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 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, which can significantly improve the reading throughput when comparing with the existing ALOHA-based 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 merchandise item. It also has serious limitation. A barcode can only be read in close range. Suppose an inventory management policy requires the 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. 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 tags that can be read in a long distance (depending on the transmission power).

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. The tree-based protocols organize all IDs in a tree of ID prefixes [2], [3], [4]. Each in-tree prefix has two child nodes that have one additional bit, ‘0’ or ‘1’. The tag IDs are leaves of the tree. The RFID reader walks through the tree. As it reaches an in-tree node, it queries for tags with the prefix represented by the node. When multiple tags match the prefix, they will all respond and cause collision. Then the reader moves to a child node by extending the prefix with one more bit. If zero or one tag responds (in the one-tag case, the reader receives an ID), it moves up in the tree and follow the next branch. Another type of tree-based protocols tries to balance the tree by letting the tags randomly pick which branches they belong to [2], [5], [6], [7]. In the best case, when the tree is completely balanced, the reader takes two (or three, depending on the actual design) queries on average for discovering each ID. However, it will incur much more overhead for the average and worst cases.

The other major class is the time-slotted ALOHA-based protocols[8], [9], [10], [11], [12], [13], [14]. Each 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. It is achieved when $p$ is chosen such that the probability for exactly one tag transmitting in each slot is 36.8%.

To break the fundamental limit of the ALOHA-based protocols, we have to resort to fundamentally different approaches. In this paper, we apply a recently-emerged engineering principle, called physical-layer networking coding, 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 [15]. In particular, the analog network coding scheme [1] has been experimentally implemented. The problem is that the effectiveness of analogy network coding
(as well as other schemes of physical-layer network coding) has only been demonstrated under “toy” examples. No one has shown that it can be used in an important application under general setting.

Our contribution in this paper are two-fold: First, we show that analog network coding can be used to make a collision slot almost as useful as a non-collision slot (in which only one tag transmits). The difference is that the former allows the RFID reader to learn a new tag ID after some time, while the latter lets the reader learn a new ID right away. Second, we demonstrate the practical value of the physical-layer network coding research by providing an interesting application scenario to apply analog network coding.

Specifically, we first design a slotted collision-aware tag identification protocol (FCAT), which establishes the engineering and theoretical foundation for integrating analog network coding to the process of tag reading. We then propose a framed collision-aware tag identification protocol (SCAT) to reduce the overhead of SCAT. The protocol is able to efficiently utilize the information carried in collision slots and thus breaks the fundamental limit of ALOHA-based protocols that do not use physical-layer network coding. We show that the reading throughput can be improved more than 50% by using today’s analog network coding method and the throughput can be doubled if the coding method is improved in the future.

The rest of the paper is organized as follows. Section 2 gives the problem definition and motivation. Section 3 and 4 describe our protocols that use analog network coding to improve RFID reading throughput. Sections 5 presents the simulation results. Section 6 discusses the related work. Section 7 draws the conclusion.

II. PROBLEM DEFINITION AND MOTIVATION
A. Problem Definition
Consider a RFID system with a large number of 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 and try to optimize the reading throughput, which is the average number of new tag IDs that the reader is able to collect in each second.

B. Motivation
When a reader collects the tag IDs, multiple tags may transmit their IDs simultaneously, which causes collision. Collision is considered to be a waste of bandwidth, and multiplexing techniques have been extensively investigated to avoid it. The effectiveness of FDMA or CDMA relies on sophisticated scheduling methods that minimizes the waste due to idle sub-channels or unused codes [16]. Particularly for a RFID system where each tag only needs to deliver one piece of information (i.e., its ID) to the reader, the overhead for sophisticated scheduling can be too costly. Therefore, contention-based time-slotted protocols have become the industrial standards [17].

