2010 International Conference on Distributed Computing Systems

Using Analog Network Coding to Improve the RFID Reading Throughput

Ming Zhang Tao Li Shigang Chen Bo Li
Department of Computer & Information Science & Engineering
University of Florida, Gainesville, FL 32611, USA

Abstract—**RFID promises to revolutionize the inventory management in large warehouses, retail stores, hospitals, transportation systems, etc. Periodically reading the IDs of the tags is an important function to guard against administration error, vendor fraud and employee theft. Given the low-speed communication channel in which a RFID system operates, the reading throughput is one of the most important performance metrics. The current protocols have reached the physical throughput limit that can possibly be achieved based on their design methods. To break that limit, we have to apply fundamentally different approaches. This paper investigates how much throughput improvement the analog network coding [1] can bring when it is integrated into the RFID protocols. The idea is to extract useful information from collision slots when multiple tags transmit their IDs simultaneously. Traditionally, those slots are discarded. With analog network coding, we show that a collision slot is almost as useful as a non-collision slot in which exactly one tag transmits. We propose the framed collision-aware tag identification protocol that optimally applies analog network coding to maximize the reading throughput, which is** 51.1% $\sim 70.6\%$ **higher than the best existing protocols.**

I. Introduction

The barcode system brings numerous benefits for the retail stores. It speeds up the checkout process, makes the price change easier, and allows quick access for the properties of each merchandize item. It also has serious limitation. A barcode can only be read in close range. Suppose an inventory management policy requires periodical reading of all items in order to guard against administration error, vendor fraud and employee theft. One will have to use a portable laser scanner and manually read the barcodes one after another, which is tedious and error-prone. RFID tags, which can be read wirelessly, provide an ideal solution to this problem [2], [3]. Each tag carries a unique identification number (ID), and a RFID reader can retrieve the ID of a tag even when there are obstacles between them. Although the passive tags are most popular, they are not suitable for automated inventory management in a large area because they can only be read in a few meters. In order to read all tags, we have to either deploy numerous readers, each covering a small area, or manually move a reader around, which is again inefficient and error-prone. This paper considers the battery-powered active (or semi-passive) tags that can be read in a long distance and have more software/hardware resources than the passive tags.

The communication between the RFID reader and the tags is operated in a low-speed channel. Yet the number of tags in a large RFID system is expected to be very large. Therefore, one of the most critical performance metrics is the *reading throughput*, which is the average number of unique tag IDs that the reader can collect in a second. The current protocols have reached the physical throughput limit that can be achieved based on their design methods. In the *time-slotted ALOHA-based* protocols [4], [5], [6], [7], [8], [9], [10], a tag transmits its ID in each time slot (or some slot in a frame) with a certain probability p until the receipt of its ID is acknowledged by the RFID reader. The reading throughput is fundamentally limited by the probabilistic collision that occurs in ALOHA-based networks. The optimal throughput is $\frac{1}{eT}$, where e is the natural constant and T is the length of a time slot [11]. It is achieved when p is chosen such that the probability for exactly one tag transmitting in each slot is 36.8%. The other major class is the *tree-based protocols*, which organize the reading process in a binary tree structure [12], [13], [14] and improve the reading throughput by balancing the tree [12], [15], [16]. Analytical and simulation results have shown that the best performance of the tree-based protocols is comparable to the best of the ALOHA-based protocols.

To break the fundamental limit of the ALOHA-based protocols, we have to resort to fundamentally different approaches. In this paper, we apply the recently-proposed analog network coding scheme [1] to RFID systems and investigate how significantly it can improve the reading throughput.

What limits the throughput of the ALOHA-based protocols? Radio collision, which happens when more than one tag transmits in a slot. The conventional wisdom is that collision slots do not carry useful information and therefore those slots are wasted. That is however not true. Recent research shows that, by embracing the interference of wireless communication, physical-layer network coding can significantly improve the network throughput [17]. In particular, the analog network coding scheme [1] has been experimentally implemented. However, its usefulness has only been demonstrated under "toy" examples.

The contributions in this paper are two-fold: First, we optimally integrate analog network coding into the RFID system to maximize the reading throughput by making some *collision slots* almost as useful as *non-collision slots* (in which only one tag transmits). The difference is that the former allow the RFID reader to learn new tag IDs after some time, while the latter let the reader learn new IDs right away. Second, we demonstrate the practical value of the analog network coding research by providing an interesting application scenario.

Technically, we design the first collision-aware tag identifi-

1063-6927 2010

DOI 10.1109/ICDCS.2010.48
IEEE computer society

cation protocol that establishes the engineering and theoretical foundation for integrating analog network coding into the process of tag reading. We derive the optimal system parameters for improving the reading throughput. We also reduce the protocol overhead through a framed structure and an embedded estimator for the number of tags that are currently participating in the protocol. The proposed protocol is able to efficiently utilize the information carried in collision slots and thus break the fundamental limit of ALOHA-based protocols that do not use analog network coding. Our work answers two important questions: How to optimally apply analog network coding for RFID reading? How much throughput gain can analog network coding bring? The simulation results show that the reading throughput can be improved by $51.1\% \sim 70.6\%$ when using today's analog network coding method and the throughput can be much higher if the coding method is improved in the future.

The rest of the paper is organized as follows. Section II presents the motivation of our work. Section III gives the problem definition. Section IV proposes a collision-aware tag identification protocol and derives the optimal system parameters. Section V improves the protocol for less overhead. Section VI presents the simulation results. Section VII discusses the related work. Section VIII draws the conclusion.

II. Background

A. Motivation

Consider a RFID system with a large number of active (or semi-active) tags deployed in a region. We assume that the RFID reader and the tags transmit with sufficient power such that they can communicate over a long distance. The problem is for the reader to collect the IDs of all tags within the communication range. If the communication range cannot cover the whole deployment region, the reader may have to perform the reading process at several locations and remove the duplicate IDs when some tags are covered by multiple readings. In this paper, we focus on the reading operation at a single location, Our goal is to optimize the *reading throughput*, which is the average number of tag IDs that the reader is able to collect in each second.

During the reading process, multiple tags may transmit their IDs simultaneously, causing collision. Some collision-avoidance methods such as FDMA or CDMA require sophisticated scheduling methods to minimize bandwidth waste due to idle sub-channels or unused codes [18]. The overhead for sophisticated scheduling can be too costly for a RFID system where each tag only needs to deliver one piece of information (i.e., its ID) to the reader. Hence, contention-based time-slotted protocols have become the industrial standards [19].

In a contention-based protocol, each tag transmits its ID in a time slot with a report probability p that is tuned to reduce collision. A tag stops when it receives the acknowledgement from the reader that its ID has been successfully received. It can be shown that the optimal reading throughput is theoretically bounded by $\frac{1}{eT}$, where e is the natural constant and T is the length of a time slot [11]. In such a protocol, 36.8% of the time slots will be idle and 26.4% of the slots will have collision.

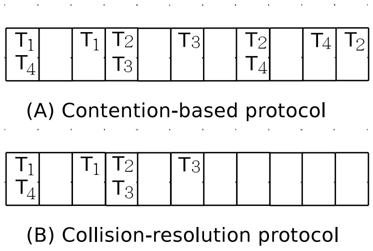
T1
T4
T1
T2
T3
T3
T2
T4
T4
T2
(A) Contention-based protocol
T1
T4
T1
T2
T3
T3
(B) Collision-resolution protocol

Fig. 1. This example shows that a collision-resolution protocol may reduce the number of time slots used to identify four tags from 11 time slots to 6 time slots.

