An experimental approach," *IEEE Systems Journal*, vol. 1, no. 2, pp. 168–176, 2007.

[6] S. Lee, S. Joo, and C. Lee, "An enhanced dynamic framed slotted aloha algorithm for rfid tag identification," in *Proc. of MobiQuitous*, 2005.

[7] J. Myung, W. Lee, and S. Jaideep, "Adaptive binary splitting for efficient rfid tag anti-collision," *IEEE Communications Letters*, vol. 10, no. 3, pp. 144–146, 2006.

[8] L. Pan and H. Wu, "Smart Trend-Traversal: A Low Delay and Energy Tag Arbitration Protocol for Large RFID Systems," in *Proc. of INFOCOM, mini-conference*, 2009.

[9] F. Schoute, "Dynamic frame length aloha," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 565–568, 1983.

[10] C. Floerkemeier, "Bayesian transmission strategy for framed aloha based rfid protocols," in *Proc. of RFID*, 2007.

[11] H. Vogt, "Efficient object identification with passive rfid tags," in *Proc. of Pervasive*, 2002.

[12] M. Kodialam and T. Nandagopal, "Fast and reliable estimation schemes in rfid systems," in *Proc. of MobiCom*, 2006.

[13] C. Qian, H.-L. Ngan, and Y. Liu, "Cardinality estimation for large-scale rfid systems," in *Proc. of PerCom*, 2008.

[14] B. Sheng, C. C. Tan, Q. Li, and W. Mao, "Finding popular categoried for RFID tags," in *Proc. of ACM Mobihoc*, 2008.

[15] C. C. Tan, B. Sheng, and Q. Li, "How to monitor for missing RFID tags," in *Proc. of IEEE ICDCS*, 2008.

[16] Y. Kawakita and J. Mitsugi, "Anti-collision performance of gen2 air protocol in random error communication link," in *Proc. of SAINT-W*, 2006.

[17] Z. Ren, C. C. Tan, D. Wang, and Q. Li, "Experimental study on mobile RFID performance," in *Proc. of WASA*, 2009.

[18] S. R. Jeffery, M. Garofalakis, and M. J. Franklin, "Adaptive cleaning for rfid data streams," in *Proc. of VLDB*, 2006.

[19] "Epcglobal. epc radio-frequency identity protocols class-1 generation-2 uhf rfid protocol for communications at 860 mhz-960mhz version 1.2.0," Tech. Rep., 2008.

[20] H. Lehpamer, *RFID Design Principles*.   Artech House, Incorporate, 2008.

[21] W. Chen, "An accurate tag estimate method for improving the performance of an rfid anticollision algorithm based on dynamic frame length aloha," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 9–15, 2009.