In a contention-based protocol, each tag transmits its ID in a time slot with a 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 $\frac{1}{p}$, where $T$ is the length of a time slot [18]. In such a protocol, 36.8% of the time slots will be idle and 26.4% of the slots will have collision.

Can we do better than $\frac{1}{p}$? We observe that the reading throughput can be improved if we turn the collision slots to be useful. Suppose the reader receives the 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 from tag $t_1$ that carries the ID of $t_1$, it can remove this signal from the mixed signal and recover the signal that $t_2$ transmits in the collision slot. Consequently, it is able to retrieve the ID of $t_2$ from the information in the collision slot.

Consider the example in Fig. 1, where four tags transmit their IDs to the reader. In Fig. 1 (a), when a contention protocol is used, it takes 11 slots for the reader to collect all four IDs. In Fig. 1 (b), when a collision-aware protocol is used to resolve collision if possible, 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 $t_1$. 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.

C. 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 wireless communication community in the context of physical-layer networking code. Significant progress has been made in both theory and implementation [15], [1]. 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 one message: Alice sends a message to the router and the router forwards it to Bob, and vice verse. However, by using ANC, only two timeslots

![Fig. 1. Analog network coding.](image-url)
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) [19] and has been implemented using Software Defined Radios (SDR). The signal transmitted by Alice can be represented as

\[ s[n] = Ae^{\theta[n]}, \]

where \( A \) is the amplitude of the \( n \)th sample and \( \theta[n] \) is its phase. Similarly, Bob’s signal can be represented as

\[ s[n] = Be^{\phi[n]}, \]

where \( B \) is the amplitude of the \( n \)th sample and \( \phi[n] \) is its phase. If Alice and Bob transmit simultaneously, the router will receive the sum of the two signals, which can be represented as

\[ y[n] = Ae^{\theta[n]} + Be^{\phi[n]}. \]  

(1)

Upon receiving the mixed signal from the router, Alice can resolve \( A \) and \( B \) from the following two energy equations

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

(2)

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

(3)

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 an 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 of Bob’s signal and obtains the phase difference \( \phi[n+1] - \phi[n] \), which can be translated into the bit stream sent by Bob.

In their paper [1], the authors noted that ANC can be applied to resolve collision involving more than two signals.

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 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.

III. SLOTTED COLLISION-AWARE TAG IDENTIFICATION PROTOCOL (SCAT)

In this section, we propose the Slotted Collision-Aware Tag identification protocol (SCAT), which establishes the theoretical foundation for our main protocol (FCAT) in the next section.

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.

1) Operations in the advertisement and report segments: 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 implement the report probability, the reader may send out an \( l \)-bit integer \( [p_i \times 2^l] \) instead of a real number \( p_i \). A tag computes a hash function \( H(ID|i) \), whose range is \( [0, 2^l] \), where \( ID \) is the tag’s identification number. If \( H(ID|i) \leq [p_i \times 2^l] \), the tag transmits its ID.

2) Resolvable collision slots: If no tag transmits, we call the time slot an empty slot. If one tag transmits, it is called a singleton slot. In this case, the reader receives the ID signal from the tag, and confirms the correctness of the received ID by verifying the CRC code carried in the ID. If more than one tag transmits in the report segment, the current time slot is called a collision slot. In particular, if \( k \) tags transmit, it is a \( k \)-collision slot, where \( k \geq 2 \). The RFID reader records a mixed signal that combines the individual signals of the participating tags as they transmit simultaneously. At 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 Analog Network Coding by Katti et al. in [1] have shown that 2-collision slots are resolvable. The same work also points out the feasibility of making 3-collision (or higher) slots resolvable. To make the design of our protocol general, we introduce a parameter \( \lambda \) and assume that \( k \)-collision slots with \( k \leq \lambda \) are resolvable. In practice, we expect \( \lambda \) to be a small number (such as 2, 3, or 4).