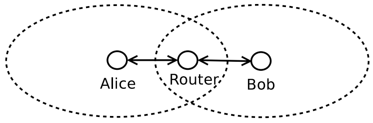

Fig. 2. Alice-Bob example for Analog Network Coding.

Can we do better than $\frac{1}{eT}$? We observe that the reading throughput can be improved if we make good use of the collision slots. Suppose the reader receives a mixed signal in a collision slot when both tag t_1 and tag t_2 transmit their IDs. In a later slot, if the reader receives the individual signal for the ID of tag t_1, it can remove this signal from the mixed signal and recover the individual signal for the ID of tag t_2.

Consider the example in Fig. 1, where four tags transmit their IDs to the reader. In Fig. 1 (a), when a contention-based protocol is used, it takes 11 slots for the reader to collect all four IDs. In Fig. 1 (b), when a collision-resolution protocol is used to resolve collision, only 6 slots are necessary. In particular, when the reader receives the signal from t_1 in the third slot, it removes this signal from the mixed signal received in the first slot and recovers the ID of t_4. Similarly, when it receives the signal from t_3 in the sixth slot, it also learns the ID of t_2 from the fourth slot.

B. Analog Network Coding (ANC)

Can we remove an individual signal from a mixed signal to recover the other constituent signal? This question has recently been brought up in the context of physical-layer networking code. Significant progress has been made in both theory and implementation [1], [17].

Katti *et al.* implemented the Analog Network Coding (ANC) and demonstrated its effectiveness in the Alice-Bob network shown in Fig. 2. Traditionally, four timeslots are needed for Alice and Bob to exchange a pair of messages: Alice sends a message to the router and the router forwards it to Bob, and vice versa. However, by using ANC, only two timeslots are necessary: Alice and Bob transmit simultaneously to the router. Instead of dropping the mixed signal, the router simply amplifies and broadcasts it to both Alice and Bob. Alice subtracts her own signal from the mixed signal and decodes Bob's message. Similarly, Bob can extract Alice's message.

We briefly describe the method used by Katti *et al.* Readers are referred to [1] for more details. The ANC protocol is designed based on MSK (Minimum Shift Keying) [20] and has been implemented using software defined radios. The signal transmitted by Alice can be represented as

$$s[n] = A_s e^{i\theta_s[n]},$$

where A_s is the amplitude of the n^{th} sample and $\theta_s[n]$ is its phase. Similarly, Bob's signal can be represented as

$$s[n] = B_s e^{i\phi_s[n]}.$$

If Alice and Bob transmit simultaneously, the router will receive the sum of the two signals, which can be represented as

$$y[n] = h' A_s e^{i(\theta_s[n]+\gamma')} + h'' B_s e^{i(\phi_s[n]+\gamma'')},$$

where h' and h'' are the channel attenuation and γ' and γ'' are the phase shift. We rewrite it for simplicity as

$$y[n] = Ae^{i\theta[n]} + Be^{i\phi[n]}, \tag{1}$$

where $A = h'A_s$, $B = h''B_s$, $\theta[n] = \theta_s[n] + \gamma'$, and $\phi[n] = \phi_s[n] + \gamma''$. Upon receiving the mixed signal from the router, Alice can resolve A and B from the following two energy equations [21], [1],

$$\mu = E[|y[n]|^2] = A^2 + B^2,$$

$$\sigma = \frac{2}{W} \sum_{|y[n]|^2 > \mu} |y[n]|^2 = A^2 + B^2 + 4AB/\pi,$$

where $E[.]$ is the expectation and W is a sampling window size. In MSK, a bit '1' is represented as a phase difference of $\pi/2$ over a time interval t, whereas a bit '0' is represented as a phase difference of $-\pi/2$ over t. For example, if the phase difference between the $(n+1)^{th}$ sample and the n^{th} sample, $\theta[n+1] - \theta[n]$, is $\pi/2$, then a bit "1" is transmitted. Since Alice knows her own signal, from (1), she can estimate the phase differences of Bob's signal, $\phi[n+1] - \phi[n]$, which can be translated into the bit stream sent by Bob [1].

The task of resolving the mixed signal in a collision slot in a RFID system is simpler than the same task in the wireless network shown in Fig. 2. First, Alice knows the amplitude of her signal when it is transmitted out, but she does not know the amplitude of her signal when it reaches the router and mixed with Bob's signal. When Alice received the amplified mixed signal from the router, it becomes difficult for her to remove her own signal from the mixed one. In the RFID system, suppose the reader receives the mixed signal from t_1 and t_2 in one slot and the individual signal of t_1 in another slot. Because the same signal of t_1 appears in the two slots, it becomes easier to remove it from the mixed signal.

Second, it is very difficult to synchronize transmissions between wireless nodes, and thus the proposed ANC protocol has to introduce a complicated mechanism to relieve this problem, whereas transmissions in a RFID system can be synchronized by the reader's signal.

Given that the technology for collision resolution exists, the next question is how to optimally use it to maximize the performance of a wireless system. This paper will answer the question in the context of RFID systems.

III. Terminology and Problem Definition

A. Terminology

During the execution of a time-slotted contention-based protocol, if no tag transmits in a time slot, we call it an *empty slot*. If one tag transmits, it is called a *singleton slot*. If more than one tag transmits, it is a *collision slot*. In particular, if k tags transmit simultaneously, the slot is called a *k-collision slot*, where $k \geq 2$. In order to guard against channel error, each ID carries a CRC code. In a singleton slot, the RFID reader receives the ID signal from a single tag. It will verify the correctness of the received ID by checking the CRC code.

B. Resolvable Collision Slots

An empty slot is easy to identify because no signal is received. The reader can distinguish a singleton slot from a collision slot by first converting the signal into an ID and then verifying the correctness of the CRC code. For a collision slot, the reader records a *mixed signal* that combines the individual signals of the tags that transmit simultaneously. In later singleton slots, the reader will receive the individual ID signals from some of those tags. When the reader eventually receives the ID signals from all but one of those tags, we say the k-collision slot is *resolvable* if we can derive the ID signal of the last tag by removing the $(k-1)$ ID signals from the mixed signal. The experimental study of Analogy Network Coding by Katti *et al.* in [1] has shown that 2-collision slots are resolvable.

C. Problem Definition

The main problem we want to solve in this paper is how to optimally apply analog network coding to maximize the RFID reading throughput. We design a collision-aware tag identification protocol and derive the optimal report probability (at which a tag transmits its ID in each slot) that maximizes the number of singleton and 2-collision slots (from which IDs can be extracted by ANC).

In [1], the authors state that ANC can be applied to resolve collision involving more than two signals. On one hand, as we will demonstrate in Section VI, resolving 2-collision slots based on today's technology will already provide a practically significant boost to the reading throughout. On the other hand, instead of restricting our work to 2-collision slots, we decide to generalize our protocol so that it can work with any future ANC method that resolves λ-collision slots, where λ (≥ 2) is an input parameter. Such generalization sheds light on the amount of throughput gain that can possibly be obtained through analog network coding. In particular, the results in Section VI show that the reading throughput will be higher when λ is larger (because more collision slots become useful). However, the rate of throughput improvement diminishes quickly as λ increases. Hence, it is not necessary to make λ too large. In practice, we expect λ to be a small number (such as 2, 3 or 4).

Clearly, ANC and other physical-layer network coding methods can be applied in various different communication contexts, each of which has its unique technical challenges. For example, collision resolution has been used in satellite access networks, where each terminal transmits a single packet twice at two randomly-selected time slots in each MAC frame [22]. The throughput upper bound can be predicted if the number of packets per slot is known (which requires the knowledge of the number of transmitting terminals). In our context, we do not derive a throughput upper bound for a given set of system parameters. Instead, we determine the best system parameter that optimizes the throughput. We do not assume the knowledge for the number of tags that is participating in the protocol. In fact, this number changes over time because after a tag successfully delivers its ID to the reader, it will stop transmitting. A tag may transmit for one, two or more times at any time slots during the reading process. Moreover, because the number of participating tags changes, the optimal system parameter also changes over time.

IV. Slotted Collision-Aware Tag Identification Protocol

In this section, we propose the Slotted Collision-Aware Tag identification protocol (SCAT). In the next section, we will optimize the protocol for less overhead.

A. Protocol Description

SCAT is a time-slotted protocol. The time slots are synchronized by the reader's signal. Each time slot consists of three segments: the advertisement segment, the report segment, and the acknowledgement segment.

In the advertisement segment, the RFID reader sends out a *slot index* i and a *report probability* p_i, where i begins from zero and increases by one after each slot.

In the report segment, each tag decides to transmit its ID with probability p_i. To actually broadcast the report probability, the reader may send out an l-bit integer $\lfloor p_i \times 2^l \rfloor$ instead of a real number p_i. A tag computes a hash function $H(\text{ID}|i)$ whose range is $[0..2^l)$. If $H(\text{ID}|i) \leq \lfloor p_i \times 2^l \rfloor$, the tag transmits its ID.

For an empty slot, the reader transmits a negative acknowledgement. For a collision slot, the reader will not be able to tell how many tags have transmitted simultaneously in the report segment. It will record the mixed signal and transmit a negative acknowledgement. The mixed signal and the slot index form a *collision record*. Over time the reader will collect a group of such records. The operation for a singleton slot is more complicated. The reader learns the ID of a tag in the report segment. Using this ID, it tries to resolve some collision records to learn more tag IDs (see the next subsection). It then transmits a positive acknowledgement, together with the IDs that are learned from the resolution of the previous collision records.

When the tag that transmits in the report segment receives the positive acknowledgement, it will stop participating in the SCAT protocol as its ID has been successfully delivered to the reader. Similarly, when a tag that transmitted its ID at an earlier slot but has not received a positive acknowledgement yet receives its own ID in the acknowledgement segment, it will stop participating in SCAT.

The SCAT protocol stops when no tag transmits any more. When the reader finds a certain number of consecutive empty slots, it sets $p_i = 1$ for one slot and if it still finds an empty slot, it knows that the IDs of all tags have been collected.

B. Collision Resolution

When the CRC received in the report segment is verified to be correct, the reader learns the ID of a tag from the current slot i. Knowing the ID, the RFID reader can easily figure out the previous slots in which this tag has transmitted. For an arbitrary collision record with slot index j, the tag must have transmitted if $H(\text{ID}|j) \leq \lfloor p_i \times 2^l \rfloor$. If that is the case, the reader removes the signal received in the current slot from the mixed signal in the collision record, treats the result as if it was the ID signal of a single tag, and extracts the CRC code. If the CRC code is verified to be correct, the collision record is resolved and the reader learns an additional tag ID. The signal for that ID can be used to resolve other collision records in a similar process as described above.

Resolving the collision slots incurs computation overhead. Hence, we expect the reader to be computationally capable or connected to a powerful computing device. It is worth noting that the RFID system works in a low speed channel (53 Kbps for the Philips I-Code system), while the original ANC [1] and the follow-up work [23] are designed for 11 Mbps or higher throughput channels, which is far more demanding, yet experimentally-demonstrated feasible.

C. Determining the Optimal Value for p_i

We want to *determine the optimal report probability p_i for each slot such that the number of slots for collecting the IDs of all tags is minimized*. Consider an arbitrary time slot with index i. When there is only one tag transmitting, the RFID reader will learn the ID of the tag. If there are two tags transmitting, the reader will not learn any ID now but will learn one ID later when the other ID is known (such that the collision record of this slot can be resolved). Similarly, when k tags transmit in this slot for $k \leq \lambda$, the reader will learn one ID from the collision record when the other $(k-1)$ IDs are known. Essentially, a singleton slot or a k-collision slot will allow the reader to learn one ID now or later. Hence, we shall choose the value of p_i that maximizes the probability for one, two, ..., or λ tags to transmit in the current slot.

Let N be the number of tags in the system. Its value can be estimated to an arbitrary accuracy [24] in a pre-step of SCAT. This pre-step will be removed in the next section. Before slot i, the reader may have successfully collected and acknowledged a number n_i of tag IDs, and those tags will no longer participate in the protocol of SCAT. Let N_i be the number of tags that the reader has not identified yet before slot i. Since n_i is known to the reader, N_i is also known.

As each tag decides to transmit with probability p_i, the number of tags that transmit will be a random variable X_i that follows the binomial distribution. The probability for $X_i = k$, $\forall k \in [0..N_i]$ is $\binom{N_i}{k} \cdot p_i^k (1-p_i)^{N_i-k}$. Our objective is to maximize the probability of $X_i \in [0..\lambda]$, which is

$$\sum_{k=1}^{\lambda} Prob\{X_i = k\} = \sum_{k=1}^{\lambda} \binom{N_i}{k} \cdot p_i^k (1-p_i)^{N_i-k}. \quad (2)$$

We expect λ to be small. In the following, we consider $\lambda = 2$, 3, or 4. When $\lambda = 2$, (2) becomes

$$\sum_{k=1}^{2} Prob\{X_i = k\} = N_i p_i (1-p_i)^{N_i-1} + \frac{N_i(N_i-1)}{2} p_i^2 (1-p_i)^{N_i-2} \simeq N_i p_i e^{-N_i p_i} + \frac{N_i^2 p_i^2}{2} e^{-N_i p_i}. \tag{3}$$

Let $\omega = N_i p_i$. Substituting $N_i p_i$ by ω in (3), we have

$$\sum_{k=1}^{2} Prob\{X_i = k\} \simeq (\omega + \frac{\omega^2}{2}) e^{-\omega}. \tag{4}$$

To find the value of ω that maximizes the above formula, we differentiate the right side and let it be zero.

$$\frac{d(\omega + \frac{\omega^2}{2})e^{-\omega}}{d\omega} = (1 - \frac{\omega^2}{2})e^{-\omega} = 0. \tag{5}$$

Solving the above equation, we have $\omega = 1.414$. Hence, the optimal report probability is $p_i = 1.414/N_i$.

Following the same process, we derive that, when $\lambda = 3$, the optimal report probability is $p_i = 1.817/N_i$, and when $\lambda = 4$, it is $p_i = 2.213/N_i$.