3) Operation in the acknowledgement segment: The reader can identify whether the current time slot is an empty, singleton, or collision slot. An empty slot is easy to identify because no signal is received. The reader can distinguish a singleton
slot form a collision slot by verifying the CRC code received in the report segment.

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. Each collision record also contains the set of individual ID signals that are received after the mixed signal.

The operation for a singleton slot is more complicated. The reader learns the ID of a tag in the report segment. Knowing the ID, the reader can easily figure out the previous slots in which this tag has transmitted. For an arbitrary previous slot with index \( i \), the tag must have transmitted if \( H(ID|i) \leq \lfloor p_i \times 2^X \rfloor \). If that is the case, the reader checks if there is a collision record for slot \( i \). If so, it adds the ID signal received in the report segment to the record. It then removes all individual ID signals stored in the record from the mixed signal, treats the result as if it was the ID signal of a single tag, and extracts the CRC code. If the verification is successful, we say 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.

Therefore, a singleton slot can result in the discovery of one, two or more tag IDs. The reader will transmit 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 receives its own ID in the acknowledgement segment, it will stop participating in SCAT.

4) Termination: 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. Pseudo Code

The pseudo code for the operation of the RFID reader during the \( j \)th 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 \( i \)
1. broadcast an advertisement \( \langle i, p_i \rangle \)
2. record the signal \( s_i \) in the report segment
3. extract \( ID_i \) from \( s_i \)
4. if the channel is idle during the report segment then
5. broadcast a negative acknowledgement
6. else if CRC in \( ID_i \) is verified to be correct then
7. \( S := \{ \langle ID_i, s_i \rangle \} \)
8. \( I := \emptyset \)
9. while \( S \neq \emptyset \) do
10. remove an element \( \langle ID, s \rangle \) from \( S \)
11. for \( j = 0 \) to \( i - 1 \) do
12. if \( H(ID|j) \leq p_i \) and \( R_j \) exists then
13. add \( s \) to the set of known individual signals in \( R_j \)
14. remove known signals from the mixed signal in \( R_j \)
15. extract \( ID' \) from the resulting signal \( s' \)
16. if CRC in \( ID' \) is verified to be correct then
17. \( S := S + \{ \langle ID', s' \rangle \} \)
18. \( I := I + \{ ID' \} \)
19. remove the collision record \( R_j \)
20. end for
21. end while
22. broadcast a positive acknowledgement and the IDs in \( I \)
23. else
24. add \( \langle i, s_i \rangle \) as a collision record
25. broadcast a negative acknowledgement

C. Determine 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 \( \lambda \) tags to transmit in the current slot.

Let \( N_i \) be the number of tags in the system. Its value can be estimated to an arbitrary accuracy [21] in a pre-step of SCAT. Before slot \( i \), the reader may have successfully collected and acknowledged a number \( n \) of tag IDs, and those tags will no longer participate in the protocol of SCAT. Let \( N_i \) be the number of tags that participate in slot \( i \). Since \( n \) 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 \( \binom{N_i}{k} \cdot p_i^k \cdot (1 - p_i)^{N_i-k} \). Our objective is to maximize the probability of \( X_i \in [0..\lambda] \), which is

\[
\sum_{k=1}^{\lambda} \binom{N_i}{k} \cdot p_i^k \cdot (1 - p_i)^{N_i-k} \quad (4)
\]

We expect \( \lambda \) to be small. In the following, we consider \( \lambda = 2, 3, \) or 4. When \( \lambda = 2 \), (10) becomes

\[
\sum_{k=1}^{2} \binom{N_i}{k} p_i^k \cdot (1 - p_i)^{N_i-k}
\]

\[
= N_i p_i (1 - p_i)^{N_i-1} + \frac{N_i(N_i - 1)}{2} p_i^2 (1 - p_i)^{N_i-2}
\]

\[
\approx N_i p_i e^{-N_i p_i} + \frac{N_i p_i^2}{2} e^{-N_i p_i} \quad (5)
\]
Let $\omega = N_i p_i$. Substituting $N_i p_i$ by $\omega$ in (11), we have

$$\sum_{k=1}^{2} \text{Prob}\{X_i = k\} \approx (\omega + \frac{\omega^2}{2})e^{-\omega}$$

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

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

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 $p_i = 2.213/N_i$.