Resolving the collision slots requires a sufficient number of singleton slots. Otherwise, if all slots have collision, none of them will be resolved. Fortunately, when λ is small (which should be the case as we have discussed in Section III-C and will further elaborate in Section VI-A), there are sufficient singleton slots to resolve most collision slots. Our simulation results in Section VI-C show that the optimal report probabilities obtained by exhaustive search match closely with the above computed values.

D. Pseudo Code

The pseudo code for the operation of the RFID reader during the ith slot is given below. Let S be the set of newly known IDs (together with their signals) that can be used to resolve some of the collision records. Let I be the set of IDs that are learned by resolving the collision records. Let R_j be the collision record for slot j.

```
Reader's Operation at Slot $i$
broadcast an advertisement $\langle i, p_i \rangle$
record the signal $s_i$ in the report segment
extract $ID_i$ from $s_i$
**if** the channel is idle during the report segment **then**
    broadcast a negative acknowledgement
**else if** CRC in $ID_i$ is verified to be correct **then**
    $S := \{\langle ID_i, s_i \rangle\}$
    $I := \emptyset$
    **while** $S \neq \emptyset$ **do**
        remove an element $\langle ID, s \rangle$ from $S$
        **for** each collision record $R_j$ **do**
            **if** $H(ID|j) \leq p_j$ **then**
                add $s$ to the set of known individual signals in $R_j$
                remove known signals from the mixed signal in $R_j$
                extract $ID'$ from the resulting signal $s'$
                **if** CRC in $ID'$ is verified to be correct **then**
                    $S := S + \{\langle ID', s' \rangle\}$
                    $I := I + \{ID'\}$
                remove the collision record $R_j$
        **end for**
    **end while**
    broadcast a positive acknowledgement and the IDs in $I$
**else**
    add $\langle i, s_i \rangle$ as a collision record
    broadcast a negative acknowledgement
```

E. Unresolvable Collision Slots and Channel Error

The reading process normally takes a short period of time (minutes for tens of thousands of tags). During this time, we expect the tags to be statically located. The MSK employed by ANC can tolerate a certain level of noise and channel variation. However, if the spontaneous noise is too large, a collision slot may not be resolvable. The *only impact* is that the slot is not useful, and the reader can still learn the IDs from other slots. A tag will stop transmitting only after it receives positive confirmation from the reader. As long as most 2-collision slots can be resolved, the proposed protocol still achieves much higher reading throughput.

Channel error may corrupt the signal transmitted by a tag or the acknowledgement transmitted by the reader. This problem is common to all RFID reading protocols. The solution is also common: The tag will keep transmitting its ID until it receives the positive confirmation from the reader. In this case, the reader may receive an ID more than once and the duplicates will be discarded.

The proposed protocol is not suitable for an environment where the channel noise is so severe or the tags move so much and so fast during the reading operation that most collision slots are not resolvable. In this case, we should use a contention-based protocol without collision resolution. It is beyond the scope of this paper to investigate the noise level of each specific environment. Instead, we are more interested in knowing what is the upper limit of throughput improvement that ANC can bring (in an environment where most 2-collision slots are resolvable).

V. Framed Collision-Aware Tag Identification Protocol (FCAT)

In this section, we propose a framed version of the previous protocol to improve its efficiency.

A. Inefficiencies of SCAT

SCAT utilizes the information carried in the collision slots. However, it is not practically efficient due to a number of reasons.

First, to calculate p_i, the RFID reader has to know N_i, which in turn requires it to know N. It incurs considerable overhead to accurately estimate the number of tags in the system as a pre-step to SCAT. We want to remove such a pre-step and estimate N as a byproduct during the protocol execution.

Second, the advertisement segment of each slot represents significant overhead which is not always necessary. For consecutive slots, the slot index changes from i to $i+1$ and the report probability changes from ω/N_i to ω/N_{i+1}, where N_i and N_{i+1} at most differ by one. As the report probability changes little when N_i is reasonably large, the reader does not have to make advertisement in each slot. It may advertise once every certain number of slots, and the tags will use the same report probability in those slots.

Third, after resolving a collision record, the reader learns an extra ID and it broadcasts the ID in order to inform the corresponding tag to stop participating in the protocol. However, instead of transmitting the whole ID (which is 96 bits for GEN2 tags), the reader may transmit the slot index of the collision record. A tag stores the indices of the slots in which it has transmitted. If the tag receives a slot index that matches a stored one, it knows that the reader must have collected its ID. A slot index can be much smaller than 96 bits. If we use 23-bit slot indices, more than 8 million slots are allowed. In our simulations, the number of slots required never exceeds $2N$.

B. Using Frames

We propose the Framed Collision-Aware Tag identification protocol (FCAT), which improves SCAT by eliminating the inefficiency described in Section V-A. FCAT shares much of the protocol details with SCAT. Below we will focus on describing their differences.

In FCAT, time is divided into frames of size f. That is, each frame consists of f time slots. Each frame has an index, starting from zero. The index of the jth slot in the ith frame is $i \times f + j$. Before a frame begins, the RFID reader broadcasts a pre-frame advertisement, including the frame index i and the report probability p_i. Each slot of the frame consists of a report segment, during which the tags transmit their IDs, and an acknowledgement segment, during which the reader transmits either a positive acknowledgement or a negative acknowledgement.

In any slot of the ith frame, each tag transmits its ID with probability p_i. After receiving the signal in the report segment, the reader performs the same operations as in SCAT, except that it does not transmit the IDs learned from resolving the collision records in the acknowledgement segment. Instead, it transmits the slot indices of the resolved collision records, which are shorter than the IDs themselves. If a tag receives a slot index that matches a slot in which it has transmitted its ID, it stops participating in FCAT. Certainly, if a tag receives a positive acknowledgement for its ID just transmitted in the report segment, it will also stop participating in FCAT.

C. Estimating the Number of Tags within FCAT

Finally, we address the problem of how to learn the value of N. There exist efficient methods for estimating the number of tags. However, using them as a pre-step of FCAT incurs considerable overhead. In the following, we embed an estimation method within FCAT.

Consider an arbitrary frame with index i. Let n_0, n_1 and n_c be the random variables for the numbers of empty, singleton and collision slots, respectively. We can estimate the statistical relationship between these random variables and the number N_i of tags that are currently participating in the protocol. Based on that relationship, we can estimate N_i from the measured values of n_0 and n_c. Our approach shares some similarity with [24]. However, in [24], each tag transmits at most once in the frame. In FCAT, each tag participates probabilistically in every slot of the frame.

Let X_j be the indicator random variable for the event that the jth slot in the frame is empty, i.e., $X_j = 1$ means the jth slot is empty and $X_j = 0$ means it is not empty. Similarly, let Y_j be the indicator random variable for the event that the jth slot is a singleton slot. Because each tag decides to transmit with probability p_i in every slot in the frame, we have

$$Prob\{X_j = 1\} = (1-p_i)^{N_i}, \;\; \forall j \in [1..f]. \tag{6}$$

The expected value of n_0 is

$$E(n_0) = \sum_{j=1}^{f} (1-p_i)^{N_i} = f(1-p_i)^{N_i}. \tag{7}$$

The probability for the jth slot in the frame to be a singleton is

$$Prob\{Y_j = 1\} = \binom{N_i}{1} p_i (1-p_i)^{N_i-1} = N_i p_i (1-p_i)^{N_i-1}. \tag{8}$$

The expected value of n_1 is

$$E(n_1) = \sum_{i=1}^{f} N_i p_i (1-p_i)^{N_i-1} = f N_i p_i (1-p_i)^{N_i-1}. \tag{9}$$

Obviously, $E(n_0) + E(n_1) + E(n_c) = f$. Hence

$$\begin{aligned} E(n_c) &= f - E(n_0) - E(n_1) \\ &= f(1 - (1-p_i)^{N_i} - N_i p_i (1-p_i)^{N_i-1}) \\ &= f(1 - (1-p_i)^{N_i-1}(1 - p_i + \omega)). \end{aligned} \tag{10}$$

The above equation can be rewritten as

$$N_i = \frac{\ln(1 - \frac{E(n_c)}{f}) - \ln(1 - p_i + \omega)}{\ln(1-p_i)} + 1. \tag{11}$$

At the end of the ith frame, the reader counts the value of n_c. Substituting $E(n_c)$ by the instance value n_c (obtained in the ith frame), the reader obtains an estimation of N_i by the following formula:

$$\hat{N_i} = \frac{\ln(1 - \frac{n_c}{f}) - \ln(1 - p_i + \omega)}{\ln(1-p_i)} + 1. \tag{12}$$

Next, we derive $E(\hat{N_i})$. To simplify the equations, let $C_1 = \frac{1}{\ln(1-p_i)}$, $C_2 = -\frac{\ln(1-p_i+\omega)}{\ln(1-p_i)} + 1$, and function $g(n_c) = \ln(1 - \frac{n_c}{f})$. We expand the right hand side of (12) by its Taylor series about $q = E(n_c)$.

$$\begin{aligned} \hat{N_i} = C_1 \Big[& g(q) + (n_c - q)g'(q) + \frac{1}{2}(n_c - q)^2 g''(q) \\ & + \frac{1}{6}(n_c - q)^3 g'''(q) + ... \Big] + C_2. \end{aligned} \tag{13}$$

Since $q = E(n_c)$, the mean of the second term in (13) is 0. Therefore, we keep the first three terms when computing the approximated value of $E(\hat{N_i})$.

$$E(\hat{N_i}) \simeq C_1 \Big[g(q) + \frac{1}{2} E((n_c - q)^2) g''(q) \Big] + C_2. \tag{14}$$

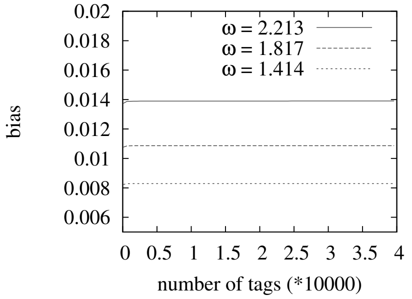

Fig. 3. The relative bias of $\hat{N}_i$ with respect to the number of tags.

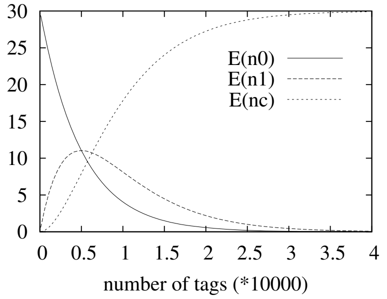

Fig. 4. The number of tags, N_i, is not a monotonic function in $E(n_1)$. Parameters: $p_i = 1.414/N_i$ and $f = 30$.

We have $E((n_c - q)^2) = V(n_c)$ by definition and $g''(q) = -\frac{1}{(q-f)^2}$ since $g(q) = \ln(1 - \frac{q}{f})$. The variance $V(n_c)$ is derived in the appendix. Applying (19) from the appendix, we have

$$E(\hat{N}_i) \simeq N_i - \frac{e^{\omega} - 1 - \omega}{2f\ln(1-p_i)(1+\omega)}. \quad (15)$$

Therefore,

$$Bias(\frac{\hat{N}_i}{N_i}) = E(\frac{\hat{N}_i}{N_i}) - 1 = \frac{1 + \omega - e^{\omega}}{2fN_i\ln(1-p_i)(1+\omega)}. \quad (16)$$

Fig. 3 shows the absolute value of $Bias(\frac{\hat{N}_i}{N_i})$ with respect to the number of tags N_i. The three lines show that the absolute values of $Bias(\frac{\hat{N}_i}{N_i})$ are $0.0082, 0.011$ and 0.014, for $\omega = 1.414, 1.817$ and 2.213, respectively. They are all very small.

Adding the number of tags whose IDs are already known, the reader has an estimation for the total number of tags in the system, denoted as $\hat{N}_i^*$. The variance of $\hat{N}_i^*$ is the same as the variance of $\hat{N}_i$, i.e., $V(\hat{N}_i^*) = V(\hat{N}_i)$. Because $N_i < N$, $V(\frac{\hat{N}_i^*}{N}) < V(\frac{\hat{N}_i}{N_i})$. The value of $V(\frac{\hat{N}_i}{N_i})$ is derived in the appendix. It is approximately 0.0342, 0.0287 or 0.0265, for $\omega = 1.414, 1.817$ and 2.213, respectively (i.e., when 2-collision slots, 3-collision slots or 4-collision slots are resolvable). This is the variance when only one instance of n_c is used. It is small though not negligible. The RFID reader obtains one estimation after each frame. If it uses the average $\frac{\sum_{j=0}^{i}\hat{N}_j^*}{i}$ as the estimation for N, then the variance will decrease in the square root of i and therefore diminish as the protocol executes frame after frame.

We can also design a similar estimator by using the number of empty slots, n_0, based on (7). However, we find in our simulations that the variance of such an estimator is larger. As shown in Fig. 4, N_i is not a monotonic function with respect to the number of singleton slots. Hence, we cannot use n_1 to estimate the value of N_i.

VI. Simulation Results

In this section, we present simulation results to evaluate the performance of our main protocol FCAT. We compare FCAT with the existing work, including the Dynamic Framed Slotted ALOHA (DFSA) [6], Enhanced Dynamic Framed Slotted ALOHA (EDFSA) [5], Adaptive Binary Splitting (ABS) [12] and Adaptive Query Splitting (AQS) [12]. The first two are ALOHA-based and the next two are tree-based.

We use FCAT-λ to denote the FCAT protocol in which k-collision slots with $k \leq \lambda$ are resolvable, where $\lambda = 2, 3, 4$. The report probability p_i is determined based on the formula given in Section IV-C. Specifically, p_i is set to be $1.414/N_i$, $1.817/N_i$ and $2.213/N_i$ in FCAT-2, FCAT-3 and FCAT-4, respectively. Other values of p_i are also investigated in Section VI-C. The frame size f is set to 30 time slots; the performance of FCAT under different f values will also be studied. The parameters used in other protocols are selected based on their original papers whenever possible.

In the simulations, we set the time slot length based on the Philips I-Code specification [25]. The transmission rate is 53 kbit/sec. Hence, it takes 18.88 μs to transmit each bit. We set the ID length to be 96 bits (including the 16 bits CRC code), which takes 1812 μs. The reader's acknowledgement consists of 20 bits, (including the CRC code), which takes 378 μs. The waiting time before the report segment or the acknowledgement segment is 302 μs to separate transmissions. Therefore, each slot is about 2.8 ms. The simulation results are the average outcome of 100 runs.