D. Inefficiency to Be Improved

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 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 duration the protocol execution.

Second, the advertisement segment of each slot represents significant overhead that is not always necessary. For consecutive slots, the slot index changes from $i$ to $i + 1$ and the report probability changes from $\omega/N_i$ to $\omega/N_{i+1}$, 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 known its ID. We will show that reading 1 million tags uses no more than 2 million slots in our simulations. A length of 23 bits for slot indices will give sufficient safe margin.\(^1\) We will also show that the number of tags that transmits in more than three slots is exceedingly small. Hence, each tag only needs to remember no more than three indices. While a tag that does transmit more than three times may fail to stop and end up transmitting its ID again, the resulting overhead is negligible because such event is rare.

\(^1\)To handle the extremely rare case that the slot index space is exhausted, the reader may start the protocol anew to collect the IDs of the remaining tags.

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

We propose the Framed Collision-Aware Tag identification protocol (FCAT), which improves SCAT by eliminating the inefficiency described in Section III-D. FCAT shares much of the protocol details with SCAT. In the following, we will focus on describing their difference.

A. Protocol Description

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 $j$th slot in the $i$th 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 has a fixed length, consisting 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 $i$th 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.

B. Estimating Number of Tags within FCAT

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 similarity with [21]. However, in [21], 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 $j$th slot in the frame is empty, i.e., $X_j = 1$ means the $j$th 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 $j$th slot is a singleton slot. Because each tag decides to transmit with probability $p_i$ in every slot in the frame, we have

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

The expected value of $n_0$ is

$$E(n_0) = \sum_{j=1}^{f} (1 - p_i)^{N_i} = f(1 - p_i)^{N_i}. \quad (7)$$
The probability for the jth slot in the frame to be a singleton is
\[ \text{Prob}(Y_j = 1) = \binom{N_i}{1}p_i(1 - p_i)^{N_i - 1} = N_i p_i (1 - p_i)^{N_i - 1}. \]

The expected value of \( n \) is
\[ E(n_1) = \sum_{i=1}^{f} N_i p_i (1 - p_i)^{N_i - 1} = f N_i p_i (1 - p_i)^{N_i - 1}. \quad (8) \]

Obviously, \( E(n_0) + E(n_1) + E(n_c) = f. \) Hence
\[ E(n_c) = f - E(n_0) - E(n_1) = f (1 - (1 - p_i))^{f N_i - N_i p_i (1 - p_i)^{N_i - 1}} = f (1 - (1 - p_i)^{N_i - 1} (1 - p_i + \omega)). \quad (9) \]

The above equation can be rewritten as
\[ N_i = \frac{\ln(1 - \frac{E(n_c)}{f}) - \ln(1 - p_i + \omega)}{\ln(1 - p_i)} + 1 \quad (10) \]

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
\[ \hat{N}_i = \frac{\ln(1 - 2^{\frac{1}{n_c}}) - \ln(1 - p_i + \omega)}{\ln(1 - p_i)} + 1 \quad (11) \]

We derive \( E(\hat{N}_i) \) as follows. We first denote \( \frac{1}{\ln(1 - p_i)} = C_1 \) and \( -\ln(1 - p_i + \omega) + 1 = C_2 \) for simplicity. Letting \( \ln(1 - n_c^{\frac{1}{n_c}}) = g(n_c) \), we then expand the right hand side of the above equation by its Taylor series about \( q = E(n_c) \)
\[ \hat{N}_i = C_1 \left[ g(q) + (n_c - q)g'(q) + \frac{1}{2}(n_c - q)^2 g''(q) + \ldots \right] + C_2 \quad (12) \]