A. Reading Throughput Comparison

We first compare the protocols in terms of the reading throughput, which is the average number of tag IDs that the RFID reader can collect in each second during the protocol execution time before all IDs are read. Table I shows the reading throughputs of the protocols when the number of tags varies from 1,000 to 20,000. Due to collision resolution, FCAT-2 achieves 51.1% $\sim$ 55.6% throughput improvement over DFSA, 54.8% $\sim$ 70.6% improvement over EDFSA, 59.6% $\sim$ 62.9% improvement over ABS, 64.1% $\sim$ 67.7% improvement over AQS.

As expected, FCAT-3 performs better than FCAT-2, and FCAT-4 performs better than FCAT-3. However, the improvement of FCAT-4 over FCAT-3 is much smaller than that of FCAT-3 over FCAT-2. FCAT-5 (whose results are not shown in the table) performs only slightly better FCAT-4. For example, when $N = 10,000$, its reading throughput is 270.9 tag IDs per second, which is slightly better than 265.1 of FCAT-4. This indicates a quickly shrinking margin of improvement as λ increases and suggests that a large value of λ is practically unnecessary.

We also evaluate the reading time in terms of time slots. Table II shows the numbers of empty, singleton and collision slots used to read 10,000 tags. We can see that fewer empty

TABLE I
READING THROUGHPUT COMPARISON WHEN N VARIES FROM 1,000 TO 20,000

N	FCAT-2	FCAT-3	FCAT-4	DFSA	EDFSA	ABS	AQS
1000	197.7	234.8	238.8	130.8	115.9	123.9	117.9
2000	199.5	237.2	257.5	131.8	121.5	123.7	119.4
3000	200.2	239.7	261.4	132.1	122.9	123.8	120.4
4000	201.0	240.1	262.1	132.8	124.8	123.9	120.5
5000	201.3	240.4	262.3	130.1	126.1	123.8	120.8
6000	201.3	241.5	263.7	132.4	126.3	123.6	120.9
7000	201.3	241.2	264.9	131.1	126.4	123.8	121.1
8000	201.4	241.8	265.1	131.9	127.1	123.6	121.1
9000	201.2	241.5	265.4	131.0	127.8	123.7	121.1
10000	201.3	241.8	265.1	131.4	127.8	123.9	121.2
11000	201.7	241.5	266.0	130.0	127.6	123.9	121.1
12000	200.8	241.8	265.9	130.3	126.8	123.8	121.2
13000	201.0	241.7	265.9	129.2	127.3	123.8	121.2
14000	200.4	241.3	266.2	130.9	127.6	123.5	121.3
15000	200.8	241.2	266.0	131.7	127.7	124.2	121.3
16000	200.9	241.8	265.9	131.3	128.2	123.8	121.3
17000	200.2	241.3	265.5	130.5	128.1	124.1	121.3
18000	199.7	240.7	265.9	130.0	128.2	123.6	121.3
19000	199.1	240.9	266.4	129.2	128.2	123.7	121.3
20000	199.1	241.3	266.1	129.1	128.6	123.9	121.3

TABLE II
EMPTY, SINGLETON AND COLLISION TIME SLOTS WHEN N = 10000

	FCAT-2	FCAT-3	FCAT-4	DFSA	EDFSA	ABS	AQS
empty	4189	2257	1345	10076	10705	4410	4737
singleton	5861	4055	2935	10000	10000	10000	10000
collision	7016	7497	8050	7208	7234	14409	14735
total	17066	13809	12330	27284	27939	28819	29472

slots are wasted in FCAT than in all other compared protocols. FCAT also uses much fewer singleton slots to collect all tag IDs because FCAT can extract tag information from the collision slots, while other protocols have to read tags solely in the singleton slots. FCAT-4 has more collision slots than FCAT-2. The reason is that FCAT-4 can utilize a collision slot in which up to four tags collide, and hence FCAT-4 encourages more tags to transmit simultaneously.

B. Effectiveness of Collision Resolution

In Table III, we show the number of tag IDs that are resolved from the collision slots. FCAT-2 obtains about 40% of tag IDs from the collision slots. The percentage is above 57% for FCAT-3 and above 68% for FCAT-4. For example, when there are 10,000 tags in the system, FCAT-2 will read more than 4,000 of them from the collision slots, which are ignored by the previous protocols.

C. Report Probability

The report probability p_i is calculated as ω/N_i. N_i is the number of tags participating in slot i and the method in Section V-C is used to estimate N_i after each frame. The optimal value of ω is set in Section IV-C. We use simulation to confirm our analytical result and demonstrate how the value of ω affects the performance of FCAT. Fig. 5 shows the reading throughput with respect to ω when there are 10000 tags. If ω is set too small, the reading throughput decreases because many slots are empty and thus wasted. If ω is set too large, it also hurts the performance because there are too many collision slots and too many tags collide in those slots, making them unresolvable.

TABLE III
TAG IDS RESOLVED FROM COLLISION SLOTS

N	FCAT-2	FCAT-3	FCAT-4
1000	423	600	707
5000	2102	3008	3561
10000	4139	5945	7065
15000	6062	8819	10482
20000	7905	11507	13656

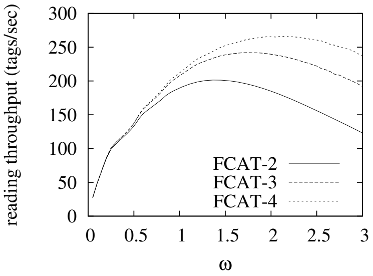

Fig. 5. FCAT reading throughput with respect to ω.

By trying all possible values of ω, we can use simulation to find the true optimal ω (and the corresponding optimal report probability) that maximizes the reading throughput. As shown in Table IV, the optimal value of ω observed in the simulation matches closely with the value computed in Section IV-C, i.e., 1.414 when $\lambda = 2$, 1.817 when $\lambda = 3$, and 2.213 when $\lambda = 4$. Also shown in the same table, the reading throughput achieved by FCAT using the computed reporting probability is almost the same as the maximum-achievable throughput under the optimal reporting probability obtained by simulation through exhaustive search.

D. Impact of Frame Size

Fig. 6 shows the impact of the frame size f in a system with 10,000 tags. We can see that the reading throughput is stabilized when $f \geq 10$.

VII. RELATED WORK

All existing contention-based tag reading protocols are called anti-collision protocols because they treat collision as waste and try to avoid it [26]. Most of these protocols fall into two classes: the ALOHA-based protocols [4], [5], [6], [7], [8],

TABLE IV
THE COMPUTED VALUE OF ω MATCHES CLOSELY WITH THE OPTIMAL VALUE OF ω OBTAINED BY SIMULATION.

λ	Optimal ω	Maximum Throughput	computed ω	FCAT Throughput
2	1.42	202.1	1.41	201.3
3	1.90	241.9	1.82	241.8
4	2.12	266.2	2.21	265.1

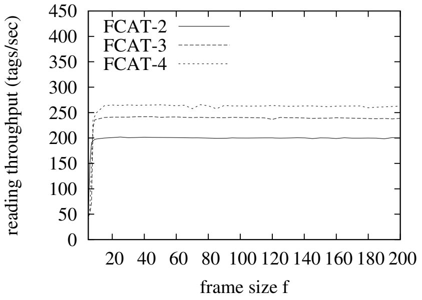

Fig. 6. The reading throughput of FCAT is stabilized when $f \geq 10$.

[9], [10] and the tree-based protocols [12], [13], [14], [15], [16].

In the ALOHA-based protocols, the reader broadcasts a request and each tag randomly selects a time slot to report its ID. If exact one tag reports, the reader retrieves its tag ID and this tag will remain silent for the rest of reading process. Simultaneous reports in a slot will lead to collision. Therefore, the ALOHA-based protocols try to maximize the probability that exact one tag reports in a slot. The ALOHA-based protocols differ in how the reader sends the request and how the tag selects a slot to report. In the *slotted* ALOHA [4], the reader sends out a contention probability at the beginning of each slot and each unread tag with this probability to reply with its ID. In the *basic framed slotted* ALOHA [5], slots are grouped into frames with the same fixed frame size. Each unread tag picks up a random slot within each frame to report. It is possible that the number of tags far exceeds the number of slots in a frame so that the frame is full of collision. To overcome this problem, the *dynamic framed slotted* ALOHA (DFSA) [6] introduces frames with dynamic frame size. It is proved that the maximal reading throughput is achieved when the frame size is equal to the number of unread tags [6]. DFSA determines the size of the next frame by estimating the number of unread tags after each frame. However, in practice, it may be impractical to set the frame size indefinitely high considering there exist a large number of tags [5]. The *enhanced dynamic framed slotted* ALOHA [5] uses frames with limited frame size by restricting the number of responding tags in a frame. The maximal reading throughput of the ALOHA-based protocols is bounded by $\frac{1}{eT}$ [11]. In other words, for each slot, the probability of successfully reading a new tag is 36.8%.

In the tree-based protocols, the tag reading procedure can be interpreted as a recursive splitting procedure. The general schema works as follows: In a slot, the reader sends a query with a certain condition and each tag that meets the condition will respond. If a set of tags respond concurrently, the reader split them into smaller subsets. The procedure repeats until every subset only contains a single tag which can be identified by the reader. Different splitting criteria lead to different protocols. The *binary-tree protocols* [12], [15], [16] split a set of tags using a random binary number. Specifically, each tag has a counter initialized to 0. Upon receiving a query, each tag that has a counter value 0 will respond. Once collision happens, the reader sends a new query with an indication of the collision. Each colliding tag draws a random binary number (i.e. 0 or 1) and adds it to its counter. The set of colliding tags is thus divided into two subsets: one is the set of tags whose counters remain 0 and the other one is the set of tags whose counters increase to 1. When collision happens, all other tags that do not transmit also increase their counters by one; otherwise, they decrease their counters by one. An analysis shows that the maximal reading throughput of the binary-tree protocols is $\frac{1}{2.88T}$ [27]. The *query-tree protocols* [12], [13], [14] use the tag ID for splitting. A tag ID is a unique bit string. Each query contains a prefix $p_1..p_i$ where i is the length of the prefix. Each tag, whose ID contains this prefix, transmits its ID as a response. If multiple responses collide, the reader will generate two new prefixes $p_1..p_i0$ and $p_1..p_i1$ by attaching a bit 0 and 1, respectively. The set of colliding tags is divided into two subsets: one subset is the group of tags whose IDs contain the prefix $p_1..p_i0$ and the other subset is the group of tags whose IDs contain the prefix $p_1..p_i1$. A query-tree protocol can have quite different reading throughputs determined by the tag ID distribution. It is shown that the maximal reading throughput is bounded by $\frac{1}{2.88T}$ for a set of uniformly distributed tag IDs [28].

VIII. Conclusion

We believe this is the first paper that applies physical-layer network coding to help boost the reading throughput of a large RFID system. We conclude that the physical-layer network coding can indeed significantly improve the speed at which a RFID reader collects information of the tags. The reason is that the information carried in many collision slots, which was previously discarded, can be utilized almost as effectively as the information carried in the singleton slots. The current analog network coding method can improve the reading throughput of a RFID system by $51.1\% \sim 70.6\%$. As the technologies of physical-layer network coding are improved, the reading throughput can potentially be doubled.

References

[1] S. Katti, S. Gollakota, and D. Katabi, "Embracing Wireless Interference: Analog Network Coding," In *Proc. of ACM SIGCOMM*, August 2007.

[2] R. Das, "Global RFID Market Tops $5.5 Billion," *http://www.convertingmagazine.com/article/CA6653688.html*, 2009.

[3] G. Zecca, P. Couderc, M. Banatre, and R. Beraldi, "Swarm Robot Synchronization Using RFID Tags," In *Proc. of IEEE PerCom*, March 2009.

[4] I. E. Telatar and R. G. Gallager, "Combining Queuing Theory and Information Theory for Multiaccess," *IEEE Journal on Selected Areas Communication*, vol. 13, no. 6, pp. 963–969, August 1995.

[5] S. R. Lee, S. D. Joo, and C. W. Lee, "An Enhanced Dynamic Framed Slotted ALOHA Algorithm for RFID Tag Identification," *IEEE International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS)*, July 2005.

[6] J. R. Cha and J. H. Kim, "Dynamic Framed Slotted ALOHA Algorithms Using Fast Tag Estimation Method for RFID Systems," *IEEE Consumer Communications and Networking Conference (CCNC)*, January 2006.

[7] H. Vogt, "Efficient Object Identification with Passive RFID Tags," In *Proc. of International Conference on Pervasive Computing*, April 2002.

[8] V. Sarangan, M. R. Devarapalli, and S. Radhakrishnan, "A Framework for Fast RFID Tag Reading in Static and Mobile Environments," *The International Journal of Computer and Telecommunications Networking*, vol. 52, no. 5, pp. 1058–1073, 2008.

[9] Q. Peng, M. Zhang, and W. Wu, "Variant Enhanced Dynamic Frame Slotted ALOHA Algorithm for Fast Object Identification in RFID System," *IEEE International Workshop on Anti-counterfeiting, Security, Identification*, April 2007.

[10] B. Zhen, M. Kobayashi, and M. Shimizu, "Framed ALOHA for Multiple RFID Object Identification," *IEICE Transactions on Communications*, March 2005.

[11] L. G. Roberts, "ALOHA Packet System with and without Slots and Capture," *ACM SIGCOMM Computer Communication Review*, vol. 5, no. 2, pp. 28–42, April 1975.

[12] J. Myung and W. Lee, "Adaptive Splitting Protocols for RFID Tag Collision Arbitration," In *Proc. of ACM MobiHoc*, May 2006.

[13] N. Bhandari, A. Sahoo, and S. Iyer, "Intelligent Query Tree (IQT) Protocol to Improve RFID Tag Read Efficiency," *IEEE International Conference on Information Technology (ICIT)*, December 2006.

[14] F. Zhou, C. Chen, D. Jin, C. Huang, and H. Min, "Evaluating and Optimizing Power Consumption of Anti-collision Protocols for Applications in RFID Systems," In *Proc. of ACM International Symposium on Low Power Electronics and Design (ISLPED)*, August 2004.

[15] "Information technology automatic identification and data capture techniques C radio frequency identification for item management air interface - part 6: parameters for air interface communications at 860-960 MHz," Final Draft International Standard ISO 18000-6, November 2003.