Since \( q = E(n_c) \), the mean of the second term in (12) is 0. Therefore, we keep the first three terms when computing the approximated value of \( E(\hat{N}_i) \)
\[ E(\hat{N}_i) = C_1 \left[ g(q) + \frac{1}{2} E((n_c - q)^2) g''(q) \right] + C_2 \quad (13) \]

We have \( E((n_c - q)^2) = V(n_c) \) by definition and \( g''(q) = -\frac{1}{(q - \frac{1}{f})^2} \), since \( g(q) = \ln(1 - \frac{q}{f}) \). Applying (18) in the appendix, we have
\[ E(\hat{N}_i) = N_i - \frac{\omega - 1 - \omega}{2f \ln(1 - p_i)(1 + \omega)} \quad (14) \]

Therefore,
\[ \text{Bias}(\frac{\hat{N}_i}{N_i}) = E(\frac{\hat{N}_i}{N_i}) - 1 = \frac{1 + \omega - \omega}{2f N_i \ln(1 - p_i)(1 + \omega)} \quad (15) \]

Figure 4 shows the absolute value of \( \text{Bias}(\frac{\hat{N}_i}{N_i}) \) with respect to the number of tags \( N_i \). The three lines show that the absolute values of \( \text{Bias}(\frac{\hat{N}_i}{N_i}) \) are 0.82%, 1.09% and 1.39%, which are all very small, with \( \omega = 1.414, 1.817 \) and 2.213 respectively.

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 \( N_i \), i.e., \( V(\hat{N}_i) = V(N_i) \). Because \( N_i < N \), \( V(\hat{N}_i) < V(N_i) \), and the value of \( V(\hat{N}_i) \) is derived in the appendix. It is approximately 0.0342, 0.0287 or 0.0265, for \( p_i = 1.414/N_i, 1.817/N_i \) or 2.213/Ni, respectively (i.e., 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 \( \bar{N}_i = \frac{\sum_{i=0}^{f} N^*_i}{f} \) as the estimation for \( N_i \), then the variance will decrease in the square root of \( i \) and therefore diminish as the protocol executes frame after frame.

V. Simulation Results

In this section, we present simulation results to evaluate the performance of our main protocol FCAT. We compare FCAT with some existing work, including the Dynamic Framed Slotted ALOHA (DFSA) [10], Enhanced Dynamic Framed Slotted ALOHA (EDFSA) [9], Adaptive Binary Splitting (ABS) [2] and Adaptive Query Splitting (AQS) [2]. The results are the average outcome of 100 runs.

In the following, we use FCAT-\( \lambda \) to denote the FCAT protocol in which \( k \)-collision slots with \( k \leq \lambda \) are resolvable, where \( \lambda = 2, 3, 4 \). FCAT uses the optimal report probability derived in Section III-C. Specifically, \( p_i \) is set to \( 1.414/N_i, 1.817/N_i \) and 2.213/Ni for FCAT-2, FCAT-3 and FCAT-4.
respective. The impact of $p_i$ is also studied in Section V-C. FCAT sets the frame size to 30 time slots. The parameters used in other compared protocols are selected based on their original papers whenever possible.

A. Reading Throughput

The reading speed is one of the most important design considerations for all tag identification protocols. We first compare it in terms of reading throughput which is defined as the number of tags read every second.

In our simulations, we set the time slot length according to Philips I-Code specification [22]. A slot consists of four segments: a tag reply segment, a reader acknowledgement segment and two waiting segments used to separate transmissions. The tag reply segment is 96 bits including 16 bits CRC code and the reader acknowledgement segment is 20 bits. The two waiting segments each costs 302 $\mu$s. The data transfer rate is 53 kbit/s and to transmit a bit costs 18.88 $\mu$s. Thus, it can be calculated that a slot is approximately 2.8 ms. From Table I, we can see that FCAT-2 achieves 53% improvement in reading throughput over DFSA which performs the best among all compared protocols. FCAT-4 can even double the reading throughput. As illustrated in Figure 5, FCAT also achieves good stability in reading throughput under different tag sizes.