[16] S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels, "Security and Privacy Aspects of Low-cost Radio Frequency Identification Systems," In *Proc. of International Conference on Security in Pervasive Computing (SPC)*, March 2003.

[17] S. Zhang, S. Liew, and P. P. Lam, "Physical Layer Network Coding," In *Proc. of ACM MobiCom*, September 2006.

[18] D. Shih, P. L. Sun, D. C. Yen, and S. M. Huang, "Taxonomy and Survey of RFID Anti-collision Protocols," *Computer and Communications*, vol. 29, no. 11, pp. 2150–2166, 2006.

[19] M. C. Azambuja, C. A. M. Marcon, and F. P. Hessel, "Survey of Standardized ISO 18000-6 RFID Anti-collision Protocols," In *Proc. of IEEE International Conference on Sensor Technologies and Applications (SENSORCOMM)*, August 2008.

[20] S. Pasupathy, "Minimum Shift Keying: A Spectrally Efficient Modulation," *IEEE Communications Magazine*, 1979.

[21] J. Hamkins, "An Analytic Technique to Separate Cochannel FM Signals," *IEEE Transactions on Communications*, vol. 48, no. 4, pp. 543–546, April 2000.

[22] E. Casini, R. D. Gaudenzi, and O. R. Herrero, "Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Schemefor Satellite Access Packet Networks," *IEEE Transactions on Wireless Communications*, vol. 6, no. 4, pp. 1408–1419, April 2007.

[23] S. Gollakota and D. Katabi (MIT), "ZigZag Decoding: Combating Hidden Terminals in Wireless Networks," In *Proc. of ACM SIGCOMM*, August 2008.

[24] M. Kodialam and T. Nandagopal, "Fast and Reliable Estimation Schemes in RFID Systems," In *Proc. of ACM MobiCom*, September 2006.

[25] Philips Semiconductors, "I-CODE Smart Label RFID Tags," *http://www.nxp.com/acrobat_download/other/identification/SL092030.pdf*, January 2004.

[26] K. Finkenzeller, "RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification," John Wiley & Sons, 2003.

[27] J. I. Capetenakis, "Tree Algorithms for Packet Broadcast Channels," *IEEE Transactions on Information Theory*, vol. 25, no. 5, pp. 505–515, 1979.

[28] C. Law, K. Lee, and K. Y. Siu, "Efficient Memoryless Protocol for Tag Identification," In *Prof. of International Workshop on Discrete Algaorithms and Methods for Mobile Computing and Communications*, pp. 75–84, August 2000.

[29] G. Casella and R. L. Berger, "Statistical Inference," *2nd edition, Duxbury Press*, 2002.

Appendix. Estimation Variance, $V(\frac{\hat{N}_i}{N_i})$

Consider an arbitrary frame with index i. Let Z_j be the indicator random variable for the event that the jth slot in the frame is a collision slot. Since no slot is special, Z_j, $\forall j \in [1..f]$, follows the same distribution. They are independent random variables. Because $n_c = \sum_{j=1}^{f} Z_j$, we have

$$V(n_c) = \sum_{j=1}^{f} V(Z_j) = fV(Z_1). \tag{17}$$

$E(Z_1) = 1 - (1-p_i)^{N_i} - N_i p_i (1-p_i)^{N_i - 1} \approx 1 - e^{-N_i p_i} - N_i p_i \cdot e^{-N_i p_i}$. $E(Z_1^2) = E(Z_1)$ because Z_1 is an indicator random variable. Hence, we have

$$\begin{aligned} V(Z_1) &= E(Z_1^2) - (E(Z_1))^2 \\ &= (1 + N_i p_i) e^{-N_i p_i} (1 - (1 + N_i p_i) e^{-N_i p_i}). \end{aligned} \tag{18}$$

Therefore,

$$V(n_c) = f(1 + N_i p_i) e^{-N_i p_i} (1 - (1 + N_i p_i) e^{-N_i p_i}). \tag{19}$$

According to the central limit theorem, if f is large, n_c is approximately normally distributed. When $f \to \infty$, n_c converges to the normal distribution,

$$n_c \overset{D}{\to} Norm(\theta, \delta^2)$$

where θ is $E(n_c)$ as given in (10), δ^2 is $V(n_c)$ as given in (19), and $\overset{D}{\to}$ means convergence in distribution.

According to the δ-method [29], we have

$$h(n_c) \overset{D}{\to} Norm(h(\theta), \delta^2 \left[h'(\theta)\right]^2) \tag{20}$$

for any function $h(.)$ such that $h'(\theta)$ exists and takes a nonzero value.

In Section V-C, the estimation formula is designed based on (10), which is copied below.

$$E(n_c) = f(1 - (1-p_i)^{N_i} - N_i p_i (1-p_i)^{N_i - 1}).$$

Let $g(.)$ be the mapping function from N_i to n_c. The above equation can be rewritten as $E(n_c) = g(N_i)$. Fig. 4 shows that $g(.)$ is a monotonic function, and hence it has a unique inverse function, denoted as $h(.)$.

According to Section V-C, $\hat{N}_i$ is computed from (10) by substituting $E(n_c)$ with the instance value of n_c (obtained after the ith frame).

$$\begin{aligned} n_c &= f(1 - (1-p_i)^{\hat{N}_i} - \hat{N}_i p_i (1-p_i)^{\hat{N}_i - 1}) \\ &\approx f(1 - e^{-\hat{N}_i p_i} - \hat{N}_i p_i e^{-\hat{N}_i p_i}). \end{aligned} \tag{21}$$

Clearly, $n_c = g(\hat{N}_i)$ and $\hat{N}_i = h(n_c)$. Applying $\hat{N}_i = h(n_c)$ to (20), we have

$$\hat{N}_i \overset{D}{\to} Norm(h(\theta), \delta^2 \left[h'(\theta)\right]^2). \tag{22}$$

We know that $h(g(N_i)) = N_i$. Differentiating both sides, we have $h'(g(N_i))g'(N_i) = 1$. Hence,

$$h'(\theta) = h'(E(n_c)) = h'(g(N_i)) = \frac{1}{g'(N_i)}. \tag{23}$$

Therefore, from (22), the variance of $\hat{N}_i$ is

$$\begin{aligned} V(\hat{N}_i) &= \delta^2 \left[h'(\theta)\right]^2 = \frac{V(n_c)}{[g'(N_i)]^2} \\ &= \frac{(1 + N_i p_i) e^{N_i p_i} - (1 + 2N_i p_i + N_i^2 p_i^2)}{f N_i^2 p_i^4}, \end{aligned} \tag{24}$$

$$V(\frac{\hat{N}_i}{N_i}) = \frac{(1 + N_i p_i) e^{N_i p_i} - (1 + 2N_i p_i + N_i^2 p_i^2)}{f N_i^4 p_i^4}. \tag{25}$$

Below we perform approximate computation to give a rough idea on how big this variance is. In SCAT or FCAT, $\hat{N}_i p_i = \omega$, where ω is 1.414, 1.817 or 2.213 for $\lambda = 2$, 3 or 4, respectively. Our simulations show that $\hat{N}_i$ reliably converges to N_i when i is large. Hence, we substitute $N_i p_i$ with ω in (25), and the variance $V(\frac{\hat{N}_i}{N_i})$ is 0.0342, 0.0287 or 0.0265 respectively for different ω values.