We also evaluate the reading speed in terms of time slots. Table II shows the number of empty, singleton and collision time slots used to read 10000 tags. FCAT has much fewer singleton slots than all other compared protocols because FCAT can extract tags from collision slots, while others can only read a tag in a singleton slot. FCAT-4 has more collision slots than FCAT-2. The reason is that FCAT-4 can utilize a collision slot having up to four colliding reports, and hence FCAT-4 encourages more tags to transmit simultaneously.

B. Number of Reports

Active tags are equipped with limited battery power which can be very hard to recharge or replace, and thus energy efficiency becomes another important concern. Since a large portion of an active tag’s energy is spent on transmitting its ID to the reader, we evaluate the energy cost by counting the number of tag reports sent during the process of tag identification. Figure 6 shows the number of reports with various number of tags. ABS and AQS require much more reports than any of other protocols. Our protocols achieve relatively better performance than DFSA and EDFSA. We can see from Table III that with 10000 tags FCAT-2 reduces the number of reports by up to 12% and 35% respectively, when comparing with DFSA and EDFSA.

C. Report Probability

As we have discussed in Section [], the report probability $p_i$ is calculated by $\omega/N_i$, where $N_i$ is the number of tags participating in slot $i$ and $\omega$ is a constant parameter. An estimator is used to estimate $N_i$ after each frame. We also present a formula to calculate the optimal $\omega$ for different $\lambda$. Figure 7 shows the reading throughput with respect to $\omega$ when there are 10000 tags. The performance decreases as $\omega$ is set too small or too large. The optimal value is quite close to what we derived from the formula.

VI. 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 [23]. Most of these protocols fall into

\begin{table}[ht]
\centering
\caption{Reading Throughput when $N = 10000$}
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
$N$ & FCAT-2 & FCAT-3 & FCAT-4 & DFSA & EDFSA & ABS & AQS \\
\hline
10000 & 201.3 & 241.8 & 265.1 & 130.9 & 127.8 & 123.9 & 121.2 \\
\hline
\end{tabular}
\end{table}

\begin{table}[ht]
\centering
\caption{Empty, Singleton and Collision Time Slots when $N = 10000$}
\begin{tabular}{|c|c|c|c|c|c|c|c|}
\hline
 & FCAT-2 & FCAT-3 & FCAT-4 & DFSA & EDFSA & ABS & AQS \\
\hline
empty & 4189.0 & 2256.9 & 1345.4 & 10076.2 & 10705.2 & 4410.3 & 4737.3 \\
singleton & 5860.9 & 4054.9 & 2934.9 & 10000 & 10000 & 10000 & 10000 \\
collision & 7015.6 & 7496.6 & 8050.0 & 7207.9 & 7233.5 & 14409.3 & 14735.3 \\
total & 17742.6 & 14772.6 & 13473.6 & 27284.1 & 27938.6 & 28819.5 & 29472.5 \\
\hline
\end{tabular}
\end{table}

\begin{table}[ht]
\centering
\caption{Reading Throughput when $N = 10000$}
\begin{tabular}{|c|c|c|c|c|c|c|}
\hline
$N$ & FCAT-2 & FCAT-3 & FCAT-4 & DFSA & EDFSA & ABS & AQS \\
\hline
10000 & 7233.5 & 7496.6 & 8050.0 & 7207.9 & 7233.5 & 14409.3 & 14735.3 \\
\hline
\end{tabular}
\end{table}

\begin{figure}[ht]
\centering
\includegraphics[width=\textwidth]{fig6.png}
\caption{Number of tag reports under different tag sizes.}
\end{figure}
two classes: ALOHA-based protocols [8], [9], [10], [11], [12], [13], [14] and tree-based protocols [2], [5], [6], [7], [3], [4].

In the ALOHA-based protocols, the reader broadcasts a request and each tag randomly selects a slot to report its ID. If exact one tag response, the reader retrieves its ID and this tag will remain silent for the rest of reading process. Simultaneous responses in a slot will lead to collision. Thus, 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 [8], 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 [9], slots are grouped into frames with the same fixed 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. For this reason, the dynamic framed slotted ALOHA (DFSA) [10] introduces frames with dynamic sizes. It is proved that the maximal reading throughput is achieved when the frame size is equal to the number of unread tags [10]. So, 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 [9]. The enhanced dynamic framed slotted ALOHA [9] uses frames with limited size by restricting the number of responding tags in a frame. The maximal reading throughput of the ALOHA-based protocols is bounded by $\frac{1}{\tau}$ [18]. 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 regarded 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 response concurrently, the reader split them into smaller subsets. The procedure repeats until every set only contains a single tag which can be identified by the reader. Different splitting criteria lead to different protocols. The binary-tree protocols [2], [5], [6], [7] 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 response. Once a collision happens, the reader sends a new query with an indication of the collision. A 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 one. During a collision, all other tags which do not transmit also increase their counters by one. If no collision happens, all tags decrease their counters by one. An analysis shows that the maximal tag reading throughput of binary-tree protocols is $\frac{1}{\tau}$ [7]. The query-tree protocols [2], [3], [4] use the tag ID for splitting. A tag ID is a unique bit string. Each query contains a prefix $p_0 p_1 .. p_i$ where $i$ is the length of the prefix. For each tag whose ID contains this prefix, it transmits its ID as a reply. If multiple responses collide, the reader generates two longer prefixes $p_0 p_1 .. p_i 0$ and $p_0 p_1 .. p_i 1$ by attaching a bit 0 and 1, respectively. So, the set of colliding tags is divided into two subsets according to prefixes contained in their IDs. A query-tree protocol can have quite different reading throughput determined by the tag ID distribution.

VII. 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 58%. As the technologies of physical-layer network coding are improved, the reading throughput can potentially be doubled.

REFERENCES

Therefore, from (21), the variance of $\hat{N}_i$ is

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

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.

$$n_c \xrightarrow{D} \text{Norm}(\theta, \delta^2)$$

where $\theta$ is $E(n_c)$ as given in (9), $\delta^2$ is $V(n_c)$ as given in (18), and $\xrightarrow{D}$ means convergence in distribution.

According to the $\delta$-method [24], we have

$$h(n_c) \xrightarrow{D} \text{Norm}(h(\theta), \delta^2 [h'(\theta)]^2)$$

(19)

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

In Section IV-B, the estimation formula is designed based on (9), 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 $E(n_c) = g(N_i)$. Figure 3 shows that $g(.)$ is a monotonic function, and hence it has a unique inverse function, denoted as $h(.)$

According to Section IV-B, $N_i$ is computed from (9) by substituting $E(n_c)$ with the instance value of $n_c$ (obtained after the $t$th frame).

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

$$\approx f(1 - e^{-N_i p_i} - N_i p_i e^{-N_i p_i})$$

(20)

Clearly, $n_c = g(\hat{N}_i)$ and $\hat{N}_i = h(n_c)$. Applying $\hat{N}_i = h(n_c)$ to (19), we have

$$\hat{N}_i \xrightarrow{D} \text{Norm}(h(\theta), \delta^2 [h'(\theta)]^2)$$

(21)

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

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

(22)

Therefore, from (21), the variance of $\hat{N}_i$ is

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

(23)

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 $\omega$ 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 $\omega$ in (24), and the variance $V(\frac{\hat{N}_i}{N_i})$ is 0.0342, 0.0287 or 0.0265 respectively for different $\omega$ values.